

# Decoding Order Recovery for Multi Flow Transmission of Scalable Video Coding (SVC) over mobile IP channels

Thomas Schierl<sup>1</sup>, Yago Sanchez de la Fuente<sup>1</sup>, Cornelius Hellge<sup>2,1</sup> and Thomas Wiegand<sup>1,2</sup>

<sup>1</sup>Fraunhofer HHI  
Multimedia Networking/Image Processing Department  
10587 Berlin, Germany

<sup>2</sup>Technische Universität Berlin  
Image Communication Group/Dept. Tele. Com. Sys.  
10587 Berlin, Germany

## Abstract

Recent standardization activities for the transport of Scalable Video Coding over multimedia networking channels have introduced means to differentiate layers of a scalable media in transport. Therefore in RTP, those layers are transported in different RTP flows respectively RTP sessions. To decode such layers transported on different IP network flows, a decoding order recovery mechanism is required that reorders the media data of the layers to decoding order at the receiver or at an intelligent network element. Since today's multimedia networking channels typically provide presentation timestamps on the transport layer, it was obvious to rely on those data for decoding order recovery. Due to the fact that presentation timestamps do not appear in the media stream in increasing order, an algorithm purely based on presentation timestamps may fail in some scenarios where temporal scalability is used. Another approach to solve the problem is to use a numbering mechanism in the bit stream giving the decoding order. In this paper, we highlight both mechanisms as defined in the RTP payload format for SVC video and compare them in loss scenarios typically present in mobile broadcast channels. We show that the presentation timestamp based solution gives over a wide range of loss scenarios almost the same results in terms of decoded video quality as the numbering based approach.

## Keywords

SVC, RTP, multi session transmission, decoding order recovery, timestamp based, mobile broadcast channels.

## 1. Introduction

Recent media transport standardization activities as the specification of the RTP payload format for SVC video [1] [2] include means for differentiated media transport. For a layered media codec as SVC, layers or set of layers are transferred in different network flows, which may get different transport layer protection. Such a differentiated handling of layered media transport is already known from Layered Multicast [3], where according to the connection condition a Multicast receiver joins or leaves Multicast sessions carrying layers of a scalable bit stream. Other approaches extended Layered Multicast to today's IP-based mobile multicast/broadcast channels as 3GPP's MBMS as well as DVB's H, NH and SH services. Differentiation in transport may also make sense in combination with IP Differentiated Service (DiffServ – IETF RFC 2475).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
Mobimedia'09, September 7-9, 2009, London, UK.  
Copyright 2009 ICST 978-963-9799-62-2/00/0004 ... \$5.00.

In order to receive data from multiple network flows, a mechanism is required to reconstruct the decoding order of media data received from different network flows. This is due to the fact that IP flows and packets in particular may take different paths over the Internet and therefore may arrive at the receiver in different order. RTP [4] itself provides a mechanism to re-order received packets to transmission respectively decoding order within a single flow using RTP sequence numbers. But RTP is lacking any means to re-order data from different flows. In RTP, different flows may be a result of different multiplexing techniques such as RTP session multiplexing which relies on the use of different transport addresses (such as network address and/or port) as well as Synchronization Source (SSRC) Multiplexing [4] which relies on a field in the RTP header indicating the source of the RTP stream.

In order to address the issue of re-ordering data of different flows to decoding order, two different means have been standardized in the RTP payload format for SVC [1]. One method relies on RTP presentation timestamps and other already existing RTP header information and is therefore called the timestamp-based (*TS-based*) decoding order recovery (DOR), in [1] NI-T mode. The other method is based on an additional numbering as similarly introduced in [5] and is therefore called the cross-session decoding order number-based (*CS-DON-based*) DOR, in [1] NI-C mode.

Different concerns have been raised in the standardization process of [1] that the CS-DON-based method outperforms the TS-based method for the following reasons: Since the RTP timestamps are presentation timestamps and do not linearly increase with the decoding order, TS-based DOR may not be able to correctly recover the decoding order in all packet loss cases.

