Network Working Group
Request for Comments: 5583
Category: Standards Track

T. Schierl
Fraunhofer HHI
S. Wenger
Independent

July 2009

Signaling Media Decoding Dependency in the Session Description Protocol (SDP)

Abstract

This memo defines semantics that allow for signaling the decoding dependency of different media descriptions with the same media type in the Session Description Protocol (SDP). This is required, for example, if media data is separated and transported in different network streams as a result of the use of a layered or multiple descriptive media coding process.

A new grouping type "DDP" -- decoding dependency -- is defined, to be used in conjunction with RFC 3388 entitled "Grouping of Media Lines in the Session Description Protocol". In addition, an attribute is specified describing the relationship of the media streams in a "DDP" group indicated by media identification attribute(s) and media format description(s).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

An SDP session description may contain one or more media descriptions, each identifying a single media stream. A media description is identified by one "m=" line. Today, if more than one "m=" lines exist indicating the same media type, a receiver cannot identify a specific relationship between those media.

A Multiple Description Coding (MDC) or layered Media Bitstream contains, by definition, one or more Media Partitions that are conveyed in their own media stream. The cases we are interested in are layered and MDC Bitstreams with two or more Media Partitions. Carrying more than one Media Partition in its own session is one of the key use cases for employing layered or MDC-coded media. Senders, network elements, or receivers can suppress sending/forwarding/subscribing/decoding individual Media Partitions and still preserve perhaps suboptimal, but still useful, media quality.

One property of all Media Bitstreams relevant to this memo is that their Media Partitions have a well-defined usage relationship. For example, in layered coding, "higher" Media Partitions are useless without "lower" ones. In MDC coding, Media Partitions are complementary -- the more Media Partitions one receives, the better a reproduced quality may be. This document defines an SDP extension to indicate such a decoding dependency.

The trigger for the present memo has been the standardization process of the RTP payload format for the Scalable Video Coding (SVC) extension to ITU-T Rec. H.264 / MPEG-4 AVC [AVT-RTP-SVC]. When drafting [AVT-RTP-SVC], it was observed that the aforementioned lack in signaling support is one that is not specific to SVC, but applies to all layered or MDC codecs. Therefore, this memo presents a generic solution. Likely, the second technology utilizing the mechanisms of this memo will be Multi-View video coding. In Multi-View Coding (MVC) [AVT-RTP-MVC], layered dependencies between views are used to increase the coding efficiency, and, therefore, the properties of MVC with respect to the SDP signaling are comparable to those of SVC.

The mechanisms defined herein are media transport protocol dependent, and applicable only in conjunction with the use of RTP [RFC3550].

The SDP grouping of Media Lines of different media types is out of scope of this memo.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

3. Definitions

Media stream:
As per [RFC4566].

Media Bitstream:
A valid, decodable stream, containing all Media Partitions generated by the encoder. A Media Bitstream normally conforms to a media coding standard.

Media Partition:
A subset of a Media Bitstream intended for independent transportation. An integer number of Media Partitions forms a Media Bitstream. In layered coding, a Media Partition represents one or more layers that are handled as a unit. In MDC coding, a Media Partition represents one or more descriptions that are handled as a unit.

Decoding dependency:
The class of relationships Media Partitions have to each other. At present, this memo defines two decoding dependencies: layered coding and Multiple Description Coding.

Layered coding dependency:
Each Media Partition is only useful (i.e., can be decoded) when all of the Media Partitions it depends on are available. The dependencies between the Media Partitions therefore create a directed graph. Note: normally, in layered coding, the more Media Partitions are employed (following the rule above), the better a reproduced quality is possible.

Multiple Description Coding (MDC) dependency:
N of M Media Partitions are required to form a Media Bitstream, but there is no hierarchy between these Media Partitions. Most MDC schemes aim at an increase of reproduced media quality when more media partitions are decoded. Some MDC schemes require more than one Media Partition to form an Operation Point.

