

# Session Multiplexing for SVC Video – draft-hannuksela-avt-rtp-svc-01

Miska M. Hannuksela and Ye-Kui Wang, Nokia  
miska.hannuksela@nokia.com, ye-kui.wang@nokia.com

28 July 2008, Dublin, AVT@72<sup>nd</sup> IETF

# Outline

- Concepts
- Sender and receiver operations
- Example

# Concepts

- Goal: Decoding order recovery for multi-session SVC transmission
- The design in this draft
  - Avoids the use of cross-session decoding order number (CS-DON) for better compatibility with the single-session non-interleaved mode
    - Same as the 3 alternative modes (NI-T, NI-C and NI-TC) in the SVC draft
  - Keeps the specification and sender & receiver operations simple, e.g., no new NAL unit types
    - If one of the two alternatives is used to replace the NI-T, NI-C and NI-TC modes
- The two alternatives in this draft
  - Access-unit-identifier-based mode (NI-A)
    - Access unit identifier (AUID): Remains the same for all NAL units of one access unit, and MUST differ in consecutive access units
  - Timestamp-difference-based mode (NI-TSD)

# Sender and receiver operations

## Sender/transmitter operation

- Indicates the previous access unit in decoding order in this session or any lower session with its AUID or timestamp difference (between this and the previous access unit)
  - AUID/previous AUID or timestamp difference, instead of DONC, carried in the PACSI NAL unit or in the FU-B NAL unit
  - No change in syntax structures, only small changes in semantics
- In NI-TSD, random initial timestamp offset is the same in all associated sessions
  - Or, timestamps should be “normalized” between sessions through RTCP SRs

## Finding of the next access unit in decoding order recovery in receiver

1. Collect the candidate access units that come next in decoding/transmission order in each of all the sessions
2. One or more access units among the candidate access units are indicated as the next access unit. The one from the highest session is chosen.

# Example – pure temporal scalability

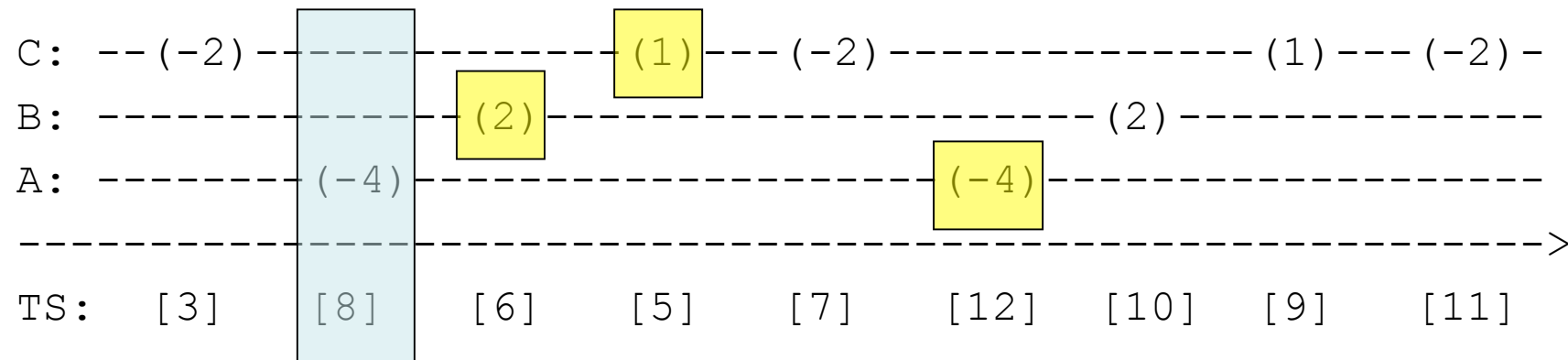
C: -- (-2) ----- (1) --- (-2) ----- (1) --- (-2) -  
B: ----- (2) ----- (2) -----  
A: ----- (-4) ----- (-4) -----  
----->  
TS: [3] [8] [6] [5] [7] [12] [10] [9] [11]

`( )' - (timestamp difference)

`[]' - [timestamp]

--> - transmission order with one session

# Example – pure temporal scalability



`( )' - (timestamp difference)

`[]' - [timestamp]

--> - transmission order with one session