

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 21, 2015

R. Finlayson
Live Networks, Inc.
October 18, 2014

**Notifying RTSP Clients and Proxy Servers About Streams: The RTSP
"REGISTER" Command
draft-finlayson-rtsp-register-command-02**

Abstract

We define a new RTSP command - named "REGISTER" - that allows a server (or a third party) to notify a RTSP client or a RTSP proxy server about a RTSP stream (given by a "rtsp://" URL). This also provides a mechanism whereby a client (or proxy server) on the public Internet can access a RTSP server that is behind a firewall or NAT.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on April 21, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Requirements Language	2
2.	Introduction	2
3.	An Example	3
4.	Optional Transport Parameters	4
5.	Comparison with the "ANNOUNCE" command	5
6.	IANA Considerations	5
7.	Security Considerations	6
8.	References	6
8.1.	Normative References	6
8.2.	Informative References	7
	Author's Address	7

[1.](#) Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

[2.](#) Introduction

The Real Time Streaming Protocol (RTSP) [\[RFC2326\]](#) is used primarily by a client to access a server's multimedia stream - specified by a "rtsp://" URL. The client is assumed to know the stream's URL in advance; the RTSP protocol does not specify any mechanism by which a client can be informed of the stream's URL.

This document defines an optional new RTSP command - named "REGISTER" - that allows a server (or a third party) to notify a RTSP client or a RTSP proxy server about a RTSP stream (given by a "rtsp://" URL). (Note that this command is defined only when sent to a RTSP client (or a proxy server, which is effectively both a client and a server). Unlike most other RTSP commands, it is not defined when sent to a (pure) RTSP server.)

Possible applications of the "REGISTER" command include:

Initialization of a proxy server

A RTSP proxy server acts as both a client (to access a stream from a 'back-end' server) and as a server (to serve this stream to one or more 'front-end' clients). The "REGISTER" command can be used to inform a proxy server about a stream to deliver from a 'back-end' server.

This is particularly useful if the proxy server is on the public Internet, but the 'back-end' server is on a private network (behind a NAT), without a public IP address. In this case the private 'back-end' server can use the "REGISTER" command to inform the proxy server about one of its streams. In addition, optional parameters - described below - allow the proxy server to access this stream via the same TCP connection that was used to send the "REGISTER" command, and perhaps also to deliver RTP and RTCP packets over this same TCP connection (using RTP/RTCP-over-TCP encapsulation, i.e., 'interleaving'). Note that this basic approach to NAT traversal - TCP tunneling - is just one of the many possible NAT traversal techniques for RTSP outlined in [\[I-D.ietf-mmusic-rtsp-nat-evaluation\]](#). Other more general (but also much more complex) approaches are also possible - for example [\[I-D.ietf-mmusic-rtsp-nat\]](#).

Initialization of a 'pure' client

The "REGISTER" command can also be used to inform a 'pure' RTSP client (i.e., a RTSP client that does not also act as a (proxy) server) about a stream to be played. This is less useful than for a proxy server, because for most (pure) RTSP clients, the stream to be played is usually entered by a human user (e.g., in entered in a GUI dialog box, or on the command-line). However, it might still be useful for systems that have many RTSP clients to choose from, and/or wish to control the time at which a RTSP client should request a stream.

3. An Example

The following protocol exchange illustrates how a 'back-end' media server (on a private network) can inform a proxy server (on the public Internet) about one of its streams - in a way that allows the proxy server (and thus 'front-end' clients on the public Internet) to access the stream.

Server (10.42.42.42) -> Client (a proxy server: example.com)

```
REGISTER rtsp://10.42.42.42/camera RTSP/1.0
```

```
CSeq: 1
```

```
Transport: reuse_connection;
```



```
preferred_delivery_protocol=interleaved;  
proxy_url_suffix=proxiedCamera
```

```
Client (a proxy server: example.com) -> Server (10.42.42.42)  
RTSP/1.0 200 OK  
CSeq: 1
```

Because of the "reuse_connection" parameter, the proxy server (example.com) can then use the existing RTSP (TCP) connection to access the 'back-end' stream "rtsp://10.42.42.42/camera" using the usual RTSP "DESCRIBE", "SETUP" and "PLAY" commands. In addition, because of the "preferred_delivery_protocol=interleaved" parameter, the "SETUP" command can request RTP/RTCP-over-TCP streaming, using the standard "interleaved=" mechanism ([\[RFC2326\]](#), [section 10.12](#)). Finally, the "proxy_url_suffix=proxiedCamera" parameter tells the proxy server that it can provide access to the stream using the URL: "rtsp://example.com/proxiedCamera".

4. Optional Transport Parameters

Three optional parameters (noted in the example above) can be specified in the "Transport:" header of a "REGISTER" command:

reuse_connection

This parameter is a flag, with no value.

If this parameter is present, the receiving client SHOULD reuse the TCP connection (on which it received the "REGISTER" command) for subsequent commands on the Request-URI.

preferred_delivery_protocol

Possible values: "udp", "interleaved". Default value: "udp".