Operation Point:
In layered coding, a subset of a layered Media Bitstream that includes all Media Partitions required for reconstruction at a

certain point of quality, error resilience, or another property, and that does not include any other Media Partitions. In MDC coding, a subset of an MDC Media Bitstream that is compliant with the MDC coding standard in question.

4. Motivation, Use Cases, and Architecture

4.1. Motivation

This memo is concerned with two types of decoding dependencies: layered and multi-description. The transport of layered and Multiple Description Coding share as key motivators the desire for media adaptation to network conditions, i.e., related to bandwidth, error rates, connectivity of endpoints in multicast or broadcast scenarios, and the like.

o Layered decoding dependency:

In layered coding, the partitions of a Media Bitstream are known as media layers or simply layers. One or more layers may be transported in different media streams in the sense of [RFC4566]. A classic use case is known as receiver-driven layered multicast, in which a receiver selects a combination of media streams in response to quality or bit-rate requirements.

Back in the mid 1990s, the then-available layered media formats and codecs envisioned primarily (or even exclusively) a one-dimensional hierarchy of layers. That is, each so-called enhancement layer referred to exactly one layer "below". The single exception has been the base layer, which is self-contained. Therefore, the identification of one enhancement layer fully specifies the Operation Point of a layered coding scheme, including knowledge about all the other layers that need to be decoded.

SDP [RFC4566] contains rudimentary support for exactly this use case and media formats, in that it allows for signaling a range of transport addresses in a certain media description. By definition, a higher transport address identifies a higher layer in the one-dimensional hierarchy. A receiver needs only to decode data conveyed over this transport address and lower transport addresses to decode this Operation Point.

Newer media formats depart from this simple one-dimensional hierarchy, in that highly complex (at least tree-shaped) dependency hierarchies can be implemented. Compelling use cases for these complex hierarchies have been identified by industry. Support for it is therefore desirable. However, SDP, in its current form, does not allow for the signaling of these complex relationships. Therefore, receivers cannot make an informed decision on which layers to subscribe (in case of layered multicast).

Layered decoding dependencies may also exist in a Multi-View Coding environment. Views may be coded using inter-view dependencies to increase coding efficiency. This results in Media Bitstreams, that logically may be separated into Media Partitions representing different views of the reconstructed video signal. These Media Partitions cannot be decoded independently, and, therefore, other Media Partitions are required for reconstruction. To express this relationship, the signaling needs to express the dependencies of the views, which in turn are Media Partitions in the sense of this document.

o Multiple descriptive decoding dependency:

In the most basic form of MDC, each Media Partition forms an independent representation of the media. That is, decoding of any of the Media Partitions yields useful reproduced media data. When more than one Media Partition is available, then a decoder can process them jointly, and the resulting media quality increases. The highest reproduced quality is available if all original Media Partitions are available for decoding.

More complex forms of Multiple Description Coding can also be envisioned, i.e., where, as a minimum, N-out-of-M total Media Partitions need to be available to allow meaningful decoding.

MDC has not yet been embraced heavily by the media standardization community, though it is the subject of a lot of academic research. As an example, we refer to [MDC].

In this memo, we cover MDC because we a) envision that MDC media formats will come into practical use within the lifetime of this memo, and b) the solution for its signaling is very similar to the one of layered coding.

o Other decoding dependency relationships:

At the time of writing, no decoding dependency relationships beyond the two mentioned above have been identified that would warrant standardization. However, the mechanisms of this memo could be extended by introducing new codepoints for new decoding dependency types. If such an extension becomes necessary, as formally required in Section 5.2.2, the new decoding dependency type MUST be documented in an IETF Standards-Track document.

4.2. Use Cases

- Receiver-driven layered multicast:

  This technology is discussed in [RFC3550] and references therein. We refrain from elaborating further; the subject is well known and understood.

- Multiple end-to-end transmission with different properties:

  Assume a unicast and point-to-point topology, wherein one endpoint sends media to another. Assume further that different forms of media transmission are available. The difference may lie in the cost of the transmission (free, charged), in the available protection (unprotected/secure), in the quality of service (QoS) (guaranteed quality / best effort), or other factors.