Therefore, we evaluate both decoding order recovery methods over a simulated IP-based mobile broadcast channel with burst loss characteristics and provide detailed results in terms of received video quality for random as well as burst losses as usual for mobile broadcast/multicast channels as DVB-H, DVB-SH or 3GPP MBMS.

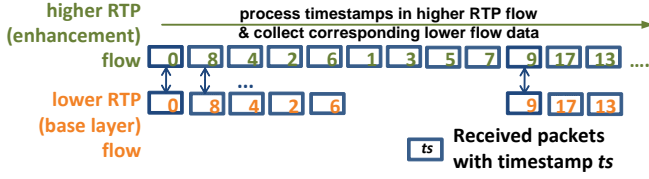
In the next section we explain the two decoding order recovery mechanisms and detail the problem. In section 2.1, we present the used simulation testbed and we present the results in section 4 and conclude in section 5.

## 2. Decoding Order Recovery (DOR)

### 2.1 Timestamp (TS)-based DOR

The RTP payload format for SVC defines an algorithm to reorder the SVC NAL units (video packets) received from multiple flows using only information already present in the RTP, i.e. the RTP timestamps to align data between flows, the RTP sequence numbers to re-order data within one flow as well as the

knowledge about the dependencies of the media in the different RTP flows to place the media data from the flows in decoding order into the received bit stream.



**Figure 1 – Timestamp-based decoding order recovery**

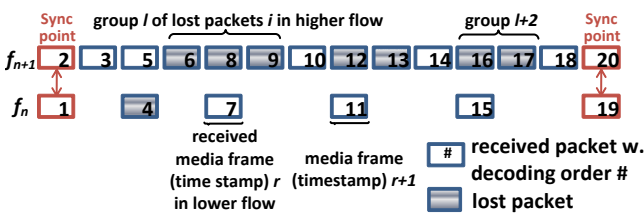
Furthermore, this mode relies on the knowledge that in each depending *higher RTP flow* data for a particular timestamp is present, if such timestamp is also present in the flow the depending flow depends on (the *lower RTP flow*) [1]. Figure 1 shows the principle of the algorithm. Two RTP flows are shown containing two layers of a SVC bit stream, where the base layer has a lower temporal resolution than the enhancement layer.

The algorithm proceeds as follows with the assumption that data within each RTP flows is already re-ordered to RTP sequence number order (i.e. decoding order):

- Identify first timestamp available in all RTP flows and call it *target timestamp* and proceed to target timestamp in the *highest RTP flow* (the flow in the hierarchical SVC layer dependency, which no other flow depends on; the *lowest RTP flow* is the flow containing the base layer)
- Collect all media data in the different RTP flows with the same target timestamp.
- Reorder media data according to (RTP sequence number and) dependency order of flows to access unit order of target timestamp.
- Place access unit into outgoing bit stream.
- Proceed to next timestamp in highest flow, call it target timestamp and continue at b.)

This algorithm is applied straight forward, if there are no losses in the flows.

If there are losses in the RTP flows, the situation is different. Since the algorithm relies on the appearance of timestamps in the highest RTP flow to recover the decoding order of the media data, some packet loss constellations may not allow for correct decoding order recovery. For this reason, it is proposed in [1] to stay in the lowest RTP flow for which a receiver can safely recover the decoding order. In a two session scenario as shown in Figure 1 the decoding order can be always safely recovered in the lowest session respectively the base layer assuming that one layer is sent per flow.



**Figure 2 – Loss example within an out-of-sync interval**

The example in Figure 2 shows loss cases, which may lead to uncertainty about the decoding order of the transmitted media packets. Synchronization between the higher and the lower RTP

flow can be lost if corresponding packets in the two flows are lost for a given timestamp, e.g. in Figure 2 it is not clear if packet #7 (and #11, #15) appears in decoding order before packet #10 or even directly before packet #18. The interval of packets between two correctly received *synchronization points* in lower and higher RTP flow is called an *out-of-sync interval*.

