RTSP Announce Method
draft-ietf-mmusic-rtsp-announce-01

Status of this Memo

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

By submitting this Internet-Draft, I (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 obsolete by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or cite them other than as "work in progress".

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

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

This document is an individual submission to the IETF. Comments should be directed to the authors.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

Changes since the last revision:

The only change is the update of dates in order to prevent the draft from expiring.

This memo describes an extension RTSP request, ANNOUNCE, which extends the core RTSP protocol [RTSP_NEW] to allow one end to push to the other end various types of information by RTSP means.

This RTSP extension is identified by a feature tag, "method.announce", and supports capability exchange via the feature-tag framework as detailed in [RTSP_NEW].

Examples of information in ANNOUNCE requests include session descriptions and end of stream events.

The receiver of the ANNOUNCE request is expected to reply with 200 OK response.
1. Motivation

In RFC2326, ANNOUNCE is part of the RTSP protocol. In the updated core RTSP protocol [RTSP_NEW], however, ANNOUNCE method has been removed from the core RTSP protocol because, for one, ANNOUNCE is not required for basic RTSP playback, for the other, ANNOUNCE had a lack of implementation at the time when [RTSP_NEW] was being conceived.

Nonetheless, there are advanced use cases that require ANNOUNCE method for the server to asynchronously publish session descriptions or other event information. It is clear that such functionality needs to be made available in a way consistent to the extension mechanism in [RTSP_NEW].

The first use case is for either the server or the client to publish a new or updated session description pertinent to a RTSP session URL. Specifically, a multi-unicast live video server utilizing RTSP may want to publish an updated SDP (Session Description Protocol) when a new media track is added to the RTSP session. When client receives the ANNOUNCE request (with an SDP entity body), it has the option to perform SETUP on the newly available media track.

The second use case is for the server to signal end of stream event to its client(s). Appendix A presents the reasons why ANNOUNCE is better suited to signal end of stream than the other options using RTCP BYE packet, RTSP TEARDOWN, PAUSE or SET_PARAMETER requests.

Given the use cases presented above, we propose to utilize ANNOUNCE method to signal several types of events common to RTSP-based media applications, including session description events and End-Of-Stream events.

2. The Definition of ANNOUNCE method

This memo defines ANNOUNCE as an extension to the core RTSP protocol [RTSP_NEW]. It presents ANNOUNCE method as a general mechanism for RTSP server to signal to its clients various events including end of stream events or session description updates events. This memo will discuss the general usage of ANNOUNCE, its feature tag, as well as a new "Event-Type" header for ANNOUNCE method.

[RTSP_NEW] has defined a mechanism to extend the core RTSP protocol. Following that mechanism, a feature tag is used to identify ANNOUNCE method as an extension to the core RTSP protocol.

The ANNOUNCE method is an RTSP request that can be sent in both directions, either from client to server or server to client. When server intends to send ANNOUNCE to client, it must have the means to reach the client, because the RTSP client is not required to keep a persistent connection with the RTSP server. It is beyond the scope of this memo to define the exact means for server to reach client. It suffices to say that if the client intends to receive server’s ANNOUNCE requests, it must keep the RTSP connection open, or inform the server on how to reach it without a persistent RTSP connection.

Below is an example RTSP conversation in which an RTSP server announces an end of stream event for a media stream using a
non-aggregate URI. The new header, "Event-Type" is formally defined later in this section.

S->C: ANNOUNCE rtsp://foo.com/bar.avi/streamid=0 RTSP/1.0
    CSeq: 10
    Session: 12345678
    Event-Type: 2000 End-Of-Stream
    Range: npt=0-200
    RTP-Info: url=rtsp://foo.com/bar.avi/streamid=0;seq=45102
    Content-Type: text/parameters
    Content-Length: 49
    eos-reason: Reached the end of requested range.

C->S: RTSP/1.0 200 OK
    CSeq: 10
    Session: 12345678

2.1 Normative definitions

"ANNOUNCE" is an "extension-method" in the ABNF in section 16.2 "RTSP Protocol Definitions" in [RTSP_NEW].

The request-URI of an ANNOUNCE request can be either aggregate or non-aggregate URI.

An ANNOUNCE request must include "CSeq" header. It MAY include the following optional headers:

- "Range",
- "Session",
- "RTP-Info",
- "Event-Type"

An ANNOUNCE request MAY include entity body, in which case it MUST follow the rules for entity body defined in section 8.2 of [RTSP_NEW]. Entity body can be used to convey further details specific to an event type. For instance, if the event type is session description announcement, the actual SDP SHOULD be included in the entity body. If the event type is end-of-stream announcement, the entity body MAY contain "text/parameters" content type that conveys the reason of the end-of-stream event.

ANNOUNCE does NOT affect RTSP session state. If a receiver does not understand any of the headers in an ANNOUNCE request, it simply ignores those headers.

The next section defines a new RTSP headers for ANNOUNCE method: "Event-Type".

2.2 Event-Type Header

The Event-Type header is an optional header to identify the type of event pertaining to the ANNOUNCE request. Example event types include session description, end of stream and error.