  Layered and MDC coding allows matching of the media characteristics to the available transmission path(s). For example, in layered coding, it makes sense to convey the base layer over high QoS. Enhancement layers, on the other hand, can be conveyed over best effort, as they are "optional" in their characteristic -- nice to have, but non-essential for media consumption. In a different scenario, the base layer may be offered in a non-encrypted session as a free preview. An encrypted enhancement layer references this base layer and allows optimal quality play-back; however, it is only accessible to users who have the key, which may have been distributed by a conditional access mechanism.

5. Signaling Media Dependencies

5.1. Design Principles

The dependency signaling is only feasible between media descriptions described with an "m="-line and with an assigned media identification attribute ("mid"), as defined in [RFC3388]. All media descriptions grouped according to this specification MUST have the same media type. Other dependencies relations expressed by SDP grouping have to be addressed in other specifications. A media description MUST NOT be part of more than one group of the grouping type defined in this specification.

5.2. Semantics

5.2.1. SDP Grouping Semantics for Decoding Dependency

This specification defines a new grouping semantic Decoding Dependency "DDP":

DDP associates a media stream, identified by its mid attribute, with a DDP group. Each media stream MUST be composed of an integer number of Media Partitions. A media stream is identified by a session-unique media format description (RTP payload type number) within a media description. In a DDP group, all media streams MUST have the same type of decoding dependency (as signaled by the attribute defined in Section 5.2.2). All media streams MUST contain at least one Operation Point. The DDP group type informs a receiver about the requirement for handling the media streams of the group according to the new media level attribute "depend", as defined in Section 5.2.2.

When using multiple codecs, e.g., for the Offer/Answer model, the media streams MUST have the same dependency structure, regardless of which media format description (RTP payload type number) is used.

5.2.2. "depend" Attribute for Dependency Signaling per Media-Stream

This memo defines a new media-level attribute, "depend", with the following ABNF [RFC5234]. The identification-tag is defined in [RFC3388]. In the following ABNF, fmt, token, SP, and CRLF are used as defined in [RFC4566].

<CODE BEGINS>

```
- Neither the name of Internet Society, IETF or IETF Trust, nor the
  names of specific contributors, may be used to endorse or promote
  products derived from this software without specific prior written
  permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

depend-attribute =
        "a=depend:" dependent-fmt SP dependency-tag
           *(";" SP dependent-fmt SP dependency-tag) CRLF

dependency-tag   =
        dependency-type *1( SP identification-tag ":"
        fmt-dependency *("," fmt-dependency ))

dependency-type  = "lay"
                 / "mdc"
                 / token

dependent-fmt = fmt

fmt-dependency = fmt
<CODE ENDS>
```

dependency-tag indicates one or more dependencies of one dependent-fmt in the media description. These dependencies are signaled as fmt-dependency values, which indicate fmt values of other media descriptions. These other media descriptions are identified by their identification-tag values in the depend-attribute. There MUST be exactly one dependency-tag indicated per dependent-fmt.

dependent-fmt indicates the media format description, as defined in [RFC4566], that depends on one or more media format descriptions in the media description indicated by the value of the identification-tag within the dependency-tag.

fmt-dependency indicates the media format description in the media description identified by the identification-tag within the dependency-tag, on which the dependent-fmt of the dependent media description depends. In case a list of fmt-dependency values is given, any element of the list is sufficient to satisfy the dependency, at the choice of the decoding entity.

The depend-attribute describes the decoding dependency. The depend-attribute MUST be followed by a sequence of dependent-fmt and the corresponding dependency-tag fields, which identify all related media format descriptions in all related media descriptions of the dependent-fmt. The attribute MAY be used with multicast as well as with unicast transport addresses. The following dependency-type values are defined in this memo:

- o lay: Layered decoding dependency -- identifies the described media stream as one or more Media Partitions of a layered Media Bitstream. When "lay" is used, all media streams required for decoding the Operation Point MUST be identified by identification-tag and fmt-dependency following the "lay" string.