For simplification, we evaluate only the case of two flows in the following, since the problem statement of using more than two RTP flows can be always reduced to two RTP flows, where the lower RTP flow is the highest RTP flow up to which there is no uncertainty in decoding order. For formulating the problem statement, we define the following terms for an out-of-sync interval:

- $f_n$ : identifies the RTP flow, where  $f_0$  indicates the RTP flow containing the base layer and  $f_{n+1}$  indicates the RTP flow, which is the next higher RTP flow to  $f_n$  in the coding dependency hierarchy
- $n_r(f_n)$ : the number of media frames  $r$  or parts thereof with different timestamps received in RTP flow  $f_n$ .
- $n_l(f_{n+1})$ : the number of groups  $l$  of consecutive lost packets in the RTP flow  $f_{n+1}$ .
- $n_{l,i}(f_{n+1})$ : number of packets  $i$  within group  $l$  of consecutive lost packets in RTP flow  $f_{n+1}$ .

If the following condition is true, there is no uncertainty in the TS-based DOR process within an out-of-sync interval:

$$\left( n_l(f_{n+1}) < 2 \right) \vee \left( n_l(f_{n+1}) \geq 2 \wedge n_r(f_n) = \sum_{i=1}^{n_l} n_{l,i}(f_{n+1}) \right) \quad (1)$$

Assuming that in an out-of-sync interval, where each lower RTP flow media timestamp misses its counterpart in the higher RTP flow, the equation above gives the condition for successful decoding order recovery. There is no problem, if there is not more than one group of consecutive losses  $n_l$  in the higher RTP flow  $f_{n+1}$ , i.e. the lower RTP flow media frames  $n_r$ , for which there is an uncertainty consequently must have the same timestamps as the group of lost packets in the higher RTP flow  $f_{n+1}$ . This case corresponds to the first condition ( $n_l(f_{n+1}) < 2$ ). Additionally, if there are more than one group of consecutive losses  $n_l$ , the lower RTP flow media frame for which there is an uncertainty could have the same timestamp as any of the groups of lost packets in the higher RTP flow. There would be a unique possible solution for DOR, if the number of packet losses in the higher RTP flow  $f_{n+1}$  and packets  $n_r$  with different timestamps in the lower RTP flow  $f_n$  for which there is an uncertainty would be the same, since each group of packets in flow  $f_n$  with different timestamp must match at least with a single packet in the higher RTP flow  $f_{n+1}$ . This corresponds to the second condition of equation (1).

If there is an uncertainty, the concealment is to go on decoding in the lower RTP flow up to the next synchronization point, where decoding can be continued in the higher RTP flow. The condition in (1) can be extended to more than two RTP flows if applying the following steps for each two RTP flows  $f_n$  and  $f_{n+1}$  (*corresponding RTP flows*), starting from the lowest RTP flow:

Identify the out-of-sync interval for each two corresponding RTP flows  $f_n$  and  $f_{n+1}$  and check for uncertainty in decoding order. If there is already an uncertainty, this will also affect all the higher flows. If there is no uncertainty, the process can continue in the next pair of higher RTP flows  $f_{n+1}$  and  $f_{n+2}$ . The coding structure of the SVC layers in the transport illustration in Figure 1 shows the

presence of different frame rates in base and enhancement layer. Since the rules in [1] for TS-based DOR require the presence of a media time stamp in any higher RTP flow  $f_h$  with  $h > n$ , if present in RTP flow  $f_n$ , the lower RTP flow (base layer) may contain less media frames than the higher flow. This fact may increase the vulnerability of such streams in terms of potential uncertainty in decoding order recovery.

A possible increase in robustness may be achieved extending the aforementioned rule to also require the presence of timestamps in lower RTP flows, if a media sampling instance is present in any higher flow. [1] defines the Empty NAL unit packet which can act as carrier for those media timestamps.

## 2.2 CS-DON-based DOR

The decoding order recovery based on a cross session sequence number/DON as proposed in [5] is different from the TS-based method described in section 2.1 almost straight forward. The receiver can re-order the received media packets only according to the cross session sequence number. The only drawback is the need of adding additional data fields respectively additional packets carrying those numbers according to the protocol.

