RTP Subsessions

draft-ejzak-avtcore-rtp-subsessions

Richard Ejzak
IETF 84 avtcore
2 August 2012
Problem statement

• How can we provide differential QoS treatment to media streams of different media types that are multiplexed (using BUNDLE) on the same five-tuple?

• Why? To be able to provide differential treatment without losing the advantages of bundling (significant reduction of ICE and DTLS signaling).
Assumptions

- Focus on QoS in access network
  - Access network the most critical portion of end-to-end path for many use cases
    - E.g., wireless
- Consistent with marking-based QoS
- No dependence on markings received from network
- Application is able to request access network QoS
- Application is able to select markings assigned to media streams if necessary to ensure that –
- Packets with different markings are handled by independent queues on path between browser and access node (e.g., wireless device or modem)
  - Assumption consistent with architecture of many mobile devices (e.g., smartphones)
  - Problematic in many existing home networks but this is a problem that needs to be solved anyway
Simplified LTE QoS example

A policy function installs dedicated bearers and traffic flow templates (TFTs). Applications requests QoS-enabled streams through a QoS API. Optionally, DPI identifies and remarks packets. TFTs (packet filters) assign packets to dedicated bearers.

Applications request browser to assign packet markings to QoS-enabled streams to assure correct treatment between browser and access node.

Controllable bearer attributes include: GBR, MBR, packet error rate, transfer delay, priority, IP header compression.

TFT attributes include: Five-tuple and TOS (DSCP), RTP SSRC could be added (future).

TFTs assign packets to dedicated bearers.
Remaining issues

• Network policy requires the network to identify packet flows to determine packet markings and dedicated bearer assignments
  – i.e., don’t trust application or network markings

  ➢ Since BUNDLE multiplexes different flows onto a single five-tuple, the network must use RTP information to accomplish this

• A bundled RTP session may concatenate RTCP packets for RTP streams with different QoS, making it likely that RTCP information for an RTP stream will be assigned different QoS

  ➢ Need a way to assure that an RTCP packet receives QoS as good or better than the corresponding RTP
Proposed solution

- **RTP subsessions:**
  - Each bundled m-line is allocated an RTP subsession
  - Pre-allocate range of SSRC values for use by each RTP subsession endpoint
  - Use SDP attribute to specify SSRC prefix per m-line
  - 128 subsessions can share the same 5-tuple
  - Network uses SSRC prefix in TFT to identify packets for QoS treatment
  - No change to RTP or RTCP message formats
  - Reuse RTP session procedures on each RTP subsession
    - In particular, only RTCP packets from a single RTP subsession are concatenated
  - SSRC reservation avoids collisions
    - More consistent with features that need to identify the contents of individual RTP streams by reserving SSRCs (proposed in both rtcweb and clue)
  - Fully consistent with non-relay topologies (independent SSRC assignment per link in end systems and RTP mixers)
  - Relay (translator) topologies supported between all systems supporting RTP subsessions
    - Legacy interworking provided via SSRC mapping (RTP mixer)
Use in RTP subsession relay topologies

Non-relay

Master relay

Slave relay

Non-relay

Slave relay

Non-relay

Master relay

Slave relay

Non-relay

Non-relay

Slave relay

Non-relay

Non-relay

Non-relay

Non-relay

Non-relay
## SSRC bit assignments

<table>
<thead>
<tr>
<th>Bit Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 1:</td>
<td>identifies direction of flow</td>
</tr>
<tr>
<td>Bits 2-8:</td>
<td>subsession id (for QoS)</td>
</tr>
<tr>
<td>Bits 9-16:</td>
<td>allocated by primary relay to slave relays (except one value reserved for self)</td>
</tr>
<tr>
<td>Bits 17-24:</td>
<td>allocated by slave relays to non-relay systems</td>
</tr>
<tr>
<td>Bits 25-32:</td>
<td>allocated by non-relay systems to individual RTP streams</td>
</tr>
</tbody>
</table>
SDP attribute examples

a=ssrc-prefix non-relay 0x111111

a=ssrc-prefix relay 0x222222

- Media level attribute
- “non-relay” or “relay” and whether attribute is in an SDP offer or answer together indicate role in negotiation and identify use of SSRC prefix bits
- “master relay” always initiates SDP offer (so distinction between master and slave is implicit)
- Hex value in attribute specifies first 24 bits of SSRC
- SDP offer/answer negotiation details in draft
Summary

- RTP subsessions provides an effective solution to:
  - Allow an access network, under request from an application and consistent with network policy, to identify bundled media flows for differential treatment, assign them to appropriate bearers, and remark them.
  - Ensure that RTCP packets get the same treatment as the corresponding RTP streams.
  - Consistent with new applications that use SSRC values to identify individual streams (i.e., rtcweb and clue).
  - Consistent with non-relay and new relay topologies.
  - Reuse RTP/RTCP formats and procedures (modified).
Comparison to SHIM

- RTP subsessions and SHIM address different problems with bundling

<table>
<thead>
<tr>
<th>SHIM</th>
<th>RTP subsessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preserves independent RTP session and SSRC space per m-line</td>
<td>Partitions single RTP sessions between m-lines by allocating SSRC ranges</td>
</tr>
<tr>
<td>Poor support for differential treatment of bundled RTP sessions (SHIM header at end of packet)</td>
<td>Good support for differential treatment of RTP subsessions based on SSRC prefix filtering</td>
</tr>
<tr>
<td>Changes RTP/RTCP message formats</td>
<td>No changes to RTP/RTCP message formats</td>
</tr>
</tbody>
</table>
Next steps?