- o mdc: Multi-descriptive decoding dependency -- signals that the described media stream is part of a set of a MDC Media Bitstream. By definition, at least N-out-of-M media streams of the group need to be available to from an Operation Point. The values of N and M depend on the properties of the Media Bitstream and are not signaled within this context. When "mdc" is used, all required media streams for the Operation Point MUST be identified by identification-tag and fmt-dependency following the "mdc" string.

Further, dependency types MUST be defined in a Standards-Track document.

6. Usage of New Semantics in SDP

6.1. Usage with the SDP Offer/Answer Model

The backward compatibility in Offer/Answer is generally handled as specified in Section 8.4 of [RFC3388], as summarized below.

Depending on the implementation, a node that does not understand DDP grouping (either does not understand line grouping at all, or just does not understand the DDP semantics) SHOULD respond to an offer containing DDP grouping either (1) with an answer that ignores the grouping attribute or (2) with a refusal to the request (e.g., 488 Not acceptable here or 606 Not acceptable in SIP).

In case (1), if the original sender of the offer still wishes to establish communications, it SHOULD generate a new offer with a single media stream that represents an Operation Point. Note: in most cases, this will be the base layer of a layered Media Bitstream, equally possible are Operation Points containing a set of enhancement layers as long as all are part of a single media stream. In case (2), if the sender of the original offer has identified that the refusal to the request is caused by the use of DDP grouping, and if the sender of the offer still wishes to establish the session, it SHOULD retry the request with an offer including only a single media stream.

If the answerer understands the DDP semantics, it is necessary to take the "depend" attribute into consideration in the Offer/Answer procedure. The main rule for the "depend" attribute is that the offerer decides the number of media streams and the dependency between them. The answerer cannot change the dependency relations.

For unicast sessions where the answerer receives media, i.e., for offers including media streams that have a directionality indicated by "sendonly", "sendrecv", or have no directionality indicated, the answerer MAY remove media Operation Points. The answerer MUST use the dependency relations provided in the offer when sending media. The answerer MAY send according to all of the Operation Points present in the offer, even if the answerer has removed some of those Operation Points. Thus, an answerer can limit the number of Operation Points being delivered to the answerer while the answerer can still send media to the offerer using all of the Operation Points indicated in the offer.

For multicast sessions, the answerer MUST accept all Operation Points and their related decoding dependencies or MUST remove non-accepted Operation Points completely. Due to the nature of multicast, the receiver can select which Operation Points it actually receives and processes. For multicast sessions that allow the answerer to also send data, the answerer MAY send all of the offered Operation Points.

In any case, if the answerer cannot accept one or more offered Operation Points and/or the media stream's dependencies, the answerer MAY re-invite with an offer including acceptable Operation Points and/or dependencies.

Note: Applications may limit the possibility of performing a re-invite. The previous offer is also a good hint to the capabilities of the other agent.

6.2. Declarative usage

If a Real Time Streaming Protocol (RTSP) receiver understands signaling according to this memo, it SHALL set up all media streams that are required to decode the Operation Point of its choice.

If an RTSP receiver does not understand the signaling defined within this memo, it falls back to normal SDP processing. Two likely cases have to be distinguished: (1) if at least one of the media types included in the SDP is within the receiver's capabilities, it selects among those candidates according to implementation specific criteria for setup, as usual. (2) If none of the media types included in the SDP can be processed, then obviously no setup can occur.

6.3. Usage with AVP and SAVP RTP Profiles

The signaling mechanisms defined in this document MUST NOT be used to negotiate between using the attribute-value pair (AVP) [RFC3551] and SAVP [RFC3711] profile for RTP. However, both profiles MAY be used separately or jointly with the signaling mechanism defined in this document.

6.4. Usage with Capability Negotiation

This memo does not cover the interaction with Capability Negotiation [MMUSIC]. This issue is for further study and will be addressed in a different memo.

