

AVTCORE Working Group
INTERNET-DRAFT
Category: Informational
Expires: January 6, 2016

B. Aboba
Microsoft Corporation

6 July 2015

**Codec-Independent Selective Forwarding
draft-aboba-avtcore-sfu-rtp-00.txt**

Abstract

Selective Forwarding Units (SFUs) supporting Scalable Video Coding (SVC) typically parse the RTP payload in the forwarding plane, and often utilize codec-specific control messages within the control plane. As a result, the control and/or forwarding planes of these implementations need to be modified (sometimes substantially) in order to support additional video codecs. With SFUs now supporting VP8 in addition to H.264/SVC, and with additional video codecs expected to become popular, the inflexibility of SFU designs that depend on RTP payload parsing has become increasingly apparent. In addition, these designs cannot function where the RTP payload is inaccessible, such as when it is encrypted with a key not available to the SFU. Based on a summary of SFU implementation practice, this document develops requirements for codec-independent SFUs.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 6, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Performance	4
1.2.	Terminology	4
2.	Control Plane	6
2.1	RTCP	6
3.	Forwarding Plane	7
3.1	RTP header	7
3.2	RTP payload	8
3.3	Problematic payload fields	10
4.	Security Considerations	11
5.	IANA Considerations	12
6.	References	12
6.1.	Informative references	12
	Acknowledgments	14
	Authors' Addresses	15

1. Introduction

Selective Forwarding Units (SFUs) supporting simulcast and Scalable Video Coding (SVC) are widely deployed for use in video conferencing. While initial deployments focused on H.264/SVC [[H.264](#)] [[RFC6190](#)], SFUs supporting the VP8 [[RFC6386](#)] [[I-D.ietf-payload-vp8](#)] codec have also been deployed and implementations supporting VP9 [[I-D.grange-vp9-bitstream](#)] [[I-D.uberti-payload-vp9](#)] and HEVC [[HEVC](#)] [[I-D.ietf-payload-rtp-h265](#)] are expected.

Given the growing demand for SFUs supporting multiple codecs, the disadvantages of codec-specific forwarding and control planes have become apparent. These include:

1. Forwarding plane maintenance. SFU implementations originally developed for a specific codec such as H.264/SVC typically require updates and testing to support each additional codec. Where the original forwarding logic depended on RTP payload fields in the original supported codec that have no analog within additional video codecs (such as the PACSI NAL unit in H.264/SVC), forwarding path re-design or per-codec forwarding logic may be required, increasing complexity and potentially adversely impacting implementation performance.
2. Control plane maintenance. SFU implementations developed for a specific codec often utilize codec-specific control messages (such as the SEI layer drop message [[RFC6190](#)]) that may have no analog within other video codecs. In addition, SFU implementations may depend on support for RTCP feedback messages [[RFC4585](#)] outside the mainstream [[I-D.ietf-rtcweb-rtp-usage](#)], thereby requiring control path re-design and/or support for additional feedback messages in order to support additional codecs.
3. Interoperability issues. Experience has shown that SFUs that depend on the RTP payload can be sensitive to differences in the encoder implementation. As an example, implementations supporting H.264/SVC [[RFC6190](#)] vary in their treatment of the PACSI NAL unit. While some implementations require the PACSI NAL unit to be present in each packet and to be located first within the RTP payload, other implementations do not impose these restrictions, and some do not generate or parse the PACSI NAL unit at all. Due to these differences, even implementations that are compatible at the bitstream level [[H.264](#)] and are generally conformant to the RTP payload specification [[RFC6190](#)] can fail to interoperate.
4. Payload snooping. While all SFUs require access to the IP header as well as RTCP, and many require the ability to modify RTP header fields and to insert, delete or modify RTP header extensions, in

addition codec-dependent SFU implementations require access to the RTP payload. Therefore codec-dependent SFUs have access to audio and video data in addition to metadata. This is potentially problematic for multi-tenant SFUs where the same SFU may be shared by multiple customers, each of whom can require assurance that the contents of the RTP payload is available only to endpoints under their control.

To address these problems, this document develops requirements for codec-independent SFU operation, both within the control and forwarding planes.

1.1. Performance

This document does not take a position on whether codec-independent SFUs offer enduring performance benefits. By itself, substituting RTP header extension(s) for RTP payload fields within the SFU forwarding plane seems unlikely to have a long-term impact on SFU performance, although in the short-term, forwarding performance might be improved in some implementations.

