Real-Time Streaming Protocol

draft-ietf-mmusic-rfc2326bis-11
draft-ietf-mmusic-rtsp-nat-04

Magnus Westerlund

Changes in RTSP

- Changed version from 1.1 to 2.0.
  - This is motivated by that we should follow RTSP 1.0 rules for versioning which leans on HTTP's rules. Thus the changes to Transport header requires a new major version number.
- Updated ABNF Ref to RFC 4234
- Changed sentence ending “after further experimentation.” regarding SET_PARAMETER payloads.
- Added the consensus regarding using Allow header also in DESCRIBE and SETUP method.
- Clarified that any “string” in ABNF is case-insensitive.
- Restricted the Transport header parameter “mode” to only use a quoted string structure.

Open Issue

- The minimal Implementation section needs revising.
- Editorial Review
- Possible Simplifications

RTSP and NAT: Changes

- Added an outline on how to use ICE with RTSP to the draft.
- Updated the references in the draft to be current
- Replaced STUN language classification for NATs with BEHAVE language

RTSP and NAT: ICE Proposal

1. The session description indicate that ICE may be used. Proposal would be ”a=ICE-capable” for SDP. For DESCRIBE a feature-tag could also be used to indicate the server’s capability. However this is only a hint and not necessary.
2. Clients prepares the SETUP: Gathers its addresses.
3. Client sends a SETUP with ICE parameter containing candidates in the transport header. Syntax could be:
   ICE=“candID compID passwd TRN qvalue addr port, candID..”

   Example:
   SETUP rtsp://foo.com/test.wav/streamid=0 RTSP/2.0
   Transport: RTP/AVP/UDP;unicast;
   dest_addr="120.23.34.53:6970"/"120.23.34.53:6971";mode="PLAY**"**;**ICE=" 1 1 ytytytytyytyyt UDP 0.7 120.23.34.53 6970, 1 2 opopopopopo UDP 0.7 120.23.34.53 6971, 2 1 asfasdadasdad UDP 0.9 10.10.10.10 5310, 2 1 mmnmnmnmnmn UDP 0.9 10.10.10.10 5311"**

RTSP and NAT: ICE Proposal

4. The server gathers its address and responds with its candidate. A server in public would only provide a single candidate:
   RTSP/2.0 200 OK
   Transport: RTP/AVP/UDP;
   unicast;dest_addr="120.23.34.53:6970"/"120.23.34.53:6971";
   src_addr="192.0.2.5:45784"/"192.0.2.5:45785";
   mode="PLAY";ssrc=EAB98712; **ICE="1 1 klklklklklkklk UDP 1.0 192.0.2.5 45784, 1 2 wewewewewe UDP 1.0 192.0.2.5 45785“**
5. Connectivity checks are performed. Possible results:
   1. The primary candidate reaches valid state
   2. Another candidate reaches valid state
   3. All the checks timeout => communication failure unless more candidates are available => go to step 3

RTSP and NAT: ICE Proposal

6. If another candidate pair, for example a peer-derived, than the primary contained in the SETUP has reached valid state while not the primary then new SETUP requests must be sent to update the src_addr and dest_addr.
7. The client can now send its PLAY request to the server
8. Server upon reception of PLAY verifies that connectivity checks are completed, otherwise do not provide 200 answer until it is done (1xx answer could be defined). Upon valid state for primary candidates start playing (200 OK).

RTSP and NAT: ICE Proposal

- Issues:
  - How to prevent the that a long list of candidates consumes to much server resources.
  - Is continuing to perform STUN the best way of NAT keep-alive in regards to mechanisms and server resources.
- Cost
  - In best case only the connectivity checks in extra setup time
  - If not primary address is valid, then one extra RTT is needed. Multiple SETUPs can in this case be pipelined.

RTSP and NAT: Way forward

- Propose to continue develop the ICE solution for RTSP
  - ICE resolves the Denial of Service potential for RTSP which would also allow greater flexibility in RTSP usage
  - Will be capable of handling deployment of RTSP servers behind NATs
- Move all other proposals that requires server modifications to an informational annex for history
- Keep client side only methods that don’t have issues as possible methods, especially for RTSP 1.0 usage.
- Clarify firewall and ALG recommendations
- Align it with RTSP 2.0 specification