## 3. Simulation Testbed

In order to simulate and evaluate the TS-based and the CS-DON-based DOR methods, we used a simulator which applies the NAL unit packetization as defined in [1] using the non-interleaved packetization mode and the NI-TC—the non-interleaved TS-based and CS-DON-based mode for decoding order recovery, i.e. both information for the TS-based as well as for the CS DON-based method are present in the RTP flows. For the TS-based DOR, we assume that the RTP flows provide timestamp synchronization information at each random access point in the flows using RTP header extensions as introduced in [6]. The simulator applies a fragmentation on RTP level, using Fragmentation Unit (FU) packets, if required by the maximum transfer unit (MTU) size settings.

All RTP packets are mapped to IP packets and then into MPE—multi protocol encapsulation—packets as, e.g. used by DVB-H. Then the MPE packets are distributed into MPEG-2 Transport Stream blocks (TBs) having a payload of 184 bytes each. In order to simulate a DVB-H channel, we rely on a Gilbert-Elliot (GE) model for modeling burst losses, as e.g. used in [7] for modeling a DVB-H channel. The SVC decoder receiving erroneous streams is applying in the base layer freeze-frame error concealment if reference frames are missing. Furthermore each media frame is decoded up to the highest available layer. Therefore we rely on SVC medium grain scalability (MGS), which allows switching between layers at each frame, for more details we refer to [8]. In case of losses, the simulator determines following equation (1), whether there is a decoding order problem or not. If there is a decoding order problem, the decoder can only rely on data in the lower RTP session, i.e. the base layer.

## 4. Results

For the simulations, we used exemplary the following SVC stream distributed over two RTP flows having a Group of Picture (GOP) size of 16 plus a preceding IDR (intra) picture for each GOP, i.e. the stream has a random access interval of 0.57s. The base layer in Flow 1 is an AVC bit stream at QVGA resolution with 15fps. In Flow 2, there is a temporal enhancement of the base layer to 30 fps as well as a MGS quality enhancement of the QVGA base layer. A rate control has been used to keep the bit rate in a +/-2.5%-window

of the average value per IDR+GOP16 picture chunk. The stream length is about 40 seconds consisting of a concatenation of the ITU-T test sequences City, Crew, Foreman and Soccer:

	const. bit rate[kbps]	avg. PSNR[dB]	frame rate[fps]
Flow 1	123.75	32.00 (at 15fps)	15.0
Flow 2	273.44	36.08	30.0

In the following figures, we show the number of additional lost NAL unit packets in case the decoding order cannot be recovered within an out-of-sync interval using the TS-based method compared to DOR using the CS-DON-based method. We show the results over different TB loss rates (0.5%, 1%, 2.5%, 5%, 10%, 15%) as well as for different MTU sizes (350, 700, 1400 byte per IP packet) and burst length sizes (25TB, 50TBs). We additionally applied a random TB (TB payload size = 184 byte) loss scenario (burst length=1TB), as a lower bound in Figure 5. The average video coding layer NAL unit size is 843 bytes. We applied 500 simulations runs on the SVC stream for each of the aforementioned test cases.