6.5. Examples

a.) Example for signaling layered decoding dependency:

The example below shows a session description with three media descriptions, all of type video and with layered decoding dependency ("lay"). Each of the media descriptions includes two possible media format descriptions with different encoding parameters as, e.g., "packetization-mode" (not shown in the example) for the media subtypes "H264" and "H264-SVC" given by the "a=rtpmap:"-line. The first media description includes two H264 payload types as media format descriptions, "96" and "97", as defined in [RFC3984] and represents the base layer Operation Point (identified by "mid:L1"). The two other media descriptions (identified by "mid:L2" and "mid:L3") include H264-SVC payload types as defined in [AVT-RTP-SVC], which contain enhancements to the base layer Operation Point or the first enhancement layer Operation Point (media description identified by "mid:L2").

The example shows the dependencies of the media format descriptions of the different media descriptions indicated by "DDP" grouping, "mid", and "depend" attributes. The "depend" attribute is used with the decoding dependency type "lay" indicating layered decoding dependency. For example, the third media description ("m=video 40004...") identified by "mid:L3" has different dependencies on the media format descriptions of the two other media descriptions: Media format description "100" depends on media format description "96" or "97" of the media description indentified by "mid:L1". This is an exclusive-OR, i.e., payload type "100" may be used with payload type "96" or with "97", but one of the two combinations is required for decoding payload type "100".

For media format description "101", it is different. This one depends on two of the other media descriptions at the same time, i.e., it depends on media format description "97" of the media description indentified by "mid:L1" and it also depends on media format description "99" of the media description indentified by "mid:L2". For decoding media format description "101", both media format description "97" and media format description "99" are required by definition.

```
v=0
o=svcsrv 289083124 289083124 IN IP4 host.example.com
s=LAYERED VIDEO SIGNALING Seminar
t=0 0
c=IN IP4 192.0.2.1/127
a=group:DDP L1 L2 L3
m=video 40000 RTP/AVP 96 97
b=AS:90
a=framerate:15
a=rtpmap:96 H264/90000
a=rtpmap:97 H264/90000
a=mid:L1
m=video 40002 RTP/AVP 98 99
b=AS:64
a=framerate:15
a=rtpmap:98 H264-SVC/90000
a=rtpmap:99 H264-SVC/90000
a=mid:L2
a=depend:98 lay L1:96,97; 99 lay L1:97
m=video 40004 RTP/AVP 100 101
b=AS:128
a=framerate:30
a=rtpmap:100 H264-SVC/90000
a=rtpmap:101 H264-SVC/90000
a=mid:L3
a=depend:100 lay L1:96,97; 101 lay L1:97 L2:99
```

b.) Example for signaling of multi-descriptive decoding dependency:

The example shows a session description with three media descriptions, all of type video and with multi-descriptive decoding dependency. Each of the media descriptions includes one media format description. The example shows the dependencies of the media format descriptions of the different media descriptions indicated by "DDP" grouping, "mid", and "depend" attributes. The "depend" attribute is used with the decoding dependency type "mdc" indicating layered decoding dependency. For example, media format description "104" in the media description ("m=video 40000...") with "mid:M1" depends on the two other media descriptions. It depends on media format description "105" of media description with "mid:M2", and it also depends on media format description "106" of media description with "mid:M3". In case of the multi-descriptive decoding dependency, media format description "105" and "106" can be used by definition to enhance the decoding process of media format description "104", but they are not required for decoding.

```
v=0
o=mdcsrv 289083124 289083124 IN IP4 host.example.com
s=MULTI DESCRIPTION VIDEO SIGNALING Seminar
t=0 0
c=IN IP4 192.0.2.1/127
a=group:DDP M1 M2 M3
m=video 40000 RTP/AVP 104
a=mid:M1
a=depend:104 mdc M2:105 M3:106
m=video 40002 RTP/AVP 105
a=mid:M2
a=depend:105 mdc M1:104 M3:106
m=video 40004 RTP/AVP 106
a=mid:M3
a=depend:106 mdc M1:104 M2:105
```

7. Security Considerations

All security implications of SDP apply.

