RTCP for Inter-Destination Media Synchronization (IDMS)

draft-ietf-avtcore-idms-02

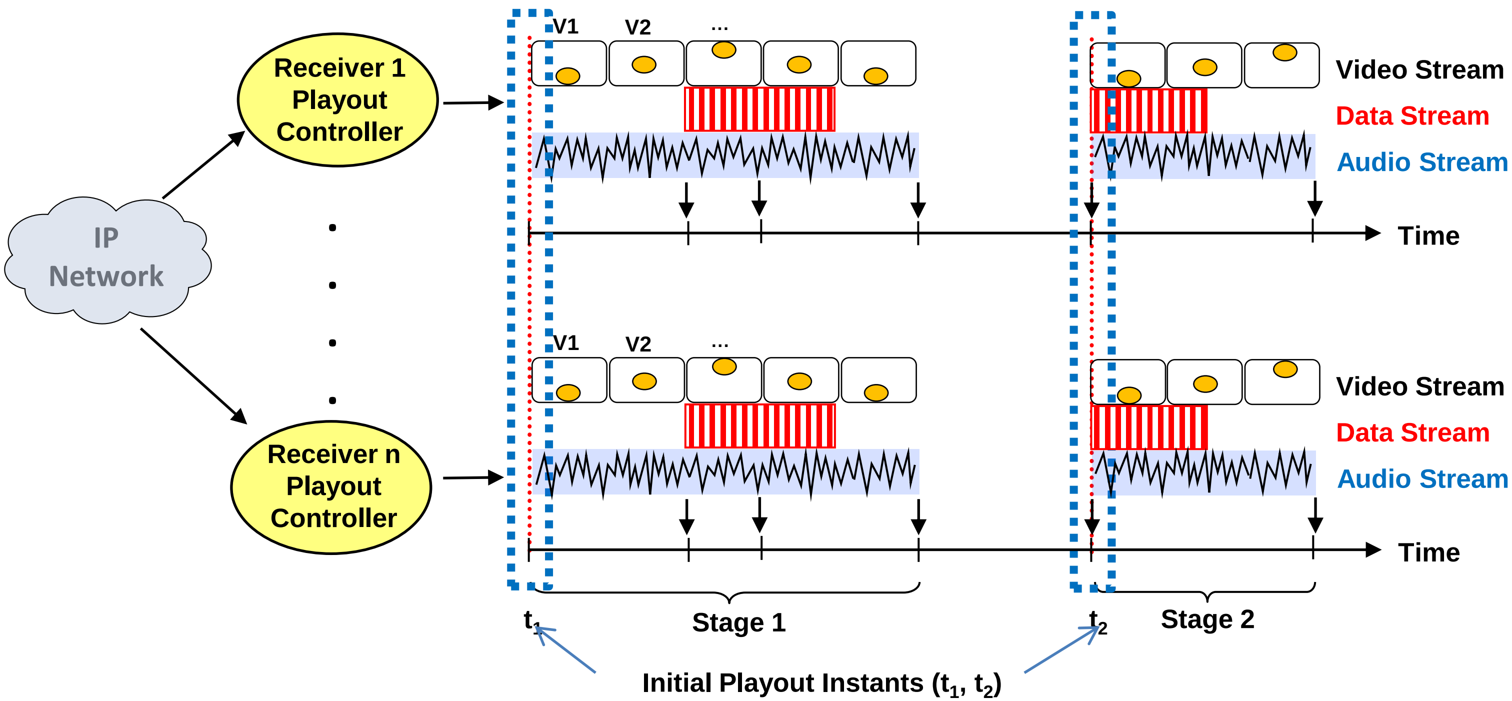
Ray van Brandenburg, TNO
Hans Stokking, TNO
Oskar van Deventer, TNO
Fernando Boronat, Universidad Politecnica de Valencia
Mario Montagud, Universidad Politecnica de Valencia
Kevin Gross, AVA Networks

Recent work on the draft

- Broadened scope of IDMS to include
  - Video walls
  - Networked speakers
  - Phased array transducers
- Increased the presentation timestamp to 64bit
  - Allows for high-accuracy use cases such as Networked Loudspeakers
- Introduced recommendations for leap second issue
  - Might also be interesting for non-IDMS RTP use
- Discussion on Initial Playout Synchronization

Initial Playout Synchronization

- Current IDMS draft provides a solution for synchronization <u>during</u> playout
- Initial playout synchronization solves problem of finding a common starting time



How to find one-way delay?

- RTT/2 is considered to not be accurate enough
  - Delay might not be symetric
  - DLRR Report block cannot be used
- Three options have been proposed
  - Use existing IDMS reports before streaming starts
  - Start streaming with null data and use existing IDMS reports
  - Create new XR block to report on one-way (sender->receiver) delay

Proposal 1: Use existing IDMS reports before streaming starts

1) Before starting streaming, the MSAS sends one or more consecutive RTCP IDMS packets
  - Not referencing any RTP packets (since none have been sent, RTP Timestamp field is set to 0)
  - Including the NTP time at moment of sending (using the packet reception timestamp field)

2) Once Synchronization Client receives packet, it sends back an IDMS XR block
  - Copying the NTP time from RTCP IDMS packet (using the packet presentation field)
    - Allows MSAS to correlate IDMS XR block with RTCP IDMS packet
  - Including the NTP time at moment of reception of RTCP IDMS packet (using the packet reception timestamp field)

3) MSAS can calculate the one-way delay between MSAS and SC.

