

Monitoring Architectures for RTP

draft-hunt-avtcore-monarch-02

G. Hunt (geoff.hunt@bt.com)

P. Arden (philip.arden@bt.com)

Qin Wu (sunseawq@huawei.com)

Outline

- History
- Changes
- Issues
- Follow Up

History

- “Submit Monitoring Architecture for RTP for Informational” is an original AVT goal for Feb 2011
- Early version of draft-hunt-avt-monarch is prepared by Geoff Hunt and focus on narrow scope of how to construct new RTCP XR block.
- It was presented in IETF 78 by Roni.
- The new version is ready after AVT WG split and include some inputs received.

Changes since avt-monarch-02

- Propose a new RTP monitoring architecture.
 - Two components (Monitor, Metric Block)
 - Should we add monitor as an RFC 3550 building block similar to mixer, translator entities?
- Monitor as defined in avt-monarch-02
 - Monitor is a functional component that acts as a source of information gathered for monitoring purposes. It may also collect statistics from multiple source, stores such information reported by RTCP XR or other RTCP extension appropriately as base metric or calculates composite metric. The end system that source RTP streams, or an intermediate-system that forwards RTP packets to End-devices can be envisioned to act as Monitor within the RTP monitoring architecture.

Changes since avt-monarch-02

- RTCP metric parameters classification.
 - Transport metric
 - Application metric
 - End system metric
- Specify monitoring methodology
 - carry raw arrival data in RTCP, Histogram Summaries, reporting following transport exceptions?

Changes since avt-monarch-02

- Merging the sections on using RTCP XR by using small blocks, identity block, Expanding the RTCP XR block namespace to guideline section.
- Some restructuring by moving Applicability to MCU and Applicability to translator to one section.
- Discuss Issues with RTCP XR extension
 - some of these block types defined in [[RFC3611](#)] are application specific
 - Namespace restriction
 - Identification information repetition

Issue- Block namespace restriction

- Until now block types 12 to 254 are unassigned and 255 is reserved for future expansion.
 - Seven block types are allocated to RFC3611
 - Block type 8 go to RFC5093
 - Block type 10 go to RFC5725
 - Block type 11 go to [[I.D-ietf-avt-multicast-acq-rtcp-xr-01](#)]
 - [[I.D-ietf-avt-ecn-for-rtp-02](#)] and more and more XR drafts are waiting for Block Type allocation.
- Is there a consensus for, or against, work to allow such namespace expansion ?
- Do we need to reserve a subset of Block type for other SDO or vendor use?

Issue-Identity information repetition vs blocks association

- An RTCP XR packet containing four metrics blocks, reporting on streams from two sources. How to associate identity information with relevant metric block?
- If the same identity information is included in each metric block, we will see repetition of identification information in multiple blocks in the same RTCP XR packet.
- Suggestions
 - Reduce overhead to carry duplicated data for measurement identity in all these metric block
 - Separate measurement identity information from metric block and form independent block.
- Proposals:
 - Define two identity blocks, metrics blocks associated with an identity block must always follow the identity block ?
 - Define two identity blocks, allocate one tag field in the identity block and metric block to associate metrics blocks ?

Issue-Monitor Declaration

- An RTP translator presence according to RFC 3550 is transparent to receivers.
- RTP translator may collocate with Monitor.
- Monitor will use its own SSRC to send RTCP XR packet and will not send RTP packet.
- RTP translators with co-located monitors can be detected by knowing that sending SSRC is not present in any RTP media packet. However ambiguity may arise when
 - For bidirectional unicast, when a source sends RTCP before it has sent any RTP media, an RTP end system or RTP mixer may be transiently misunderstood as a translator with monitoring
 - for multicast sessions – or unidirectional/streaming unicast - there is a possibility of a receive-only end system being permanently mis-categorised as a translator with monitoring
- Question:
 - This question was asked in the original document from Geoff Hunt.
 - Is this relevant to Monitoring architecture?
 - Do we need to declare explicitly Monitor in the SDP or using other ways?

Follow Up

- Address the issues discussed in this meeting
 - Any inputs will be welcome!
- Version 02 was sent to the list and will be available in AVTcore when submission is re-opened.
- Request to accept this draft as WG item