There may be a risk of manipulation of the dependency signaling of a session description by an attacker. This may mislead a receiver or middle box, e.g., a receiver may try to compose a Media Bitstream out of several RTP packet streams that does not form an Operation Point, although the signaling made it believe it would form a valid Operation Point, with potential fatal consequences for the media decoding process. It is recommended that the receiver SHOULD perform an integrity check on SDP and follow the security considerations of SDP to only trust SDP from trusted sources.

8. IANA Considerations

The following contact information shall be used for all registrations included here:

```
Contact:      Thomas Schierl
              email: ts@thomas-schierl.de
              tel: +49-30-31002-227
```

The following semantics have been registered by IANA in Semantics for the "group" SDP Attribute under SDP Parameters.

Semantics	Token	Reference
Decoding Dependency	DDP	RFC 5583

The SDP media-level attribute "depend" has been registered by IANA in Semantics for "att-field (media level only)". The registration procedure in Section 8.2.4 of [RFC4566] applies.

SDP Attribute ("att-field (media level only)"):

```
Attribute name:     depend
Long form:          decoding dependency
Type of name:       att-field
Type of attribute:  media level only
Subject to charset: no
Purpose:            RFC 5583
Reference:          RFC 5583
Values:             see this document and registrations below.
```

The following semantics have been registered by IANA in Semantics for the "depend" SDP Attribute under SDP Parameters:

Semantics of the "depend" SDP attribute:

Semantics	Token	Reference
Layered decoding dependency	lay	RFC 5583
Multi-descriptive decoding dependency	mdc	RFC 5583

New registrations for semantics of the "depend" SDP attribute are added by the "Specification Required" policy as defined in [RFC5226].

9. Informative Note on "The SDP (Session Description Protocol) Grouping Framework"

Currently, there is ongoing work on [RFC3388bis]. In [RFC3388bis], the grouping mechanism is extended in a way that a media description can be part of more than one group of the same grouping type in the same session description. However, media descriptions grouped by this document must be at most part of one group of the type "DDP" in the same session description.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3388] Camarillo, G., Eriksson, G., Holler, J., and H. Schulzrinne, "Grouping of Media Lines in the Session Description Protocol (SDP)", RFC 3388, December 2002.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

10.2. Informative References

[AVT-RTP-SVC] Wenger, S., Wang Y.-K., Schierl, T. and A. Eleftheriadis, "RTP Payload Format for SVC Video", Work in Progress, March 2009.

[RFC3388bis] Camarillo, G "The SDP (Session Description Protocol) Grouping Framework", Work in Progress, January 2009.

[MMUSIC] Andreasen, F., "SDP Capability Negotiation", Work in Progress, May 2009.

[AVT-RTP-MVC] Wang, Y.-K. and T. Schierl, "RTP Payload Format for MVC Video", Work in Progress, February 2009.

[MDC] Vitali, A., Borneo, A., Fumagalli, M., and R. Rinaldo, "Video over IP using Standard-Compatible Multiple Description Coding: an IETF proposal", Packet Video Workshop, April 2006, Hangzhou, China.

[RFC3984] Wenger, S., Hannuksela, M., Stockhammer, T., Westerlund, M., and D. Singer, "RTP Payload Format for H.264 Video", RFC 3984, February 2005.

Appendix A. Acknowledgements

The author Thomas Schierl of Fraunhofer HHI is sponsored by the European Commission under the contract number FP7-ICT-214063, project SEA.

We want to also thank Magnus Westerlund, Joerg Ott, Ali Begen, Dan Wing, Helmut Burklin, and Jean-Francois Mule for their valuable and constructive comments to this memo.

Authors' Addresses

Thomas Schierl
Fraunhofer HHI
Einsteinufer 37
D-10587 Berlin
Germany

Phone: +49-30-31002-227
EMail: ts@thomas-schierl.de

Stephan Wenger
2400 Skyfarm Dr.
Hillsborough, CA 94010
USA

Phone: +1-415-713-5473
EMail: stewe@stewe.org