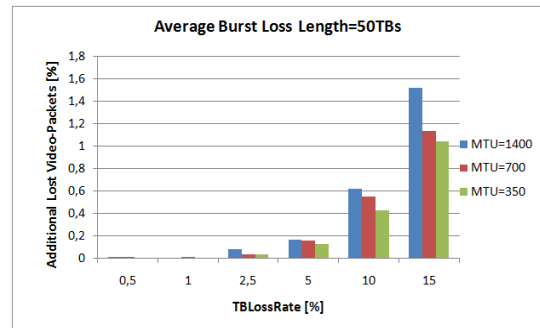


Figure 3 – Add. lost packets, avg. burst loss length=50TBs

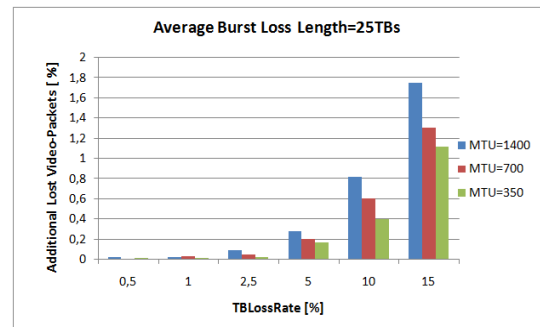


Figure 4 – Add. lost packets, avg. burst loss length=25TBs

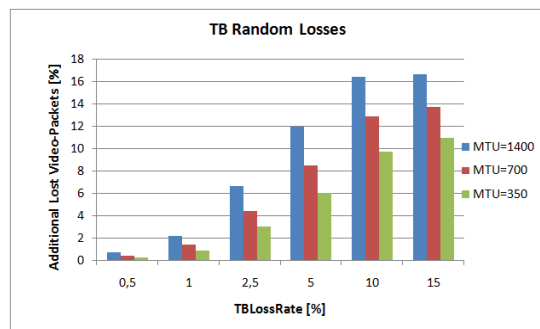


Figure 5 – Add. lost packets, random TB losses

It can be noticed in Figure 3 to Figure 5, that the percentage of additionally lost packets due to the timestamp-based method is only noticeable for random TB losses, which are unrealistic for

real mobile broadcast channels, as well with TB burst losses above 5%. Furthermore, it can be seen that higher MTU sizes are more sensitive (due to the less frequent appearance of timestamps in the flows), on the other hand higher average burst loss lengths are less sensitive to the potential problems introduced by TS-based DOR.

Figure 6 to Figure 8 show the delta PSNR for the above mentioned test cases, which can be seen as degradation when using the TS-based DOR in loss scenarios (Notice: There is no degradation in the loss free case). As reference, the received absolute PSNR is shown in Figure 9 for TB random loss, for avg. TB burst loss length=25TBs as well as for avg. TB burst loss length=50TBs for MTU size 1400bytes. In the unrealistic random loss scenario, the delta PSNR is relatively high for TB loss rates above 2.5%, but the average PSNR as shown in Figure 9 is already unacceptable for those scenarios. The delta PSNR for the TB burst loss cases is for realistic loss rates equal and below 5% clearly below 0.01 dB, which can be rated as not noticeable in the already erroneous scenarios.

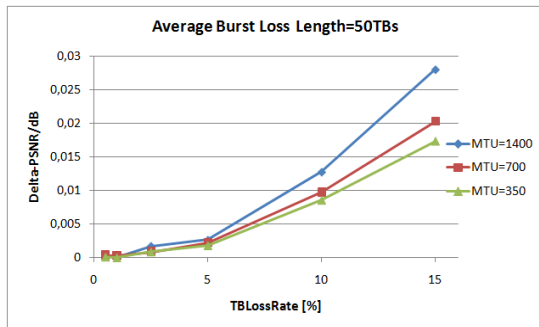


Figure 6 -- PSNR degradation, avg. burst loss length=50TBs

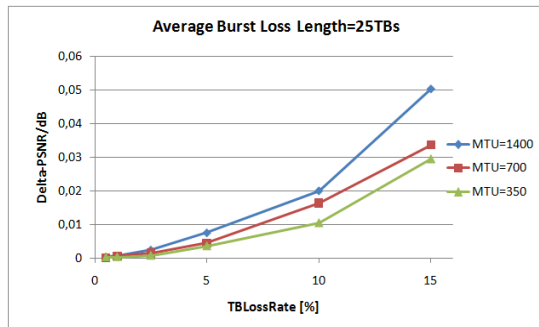


Figure 7 – PSNR degradation, avg. burst loss length=25TBs

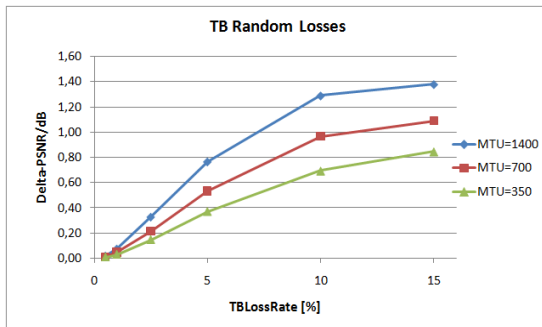


Figure 8 – PSNR degradation, random TB losses

The results above show, that the order problems introduced by using TS-based DOR do not impose any noticeable quality degradation for SVC streams including temporal scalability in

