Rapid Synch for RTP Multicast Sessions

draft-versteeg-avt-rapid-synchronization-for-rtp-01

IETF 73 – November 2008

Bill Ver Steeg, Ali C. Begen and Tom Van Caenegem

{billvs, abegen}@cisco.com, Tom.Van_Caenegem@alcatel-lucent.be
### A Typical Multicast Join Scenario

Time the receiver needs to wait to start processing multicast data

- Applications can start processing multicast data only after getting the key information at random access points.
- RAPs might be far away from each other.
- Key information might be large in size and non-contiguous.

→ These increase the delay to synchronize with the multicast flow.
Proposed Approach

- Prior to join, receiver requests a repair burst from a server caching the recent data including key information via RTCP toolkit.

Data the RTP receiver needs to get from the retransmission server:

1. Retransmission request
2. Unicast repair including key information
3. Join
Introduction

• RTP receiver says to the retransmission server:
  “I have no synch with the stream. Send me a repair burst that will get me back on the track with the multicast session”

• Differences compared to conventional retransmission:
  – Receiver does not know exactly what it is missing
  – Retransmission server
    • May need to parse data from earlier in the stream than it is needed for retransmission (Key information may be dispersed)
    • May need to burst faster than real time
    • Needs to coordinate the time for multicast join and ending the burst

• We define a method that enables a multicast receiver to acquire and process a multicast flow quickly

• The method is applicable to any RTP-encapsulated multicast flow
Changes since Version -00

• Separation of payload-independent and payload-specific messages:
  – The core of the method is now payload-independent
  – The draft may still define payload-specific messages for MPEG2-TS
    • Should it be documented in a new draft?

• Defining RTCP feedback messages:
  – RTCP APP packets have been removed
  – RTCP transport-layer/payload-specific feedback messages are defined

• Reporting rapid synchronization performance:
  – A new RTCP XR (Multicast Join) report is proposed
    • Should it be documented in a new draft?
Rapid Synchronization

--- Rapid Synch Request & Additional Requirements

--- Burst Description

--- Multicast Flows and IGMP Messages

--- Unicast Flows and RTCP Messages
RMS Request (PT=RTPFB, FMT=2)
(Note that FMT=2 is already taken by RFC 5104)

- Min RMS Buffer Fill Req: RR’s minimum data req (in ms) from the burst
  - A zero value means it is not specified

- Max RMS Buffer Fill Req: Maximum data (in ms) RR can accept from the burst
  - A zero value means it is not specified

- Max Receive Bitrate: Maximum bitrate (in bps) that RR can receive
  - A zero value means it is not specified
RMS Information (PT=RTPFB, FMT=3)  
(Note that FMT=3 is already taken by RFC 5104)

• Message sent from RS to RR including several optional information fields indicating:
  – Acceptance of RMS request
  – Extended seqnum of the 1\textsuperscript{st} RTP packet in the unicast burst
  – Indication of the (earliest) multicast join time
    • In time units with reference to the arrival of the 1\textsuperscript{st} RTP repair packet
    • W/o reference – Message triggers RR to join multicast session immediately

• Exact message format is still TBD and two formats are available:
  – RS determines a priori the size of the burst and multicast join time
  – RS signals in real-time when RR should join multicast session
RMS Termination (PT=RTPFB, FMT=4)
(Note that FMT=4 is already taken by RFC 5104)

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+------------------------------------------------------------------+
| RTP Extended Seqnum of the First Received Multicast Packet       |
+------------------------------------------------------------------+
```

- This message can be sent by RR when it joined multicast
  - If RS calculated in advance when it needed to stop bursting, message can be omitted
- If RS did not signal in advance when bursting would finish
  - RR lets RS know the seqnum of the first multicast packet
  - RS then decides (when) to stop bursting
- If RR needs to cancel an active/pending burst, RR sends BYE(s) and RS stops bursting
# Multicast Join (XR) Report Block

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>++++++++++++++++++++++++++++++++++++++++++++++++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BT</td>
<td>rsvd.</td>
<td>Status</td>
<td>Block Length</td>
</tr>
<tr>
<td>++++++++++++++++++++++++++++++++++++++++++++++++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSRC of the Multicast Session</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>++++++++++++++++++++++++++++++++++++++++++++++++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTP Extended Seqnum of the First Received Multicast Packet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>++++++++++++++++++++++++++++++++++++++++++++++++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGMP Join Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>++++++++++++++++++++++++++++++++++++++++++++++++++++</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- May indicate successful completion of multicast join
- Useful diagnostics information
- Other fields are TBD
Issues in Control Plane

• RMS Request
  – No major issues (Requirements are optional to send)

• RMS Information
  – Two encoding options
    • Option A: Full picture of what the burst will provide
    • Option B: On-the-fly IGMP join-time updates
  – Idea is to converge to single encoding format with flexible usage

• RMS Termination
  – No major issues (This message is optional to send)
Do We Need Redundancy in CP?

- **RMS Request**
  - If lost, RR times out and joins multicast

- **RMS Information**
  - If lost, RR times out and joins multicast
  - Should we attempt a retry if RMS-I is lost?

- **Payload-Specific Message (e.g., PAT/CAT/PMT in MPEG2-TS)**
  - If lost, RR times out and joins multicast
    - (In some cases, a payload-specific message may not be required)

→ With the loss of any of the above:
  - RR gives up
  - RR joins multicast

- **RMS Termination**
  - If burst does not stop, RR keeps sending “stop” message
SDP Example

```
a=group:FID 1 2
m=video 41000 RTP/AVPF 98
i=Primary Source Stream
c=IN IP4 224.1.1.2/255
a=source-filter: incl IN IP4 224.1.1.2 8.166.1.1
a=recvonly
a=rtpmap:98 MP2T/90000
a=rtcp:41001 IN IP4 9.30.30.1 ➔ Address for the feedback target
a=rtcp-fb:98 nack ➔ Retransmission support (RFC 4585/4588)
a=rtcp-fb:98 nack ssl ➔ Rapid synch support (RFC 4585/4588)
a=ssrc:123321 cname:iptv-ch32@rms.example.com
a=rtcp-xr:multicast-join ➔ XR support
a=mid:1
m=video 41002 RTP/AVPF 99
i=Unicast Retransmission Stream
c=IN IP4 9.30.30.1
a=recvonly
a=rtpmap:99 rtx/90000
a=rtcp:41003
a=fmtp:99 apt=98
a=fmtp:99 rtx-time=5000
a=mid:2
```
Open Source Implementation

Web Access:
http://www.cisco.com/en/US/docs/video/cds/cda/vqe/3_0/user/guide/ch1_over.html

FTP Access:
ftp://ftpeng.cisco.com/ftp/vqec/
Next Steps

• Shall we add a milestone to produce an RFC on rapid synchronization to AVT's charter?