This parameter specifies the delivery protocol that the receiving client SHOULD request when it subsequently sends "SETUP" commands for the Request-URI. The default value, "udp", specifies that the client should request normal RTP/RTCP-over-UDP streaming. The value "interleaved" specifies that the client should request interleaved RTP/RTCP-over-TCP streaming (using the RTSP TCP connection), as described in [\[RFC2326\]](#), [section 10.12](#).

proxy_url_suffix

A text string with syntax <path-noscheme> (as defined in [\[RFC3986\]](#) or [\[I-D.ietf-mmusic-rfc2326bis\]](#)).

This parameter is meaningful only if the receiving client is a proxy server. It specifies a suffix that the proxy server SHOULD use in the "rtsp://" URL that (front-end) clients use to access the stream. If the proxy server is already

supplying a stream with this same URL suffix, then it SHOULD return an error (e.g., "451 Invalid parameter") in response to the "REGISTER" request.

5. Comparison with the "ANNOUNCE" command

RTSP 1.0[RFC2326] defined a command - named "ANNOUNCE" - that could be used to notify a server about a new stream. The "ANNOUNCE" command worked by including - as part of the command - a SDP[RFC4566] description for the stream. A subsequent RTSP command would then be used to 'inject' media (as RTP/RTCP packets) into the server, which would then make this media available to other RTSP clients. Unfortunately the RTSP 1.0 standard did not fully specify how this would work; in particular, which RTSP command should be used to inject media into the server. Although the "RECORD" command appeared to be the most logical choice, most implementations of this mechanism used the "PLAY" command instead. Because this mechanism was not fully specified in RTSP 1.0 (and not extensively used), both the "ANNOUNCE" and "RECORD" commands were removed from the RTSP 2.0 specification[I-D.ietf-mmusic-rfc2326bis]).

Nonetheless, some commercial RTSP servers did implement this mechanism, which was useful for serving streams that originate from a private network (behind a NAT). Such a stream could be "ANNOUNCE"d and then injected into a RTSP server located on the public Internet. However, implementing this mechanism added a lot of complexity to the RTSP server, because it needed to be able to process incoming SDP descriptions and RTP/RTCP streams, in addition to its usual functionality of generating outgoing SDP descriptions and RTP/RTCP streams. In contrast, the "REGISTER" command, described in this document, is simpler and much easier to implement, because it requires merely adding familiar RTSP client functionality to the server - i.e., implementing a RTSP proxy server.

The "REGISTER" command also simplifies the media source application, because it does not have to rely upon having to "ANNOUNCE" and 'inject' its data into a separate server. Instead, the media source application can simply be a RTSP server of its own. RTSP clients on the same network can access the stream directly (just like any other RTSP stream), but if the media source application also wishes to have its stream relayed via a separate server (e.g., across a NAT), then it can do so simply by adding a small piece of extra code that sends a "REGISTER" command to this server.

6. IANA Considerations

If this proposal were adopted as an IETF standard, then two IANA registries would need to be updated:

- o "RTSP Commands". (This registry is currently defined for RTSP 1.0[RFC2326], and proposed for RTSP 2.0[I-D.ietf-mmusic-rfc2326bis].)
- o "Transport Parameters" (This registry was not defined for RTSP 1.0, but has been proposed for RTSP 2.0.)

7. Security Considerations

The addition of this new command adds no additional threat to existing RTSP clients (or servers), because they will simply reject any incoming "REGISTER" command (a command that they do not handle) with an error code.

However, for a proxy server, the "REGISTER" command causes it to access (and then potentially serve) a new stream, which is more powerful functionality than the normal server functionality of serving existing streams (by implementing the regular RTSP "DESCRIBE", "SETUP", "PLAY" etc. commands). Therefore any proxy server on the public Internet that implements the "REGISTER" command SHOULD implement access control for this command, even if it does not implement any access control for "DESCRIBE", "SETUP", "PLAY" etc.

8. References

8.1. Normative References

- [I-D.ietf-mmusic-rfc2326bis]
Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, "Real Time Streaming Protocol 2.0 (RTSP)", [draft-ietf-mmusic-rfc2326bis-28](#) (work in progress), October 2011.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2326] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", [RFC 2326](#), April 1998.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.

8.2. Informative References

[I-D.ietf-mmusic-rtsp-nat-evaluation]

Westerlund, M. and T. Zeng, "The Evaluation of Different Network Address Translator (NAT) Traversal Techniques for Media Controlled by Real-time Streaming Protocol (RTSP)", [draft-ietf-mmusic-rtsp-nat-evaluation-13](#) (work in progress), February 2014.

[I-D.ietf-mmusic-rtsp-nat]

Goldberg, J., Westerlund, M., and T. Zeng, "A Network Address Translator (NAT) Traversal Mechanism for Media Controlled by Real-Time Streaming Protocol (RTSP)", [draft-ietf-mmusic-rtsp-nat-20](#) (work in progress), February 2014.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.

Author's Address

Ross Finlayson
Live Networks, Inc.
650 Castro St., suite 120-196
Mountain View, CA 94041
USA

Email: finlayson@live555.com
URI: <http://www.live555.com/>