realistic loss cases for mobile broadcast channels. Only in the TB random loss scenario, a noticeable PSNR degradation can be seen, but these losses are irrelevant compared to the average received PSNR for the different loss cases at MTU=1400 bytes as shown in Figure 9. For burst errors of 25 and 50 TBs, it is not possible to distinguish the difference between both methods.

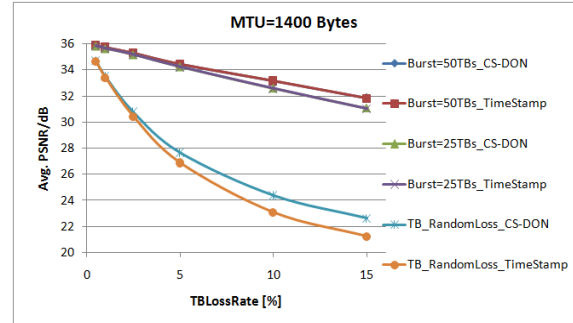


Figure 9 – Avg. PSNR, TB random/burst losses, MTU=1400

It can be further assumed, that streams without temporal scalability between network flows as well as with smaller GOP sizes will be less sensitive to the losses due to the higher appearance of synchronization points. Furthermore, more flows may also allow for enhancing the TS-based DOR due to additional possibilities for identifying synchronization points between multiple flows. It can be further summarized that higher transport block lengths and smaller MTU sizes show lower impact on the decoding order recovery.

## 5. Conclusion

We detailed the two methods for decoding order recovery (DOR) for transmission of SVC over multiple RTP flows as defined in the recently finished RTP payload format for SVC video. We discussed potential problems with the timestamp-based DOR method and showed that such discussed problems do not lead to noticeable quality degradation in realistic loss scenarios over an exemplary mobile broadcast/multicast channel as DVB-H.

## 6. References

- [1] S. Wenger, Y.-K. Wang, T. Schierl, A. Eleftheriadis (eds.): RTP payload format for SVC video, IETF AVT, March 2009
- [2] S. Wenger, Y.-K. Wang, T. Schierl: Transport and Signaling of SVC in IP Networks, IEEE TCSVT, vol. 17, no. 9, September 2007
- [3] S. McCanne, V. Jacobson, M. Vetterli: Receiver-Driven Layered Multicast, Proc. ACM Special Interest Group in Comm., ACM Press, pp. 117–130, 1996.
- [4] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson (eds.): RTP: A Transport Protocol for Real-Time Applications, IETF STD 0064, RFC 3550, July 2003, <http://tools.ietf.org/html/rfc3550>
- [5] M. Nilsson, D. Dalby, J. O'Donnell: Layered audiovisual coding for multicast distribution on IP networks, Proc. Distributed Imaging (Ref. No. 1999/109), IEE European Workshop, 1999
- [6] C. Perkins, T. Schierl (eds.): Rapid Synchronization of RTP Flows, July 2009, <http://tools.ietf.org/html/draft-ietf-avt-rapid-rtp-sync>
- [7] M. M. Hannuksela, V. Kumar M. Vadakital, and S. Jumisko-Pyykk: Comparison of Error Protection Methods for Audio-Video Broadcast over DVB-H, EURASIP Journal on Advances in Signal Processing, Article ID 71801, Vol. 2007
- [8] H. Schwarz, D. Marpe, and T. Wiegand: Overview of the scalable video coding extension of the H.264/AVC standard, IEEE TCSVT