If an ANNOUNCE request does not contain Event-Type header, the Event-Type defaults to "Session-Description", consistent with RFC2326.

The Event-Type header is defined in ABNF as:
Event-Type = "Event-Type" ":" event-type
event-type = event-type-code SP event-type-string
event-type-code = 4DIGITS
event-type-string = token

where:
-- token is defined in section 17 of [RTSP_NEW].

The only method that "Event-Type" header applies is the ANNOUNCE method, either from client to server or from server to client.

The following pairs of event-type-code and event-type-string are defined in this memo.

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>Session-Description</td>
</tr>
<tr>
<td>2000</td>
<td>End-of-Stream</td>
</tr>
<tr>
<td>3000</td>
<td>Error</td>
</tr>
</tbody>
</table>

If "Event-Type" header is missing, the default is "1000 Session-Description". This is to be consistent with the usage of ANNOUNCE in RFC2326.

If "Event-Type" is "2000 End-Of-Stream", the optional RTP-Info header SHOULD contain the "seq" attribute that indicates the sequence number of the next RTP packet. See example in section 4.2.

2.3 Limitations on serve to client "ANNOUNCE" requests

Server to client ANNOUNCE method is issued only if the server has the means to contact the client when it has information to push. This may not be possible if the RTSP connection between server and client is not persistent. In such cases, the server will simply skip the sending of ANNOUNCE requests. That is to say, the server will not queue up the ANNOUNCE requests to be sent when client eventually connects. Such a queue would unnecessarily complicate server implementations.

3. Feature tag

The support of the ANNOUNCE method is represented by the feature tag below:

    method.announce

This feature tag applies to both servers and proxies.

Implementations claiming "method.announce" feature tag MUST support the new "Event-Type" header defined in previous section.

4. Use Cases

This section presents several use cases of the ANNOUNCE method.

4.1 Client Announcing SDP To Server For Recording

This use case is the same as the first RTSP exchange presented in section 14.6 in RFC2326, with capability exchange via OPTIONS method.
The conference participant client C asks the media server M to record the audio and video portions of a meeting. After first verifying that the server supports the "ANNOUNCE" feature, the client uses the ANNOUNCE method to provide meta-information about the recorded session to the server. The client omits "Event-Type" because "Event-Type: Session-Description" is the default.

C→M: OPTIONS * RTSP/1.0
   Require: method.announce
   CSeq: 1

M→C: RTSP/1.0 200 OK
   CSeq: 1
   Supported: method.announce
   Public: DESCRIBE, SETUP, TEARDOWN, PLAY, PAUSE, RECORD, \ ANNOUNCE

C→M: ANNOUNCE rtsp://server.example.com/meeting RTSP/1.0
   CSeq: 90
   Content-Type: application/sdp
   Content-Length: 121

   v=0
   o=camera1 3080117314 3080118787 IN IP4 195.27.192.36
   s=IETF Meeting, Munich - 1
   i=The thirty-ninth IETF meeting will be held in Munich
   u=http://www.ietf.org/meetings/Munich.html
   e=IETF Channel 1 <ietf39-mbone@uni-koeln.de>
   p=IETF Channel 1 +49-172-2312 451
   c=IN IP4 224.0.1.11/127
   t=3080271600 3080703600
   a=tool:sdr v2.4a6
   a=type:test
   m=audio 21010 RTP/AVP 5
   c=IN IP4 224.0.1.11/127
   a=ptime:40
   m=video 61010 RTP/AVP 31
   c=IN IP4 224.0.1.12/127

M→C: RTSP/1.0 200 OK
   CSeq: 90

4.2 Server Announcing End Of Stream

In this example, the server announces the End-Of-Stream event to client for one live media stream, because upstream source terminates the stream after 200 seconds. The fact that the stream has played for 200 seconds is communicated by the Range header in the ANNOUNCE request. The fact that the server has sent a total of 45102 RTP packets is conveyed in the RTP-Info headers.

C→S: PLAY rtsp://foo.com/bar.avi/streamid=0 RTSP/1.0
   Supported: method.announce
   CSeq: 10
   Session: 12345678
   Range: npt=0-200

S→C: RTSP/1.0 200 OK
   Supported: method.announce
   CSeq: 10
   Session: 12345678
RTP-Info: url=rtsp://foo.com/bar.avi/streamid=0;seq=0; rtptime=0

S->C: ANNOUNCE rtsp://foo.com/bar.avi/streamid=0 RTSP/1.0
CSeq: 123
Session: 12345678
Require: method.announce
Event-Type: 2000 End-Of-Stream
Range: npt=0-200
RTP-Info: url=rtsp://foo.com/bar.avi/streamid=0;seq=45102

Content-Type: text/parameters
Content-Length: 49

eos-reason: reached the end of requested range.

C->S: RTSP/1.0 200 OK
CSeq: 123
Session: 12345678

The Require header in the above ANNOUNCE request indicates that in order to understand the "Event-Type" header the client must support the feature tag in the Require header. In this case the client happens to signal its support in its PLAY request.

From the ANNOUNCE request, the client will learn that the server has completed the stream as requested.

