Real-Time Streaming Protocol

draft-ietf-mmusic-rfc2326bis-01.txt

Magnus Westerlund

New in draft…-01

- Header table in chapter 12
- Updated all HTTP references to RFC 2616
- State machine chapter totally rewritten
- Added PING method
- IANA section created
- Clarified usage of range with open start time
- Clarified usage of pause point

Issues to Discuss

- Feature discovery
- Protocol extensions: IANA Section
- Persistent connections
- Redirect
- Connection related feature tags
- OPTIONS method implementation requirements

Extensibility and feature discovery 1/2

- The existing method with three headers:
  - **Require and Proxy-require**: Puts conditions on the request that the given features must be supported and used or the request shall not be performed.
  - **Unsupported**: In a response to a request with require this header gives the missing features.
- The current require headers can not be used to check if the other party does support a feature.

Extensibility and feature discovery 2/2

- The new proposed addition for feature discovery
  - **Supported**: Taken from SIP. Requester includes all supported feature tags. Responder replies with all his supported tags.
- The definition may not be what we want for RTSP due to possible message sizes.
- Proposed to create new header with other syntax:
  - Requester includes feature tags he wants to know if the responder supports. The responder removes any that it does not support.
- Which solution is desired?
- In which methods are the header allowed to be used?

Extensibility - IANA section

- New IANA section sets up four repositories:
  - Methods: Requires IETF standards action
  - Headers: Requires specification
  - Parameters: Done on first come, first served basis
  - Feature tags (option-tags): Done on first come, first served basis
- The chapter also defines rules what such a registration and possible specification must contain.
- Unregistered headers shall start with “X-”.
- Is these requirement levels the desired ones?

Persistent connections

- Needs backwards compatible solution
- Try to allow client maximum freedom
- Having simple rule for servers
- Should support non-persistent connections.
- Proposed Solution: Servers are not allowed to close connection until connection timeout (>= RTSP timeout). Clients may close connection at any time. Clients are strongly recommended to keep connections open, to allow Server to send requests. Closing connection may result in server closing session on old servers (feature tag).

REDIRECT Issues

- Two ways of doing redirect:
  - 3xx response: Should be used before session is established.
  - REDIRECT method: Used to redirect already established sessions.
- The REDIRECT method does not work unless a persistent connection is used.
- Shall the server directly upon receiving the response for a REDIRECT remove the session?
- The REDIRECT MUST or SHOULD contain a session header?
- How to use Range header in REDIRECT?

Two connection related feature tags

- It is proposed that we define two feature tags:
  - ping-b4-session-timeout
  - ping-b4-connection-timeout
- If these features are supported the server is required to ping the client before closing either the RTSP session or the persistent connection.
- The server will then close the session or connection if fails to ping the client with one timeout period.
- It requires the client to use a persistent connection.
- Should this be in the base spec or a extension?

OPTIONS method implementation requirements

- Minimal client implementations are not required to implement OPTIONS
- Server implementations MUST implement it
- Should also clients be required to implement it?

The nearest future of RTSP

- Update the draft significantly to version 02. Includes getting rid of all known contradictions and fix the resolved bugs.
- Continue the work on interoperability tests preparations.