- Advantages:
  - No new packet types have to be defined
- Drawbacks:
  - Does not allow for reporting on delay between packet reception and packet presentation (depends on actual data)
  - Delay between RTP sender and RTP receiver might depend on packet size

Proposal 2: Start streaming with null data and use IDMS reports

- The RTP Sender starts streaming RTP packets containing null media data
- Synchronization Clients report on these packets just as they would with normal RTP packets (i.e. by sending IDMS XR blocks)
- The MSAS receives the IDMS XR blocks and is able to calculate the delay to the various senders
- Once the delay has been calculated, the MSAS sends an RTCP IDMS control packet and starts streaming actual media data

- Advantages:
  - No new packet types have to be defined
  - Delays are based on actual media data packets (instead of ‘small’ rtcp packets)
  - Allows for reporting on delay between packet reception and packet presentation

- Drawbacks:
  - Requires extra (null) data to be sent, increasing overhead

Proposal 3: Create new XR block to report on one-way delay

- Proposed by Qin Wu:
  - Create new XR block for reporting on one-way delay between RTP sender and receiver

- Advantages:
  - New XR block might also be used for other purposes than IDMS

- Drawbacks:
  - New packets have to be defined
  - Does not allow for reporting on delay between packet reception and packet presentation (depends on actual data)
  - Delay between RTP sender and RTP receiver might depend on packet size

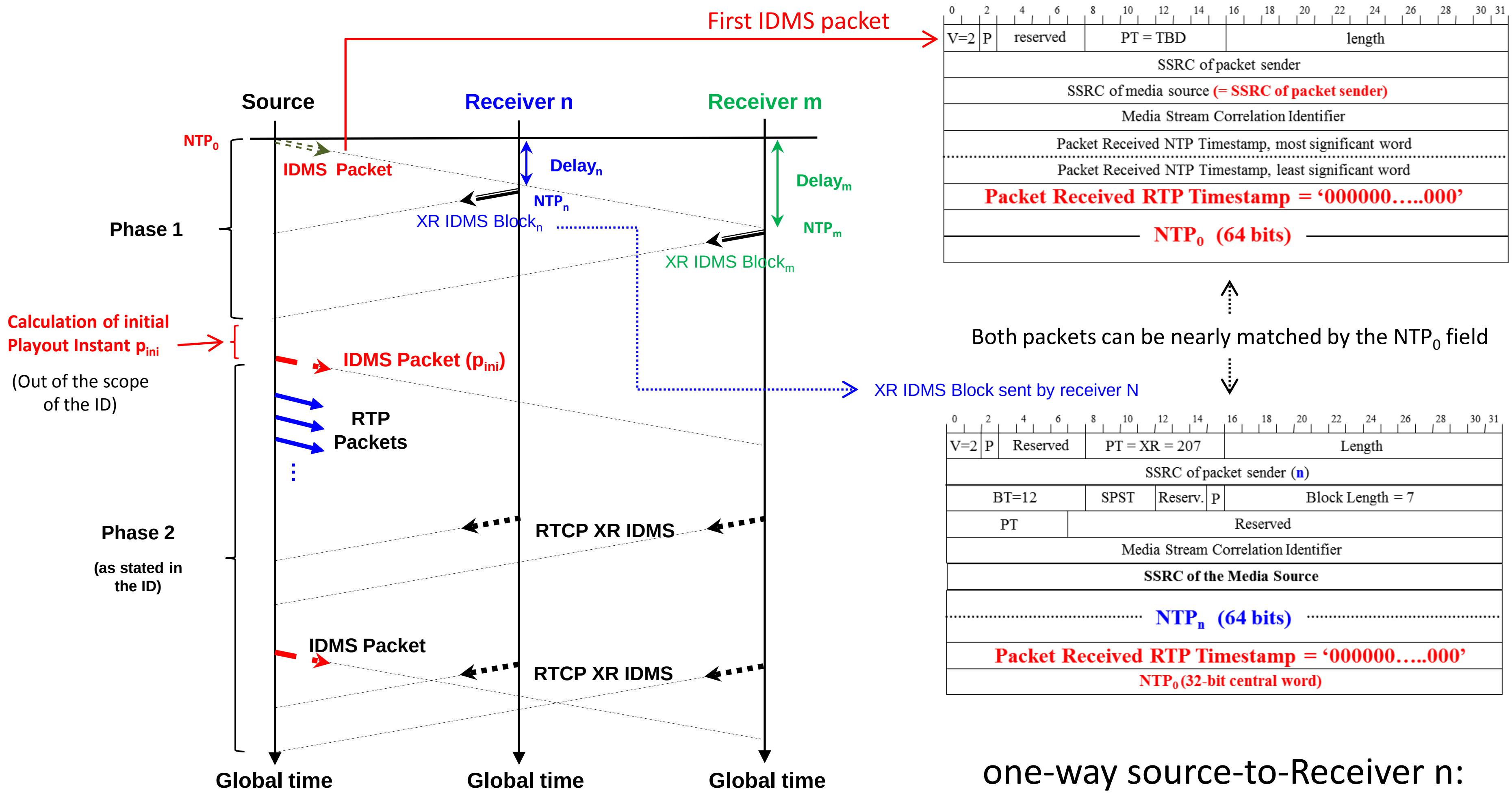
Next steps

- Decide on solution for calculating one-way delay
- Milestone for IDMS set for March 2012