As described in [Section 3.1](#), SFU implementations frequently modify one or more RTP [\[RFC3550\]](#) header fields covered by the SRTP [\[RFC3711\]](#) authentication tag, requiring the authentication tag to be recomputed and modified prior to forwarding. In addition, where an SRTP cipher-suite is negotiated that depends upon one or more modified RTP header fields (such as the SSRC field), it is also necessary for the SFU to decrypt and then re-encrypt SRTP packets. As a result, SFUs modifying RTP header fields cannot reduce cryptographic operations where SRTP per-packet integrity protection and confidentiality is desired.

Nevertheless, measurements presented in [\[Hellwagner\]](#) indicate reductions in forwarding latency and improvements in performance from removing cryptographic operations from the forwarding plane. Such a savings might be possible if only per-packet integrity protection were utilized within SRTP. Investigating the security implications of this (which would involve reliance on end-to-end confidentiality and integrity protection applied to the RTP payload) is outside the scope of this document.

1.2. 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 [\[RFC2119\]](#).

"media aware network element (MANE)" is defined as in [\[RFC6184\]](#)
[Section 4.1](#):

A network element, such as a middle-box, selective forwarding unit, or application layer gateway that is capable of parsing certain aspects of the RTP payload headers or the RTP payload and reacting to their contents.

"Selective Forwarding Unit" is defined as in [I-D.ietf-avtcore-rtp-topologies-update] [Section 3.7](#):

The selective forwarding middle-box has been introduced in recently developed videoconferencing systems in conjunction with, and to capitalize on, scalable video coding as well as simulcasting... In both scalable coding and simulcast cases the video signal is represented by a set of two or more bitstreams, providing a corresponding number of distinct fidelity points. The middle-box selects which parts of a scalable bitstream (or which bitstream, in the case of simulcasting) to forward to each of the receiving End Points. The decision may be driven by a number of factors, such as available bit rate, desired layout, etc. Contrary to transcoding MCUs, these "Selective Forwarding Units" (SFUs) have extremely low delay, and provide features that are typically associated with high-end systems (personalized layout, error localization) without any signal processing at the middle-box. They are also capable of scaling to a large number of concurrent users, and--due to their very low delay-- can also be cascaded.

A "Codec Independent SFU" is an SFU that only utilizes information from the the RTP header or header extensions, but does not parse or modify the RTP payload. This implies that, unlike a MANE, a codec-independent SFU cannot do NAL-unit re-packetization, and needs to rely on IP fragmentation in the event of an MTU mismatch. In addition, codec-independent SFUs cannot source or interpret codec-specific control messages, such as the SEI layer-drop message defined in H.264/SVC [[RFC6190](#)].

Multiple RTP streams on a Single Transport (MRST): Multiple RTP streams on a single transport, carrying either a single Scalable Video Coding (SVC) bitstream or multiple simulcast streams.

Multiple RTP streams on Multiple Transports (MRMT): Multiple RTP streams carrying a single Scalable Video Coding (SVC) bitstream on Multiple Transports.

RTP stream: See [[I-D.ietf-avtext-rtp-grouping-taxonomy](#)]. Within this document, one RTP stream can be utilized to transport one or more sub-layers.

Single RTP stream on a Single Transport (SRST): Single RTP stream

carrying a single Scalable Video Coding (SVC) bitstream on a Single (Media) Transport.

Transport: A tuple comprising the source and destination IP address and ports.

2. Control Plane

2.1. RTCP

Within the Control Plane, SFU implementations require cleartext access to RTCP messages as well as to information provided within signaling. This implies that an SFU supporting SRTP MUST have access to SRTCP master keys.

RTCP messages commonly used by SFUs include the Sender Report (SR), Receiver Report (RR), SDES and BYE messages, as well as feedback messages. Note that even though SFU implementations need to decrypt RTCP messages, designs that do not modify SSRCs can forward RTCP messages such as SDES without modification (or even parsing), which has the additional potential advantage of enabling SFUs to forward RTCP messages they do not understand.

2.1.1. Feedback Messages

RTCP feedback messages widely supported by existing SFU implementations include Generic NACK [[RFC4585](#)], PLI [[RFC4585](#)] and FIR [[RFC5104](#)]. Generic NACK, defined in [[RFC4585](#)] [Section 6.2.1](#) and retransmission payload, defined in [[RFC4588](#)] are particularly effective where temporal scalability is supported, making it possible to repair losses within the base layer of an SVC bitstream in time to enable decoding of the succeeding base layer P frame.

FIR, defined in [[RFC5104](#)] [Section 4.3.1](#), is frequently employed within SFUs when switching between video streams. When the SFU decides to forward an alternative simulcast stream to a participant, or when an SFU decides to switch from forwarding video stream(s) sent by a now-quiet participant to stream(s) originated by an active speaker, an FIR is sent by the SFU to the sender of the switched-in RTP stream. The sender responds with an I-frame that is subsequently forwarded to the recipients, ensuring that they can decode subsequent frames within the switched in RTP stream.