From the two RTP-Info headers, one in PLAY response, one in ANNOUNCE request, the client can derive the total number of RTP packets that the server has sent. In this example, the server has sent RTP packets 0 to 45101, for a total of 45102 packets. The "seq" attribute of the RTP-Info header in ANNOUNCE tells the client that the next RTP packet was going to be packet number 45102 when the stream stops.

From the npt field in the Range header, the client can derive the presentation time that this stream has covered.

5. Security Considerations

Because there is only one new TEXT header, "Event-Type", added by the extension RTSP method, the security considerations outlined in [RTSP_NEW] apply here as well.

6. IANA Considerations

A new method name, its associated feature tag, and a new header, need to be registered with IANA.

   -- Method name: ANNOUNCE. See section 2.1 for the relevant definition.
   -- Feature tag: method.announce. See section 3 for the relevant definition.
   -- Header name: Event-Type, see section 2.2 for the relevant information.

Appendix A: Justification for Using ANNOUNCE to Signal End Of Stream

This appendix presents the reasons why we have selected the ANNOUNCE proposal from several proposals to signal end of stream. The competing proposals were based on:

1) RTCP BYE packet,
2) RTSP TEARDOWN request,  
3) RTSP PAUSE request,  
4) SET_PARAMETER request.

In the core RTSP protocol [RTSP_NEW], an RTSP client relies on the media transport mechanism to signal end of stream.

When the media transport mechanism happens to be RTP over UDP, this is carried out by RTCP BYE packet [RTP_NEW]. In practice, there are some drawbacks with this approach:

1. When the server sends an RTCP BYE packet with its SSRC, the server is giving up on the SSRC (see section 8.2 in [RTP_NEW]). The server would be required to switch to a new SSRC on a subsequent PLAY of the same media stream. Since server’s SSRC is only communicated in the Transport header of SETUP response, the server would not have an opportunity to send a new value to the player, and the client would have to discover the SSRC from the incoming RTP packets -- a non-trivial process.

2. RTCP BYE packet method does not offer a simple, guaranteed method of delivering an end-of-stream announcement, given BYE packet is carried over UDP.

3. RTCP BYE packet method does not offer the option to have a single aggregate end-of-stream announcement for all media streams in the RTSP session.

4. Section 6.3.7 of RFC3550 stipulates that an RTP sender cannot send RTCP BYE before leaving the RTP session if it has not already sent at least one RTP or RTCP packet. This is a problem under error conditions. Consider the case where an RTP session has just started (i.e., RTSP PLAY has been successfully acknowledged with an RTSP 200 OK response), and the sender attempts to retrieve media frames from its media source. The media source fails to provide any media frame due to its internal error such as file corruption. The sender should inform its receiver(s) but it cannot send BYE packets.

The motivation to solve the above issues is particularly high for unicast-streaming applications that use RTSP over TCP in the control plane, and RTP over UDP in the media transport.

There is also the desire to have an EOS (End Of Stream) signaling mechanism for non-RTP delivery. One such delivery is MPEG2 transport streams used in the cable TV environment. In non-IP delivery environments, the transport typically remains allocated even if no media is being delivered. This means that a client cannot watch for the server to close the transport to signal the end of stream. Meanwhile, watching for the incoming media to stop is unreliable. Short timeouts can trigger a false end of media detection if the media flow is temporarily delayed. Long timeouts introduce
unacceptable latencies. Clients are unable to distinguish between a normal end of stream and an error condition that resulted in the media delivery stopping.

We note that using TEARDOWN from server to client is not appropriate because:
1. TEARDOWN is currently not allowed from server to client [RTSP_NEW];
2. Even if TEARDOWN is made available in server to client direction, the definition of TEARDOWN requires that, if the request URI is aggregate, that the session must be de-allocated by the server.
There are RTSP applications that use SET_PARAMETER from client to server as the means to report session QoS statistics, but if server uses TEARDOWN on aggregate URL to signal end of stream, the client can no longer use SET_PARAMETER with a session header.
3. In general, RTSP, being a client-server protocol, should let client, not server to control session state. But TEARDOWN on aggregate URL will change session from PLAYING state to INIT state.

We note that using PAUSE from server to client is not appropriate either, because PAUSE will change the state of the RTSP session.

We note that using SET_PARAMETER from server to client will require at least two parameters in the entity body, one for event type that should be set to End-Of-Stream, and the other parameter for the reason of End-Of-Stream. Also using SET_PARAMETER with an SDP entity body to update session description will not be compatible with RFC2326 where ANNOUNCE was defined for that purpose. Therefore, SET_PARAMETER is not appropriate to convey the announcement of End-Of-Stream and other events.

Normative References


Acknowledgement

Thanks to Sean Sheedy for suggesting the inclusion of "Range" header and for contributing part of the text in the motivations section.

Thanks to Anders Klemets for suggesting the current semantics of the "seq" attribute in the RTP-Info header.

Author Addresses

Thomas Zeng
PacketVideo Network Solutions
9605 Scranton Road, Suite 400
IPR Notice

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.

Copyright Notice

Copyright (C) The Internet Society (2004). 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.

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.