Backup slides

Solution 1 – Preliminar phase with control packet exchange

Example with draft packets (other control packets defined in other drafts could also be used e.g. the new ones in Qin Wu proposed draft about delay metrics)



one-way source-to-Receiver n:

$$Delay_n = NTP_n - NTP_0$$

All the receivers will start the playout at P_{ini}

Additional help:

First IDMS packet

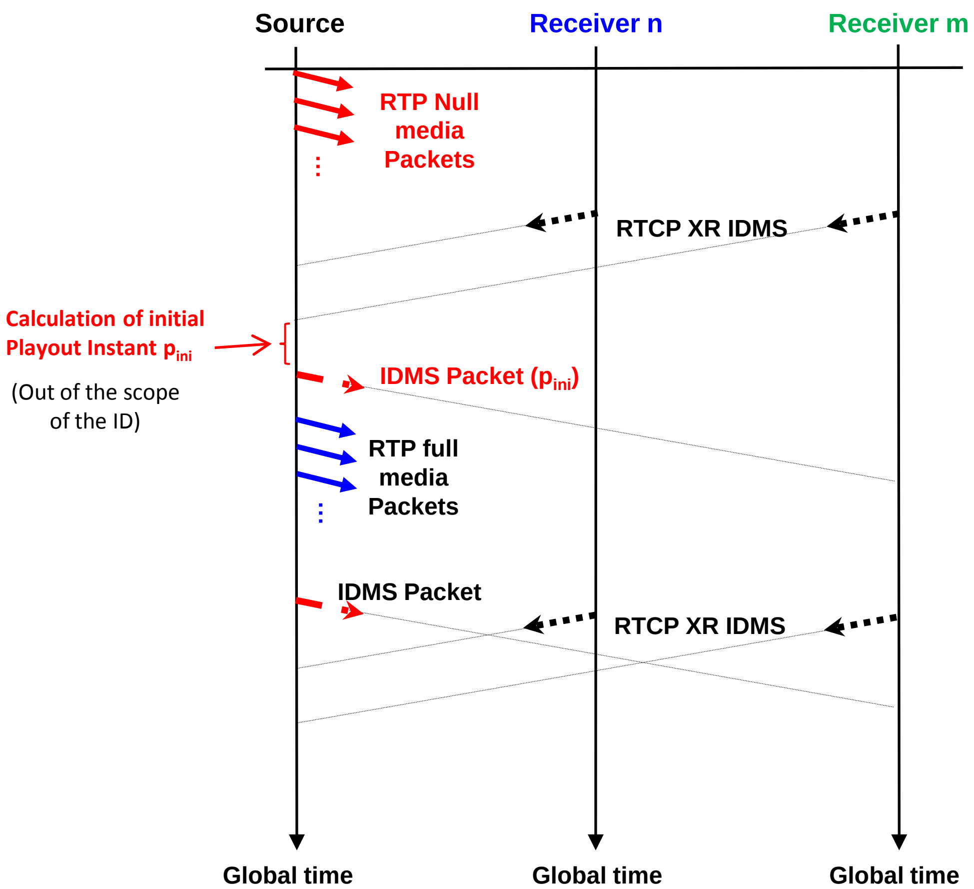
0–1	2	3–7	8–15	16–31
V=2	P	reserved	PT = TBD	length
SSRC of packet sender				
SSRC of media source (= SSRC of packet sender)				
Media Stream Correlation Identifier				
Packet Received NTP Timestamp, most significant word				
Packet Received NTP Timestamp, least significant word				
Packet Received RTP Timestamp = ‘000000…..000’				
NTP_0 (64 bits)				

XR IDMS Block sent by receiver N

0–1	2	3–7	8–15	16–31
V=2	P	Reserved	PT = XR = 207	Length
SSRC of packet sender (**n**)				
BT=12			SPST / Reserv. / P	Block Length = 7
PT		Reserved		
Media Stream Correlation Identifier				
SSRC of the Media Source				
NTP_n (64 bits)				
Packet Received RTP Timestamp = ‘000000…..000’				
NTP_0 (32-bit central word)				

Possible response to kevin’s comment: Both packets can be nearly matched by the SSRC of the Media Source and NTP_0 fields (we understand than the different bit length of the fields in both packets is not a problem to match them). Source should send several First IDMS packets (e.g. in periodic intervals) in order to avoid packet looses and to obtain mean one-way delay values.

Solution 2 – Sending Null media packets at the beginning



All the receivers will start the playout of full media units at P_{ini}