In contrast, feedback messages such as SLI, defined in [[RFC4585](#)] [Section 6.3.2](#) and RPSI, defined in [[RFC4585](#)] [Section 6.3.3](#) are rarely supported in modern SFU implementations. SLI is infrequently supported because Generic NACK provides similar functionality in a codec-independent manner, while more efficiently supporting non-

contiguous packet loss.

RPSI is rarely supported since it requires that the encoder revert to use of an alternative reference picture not only for the participant experiencing the loss but for all participants. Thus RPSI implies additional bandwidth usage not only on the link experiencing the loss, but also on the links utilized by other conference participants, thereby potentially exacerbating congestion problems. When temporal scalability is implemented, it is typically possible to employ retransmission [[RFC4588](#)] and/or Forward Error Correction [[RFC5109](#)] to protect against loss within the base layer. In the (unlikely) event that these measures are insufficient, PLI and/or FIR feedback messages can be employed, so that RPSI support is unnecessary.

3. Forwarding Plane

3.1. RTP header

Most existing SFU implementations modify one or more RTP header fields, including the payload type (PT), SSRC, CSRC, and sequence number fields. SFU implementations may also modify RTP header extensions such as the abs-send-time extension [[ABS-SEND](#)].

In signaling mechanisms such as SDP Offer/Answer [[RFC3264](#)], each conference participant can specify the PT they wish to receive and/or send for a particular codec. Since conference participants can choose different PT values for the same codec, the SFU needs to be able to modify the PT field before forwarding RTP packets to a recipient.

Some SFU implementations allocate their own SSRCs, which they place within RTP packets that they send to recipients, as well as within RTCP messages that they send. As a result, the SSRC of RTP packets they receive is re-written prior to forwarding to conference participants. As noted in [[I-D.ietf-avtcore-rtp-topologies-update](#)], SSRC re-writing has implications for the control plane as well as the forwarding plane.

Some SFUs only forward video streams from participants that are considered to be active or recent speakers or (to enable access by hearing or speech impaired users) recent real-time text and/or chat contributors. Techniques used for switching are described in [[LASTN](#)]. In order to determine the active or recent speakers, RTP header extensions such as the client-to-mixer audio level extension [[RFC6464](#)] may be used, and this header extension may therefore need to be available to the SFU in cleartext.

SFUs that allocate their own SSRCs and re-write the SSRC field may utilize the CSRC field with video streams they forward in order to indicate a switch of an RTP stream from one contributor to another. SFUs that do not allocate their own SSRCs typically do not utilize the CSRC field in this way.

Where an SFU drops one or more layers within an SVC bitstream utilizing SRST transport, it is necessary to rewrite sequence numbers as well as to customize RTCP sender reports to reflect the packets actually sent to individual participants. Rewriting of sequence numbers is also required in SFUs that receive simulcast streams with distinct SSRCs, but forward a single stream with a fixed SSRC, a practice common in SFUs supporting VP8.

Where MRST transport is used for SVC and/or simulcast, it is not necessary to rewrite sequence numbers on dropping a layer or simulcast stream, since each one uses its own SSRC and therefore has its own sequence number space. However, in order to avoid rewriting of sequence numbers on resuming a previously dropped stream, it is necessary to indicate the drop explicitly, such as via the PAUSED mechanism described in [[I-D.ietf-avtext-rtp-stream-pause](#)] or via the SEI layer drop message. The latter requires the SFU to insert a sequence number within RTP packets originated by the SFU.

One of the implications of SFU modifications to RTP header fields is that SFUs require access to SRTP master keys so as to be able to recompute the authentication tag, which covers the modified RTP header fields.

3.2. RTP payload

In addition to utilizing information from the RTP header in the forwarding path, existing SFU implementations also utilize information present in the RTP payload field. Within the VP8 and H.264/SVC codecs, this includes the following fields:

Frame type: IDR

Knowledge of the frame type enables the SFU to better protect I-frames using mechanisms such as QoS, FEC or RTX, as well as to determine when a stream switch can take place. Within H.264/SVC, the frame type can be determined from the `idr_flag`; within VP8, the P bit (Inverse key frame flag) can be used.

Discardable

A 'Discardable' bit indicates whether other frames depend on this one. For example within an implementation of temporal

scalability, only the highest temporal extension layer would provide a 'Discardable' indication - all other temporal layers would not. Within H.264/SVC this information is provided by the D bit; within VP8, it is provided by the N bit.

