MULTIPLE RTP SESSIONS OVER ONE TRANSPORT

draft-westerlund-avtcore-transport-multiplexing

Magnus Westerlund, Ericsson
Colin Perkins, University of Glasgow
OUTLINE

› Goals
› Motivation
› Overview
› Open Issues
  - SHIM location
  - SHIM Size
  - Signaling fallback
  - In-band Keying
› Next Steps
GOALS

› Create a WG item for multiple RTP sessions over one transport flow

› Resolve the main open issues
MOTIVATION

› The classic One RTP session per transport flow
› Proposed multiple media types in one RTP session
› Multiple RTP sessions over one transport flow is needed:
   - NAT/FW traversal simplification is still main reason
   - Application doesn’t need transport level separation
   - Some applications desire multiple RTP sessions
     › Logical separation of type of flows
     › Tailored RTP/RTCP Extensions to RTP session content
   - Enable mixed (single transport flow and not) multi-party sessions without special translation
   - Usage of XOR FEC etc that require multiple sessions
› We have no solution for multiple sessions over one transport
   - Bias against using multiple RTP sessions
SHIM LOCATION

› In the stack processing the SHIM is between the RTP/RTCP and the transport layer (UDP)
› However, the Session ID value in the SHIM can be located either at the end of the RTP/RTCP packet or in the beginning
› The location affects the solutions property
Enables aware network nodes to process the SHIM header

- Enables middleboxes that only process headers to add or strip the SHIM

- Cause the packet to not look like regular RTP/RTCP
  - Potential deployment issues

- Reduced efficiency in header compression (IP/UDP only)

- Third party monitors can easily take SHIM into account

- Long term better solution
Any network node doing processing on header only can’t read the SHIM header
  − Makes future stream aware processing impossible

Makes packet look like standard RTP

Multiple flows with same SSRC (different Session IDs)
  − Jumping field values from a non SHIM reading entity
    › Header compression will work poorly
    › Stateful Firewalls may block transport flow

Likely easier initial deployment, long term worse
Proposes that the SHIM is pre-fixed

- IP
- UDP
- SHIM
- RTP/RTCP
SHIM SIZE

› What is the appropriate size of the SHIM header:
  - 1 byte
  - 2 byte
  - 4 bytes

› From number of Session IDs 256 identifiers are more than sufficient:
  - Supports 128 to 256 RTP sessions depending on RTCP mux or not

› Primarily Question of alignment vs. overhead

› Any fixed ID in first byte to enable separation from STUN etc?
Latest version does not use BUNDLE
Reason is to enable better fallback
SHIM users prefer RTP sessions
  – Fallback to Individual flows
Using Bundle with extensions
  – Fallback would result in Single Session if peer supports Bundle but not SHIM
SHIM will use a small bounded set of the BUNDLE rules regarding transport parameters
  – Rest is RTP session specific
IN-BAND KEYING

› In-band keying mechanisms needs to provide unique keys to each SRTP session
› Thus DTLS-SRTP and ZRTP would need to be run also within the context of a specific Session ID(s)
› Running DTLS-SRTP for each Session ID
  – Additional overhead
  – Session resumption can be used if delay is not an issue
› Means that SHIM implementation needs to be integrated with STUN/ICE and DTLS-SRTP
› Alternatives?
NEXT STEPS

› We will update the draft based on feedback

› Does the WG support the creation of a WG item:
  – To define a method to provide multiple RTP sessions over one transport flow

› Does the WG want to adopt this document?