Layer identifier

Layer identifiers enable the SFU to determine the temporal, spatial or quality layer of a given packet. This is used in forwarding/drop decisions. Within H.264/SVC this information is present in the DID(3 bits), QID (4 bits) and TID (3 bits) fields. Within VP8, it is present in the TID field (2 bits). Within H.265, it is present in the LayerID (6 bits) and TID (3 bits) fields.

TL0PICIDX

This field enables an SFU to determine whether a received frame is dependent on a base layer frame which the SFU has not previously received in its entirety. This is important information since it may represent the first indication that all or part of a base layer frame needs to be recovered before the arrival of subsequent base layer frames that depend on it. Within H.264/SVC the IDR PICID and TL0PICIDX fields are present if the Y bit is set. Within VP8, the TL0PICIDX field is present if the Y bit is set.

S/E

The S bit indicates the first packet within a frame for a given layer. For example, the S bit would always be set in the first packet of a base layer frame, as well as in the first packet of a spatial, quality or temporal enhancement to that base layer frame. Similarly, the E bit indicates the last packet within a frame for a given layer.

Note that the E bit does not have an identical meaning to the M bit within the RTP header, which indicates the last packet in an access unit. As an example, where spatial scalability is in use, the last packet of a base layer frame would have the E bit set, but would not have the M bit set - the M bit would only be set on the last packet of the spatial enhancement to that base layer frame.

Both the S and E bits are defined within H.264/SVC as well as within H.265. Only the S bit is defined within VP8; the lack of an E bit within VP8 is not an issue since VP8 only supports temporal scalability.

3.3. Problematic payload fields

In addition to payload fields recommended for support within codec-independent SFUs, it is also useful to cover payload fields that are not recommended, because there are better alternatives. These include the following fields:

PictureID

While the PictureID field is utilized within the RPSI feedback message, as well as being supported within the VP8 and VP9 codecs, its use is not recommended within codec-independent SFUs. As noted earlier, RPSI feedback messages are rarely supported within existing SVC and simulcast implementations, since better loss recovery mechanisms are available. As a result, an SFU does not need to originate an RPSI message which would require the PictureID. Where it is necessary for the SFU to determine which picture a given packet belongs to, RTP header fields such as the timestamp and sequence number can be used instead.

Macroblocks

While macroblock identifiers are universally supported within video codecs, their use by codec-independent SFUs is not recommended. As noted earlier, SLI feedback messages (which require a starting macroblock) are rarely supported within existing SFUs, since better loss recovery mechanisms are available. Therefore a codec-independent SFU should never need to source an SLI message.

NRI

While the NRI field defined in H.264/SVC is used by some SFU implementations, its use in codec-independent SFUs is not recommended since there is no equivalent within the VP8 and VP9 codecs, and the IDR, discardable and layer identification information provide an adequate substitute.

PRID

The PRID field defined in H.264/SVC has no analog in VP8 or VP9. While some SFU implementations do use this field (sometimes for purposes other than those for which it was originally defined), its use in codec independent SFUs is not recommended, since the D/N bits as well as the layer identification fields provide much the same information.

F

The F bit defined in H.264/SVC has no analog in VP8 or VP9 and is always set to zero in H.265. Therefore its use in codec-independent SFUs is not recommended.

KEYIDX

The KEYIDX field is defined in VP8 as well as VP9, providing information on the key frame that a given packet is dependent on. Existing SFU implementations do not utilize this field in their forwarding/drop decisions, preferring to instead utilize the TL0PICIDX field.

4. Security Considerations

Codec-independent forwarding provides significant architectural benefits, even in situations where it is not possible to fully protect the privacy of conference participants.

As noted earlier in this document, SFUs require cleartext access to control plane information such as information provided in signaling as well as RTCP reports and feedback messages. SFUs also require access to information required for forwarding plane operation, such as information available within RTP extensions (such as frame start/end bits, layer information, etc.) and RTP header extensions (e.g. abs-send-time [[ABS-SEND](#)] and client-to-mixer audio level indication [[RFC6464](#)]).

In addition, it should be understood that the information available to passive observers is significant, since this includes metadata such as the IP addresses of conference participants (unless protected via a relay or onion routing service), statistics such as packets/bytes sent, and packet sizes.

Based on frame size, an SFU can determine periods of rapid motion, and based on the client-to-mixer audio level indication, the SFU can determine recent speakers. Much of this same information may also be gleaned by a passive observer with access to packet size and RTP header data (such as the SSRC field). Combined with metadata, this information would allow the SFU or a passive observer to determine the roles of various participants within the conference as well as their levels of engagement.

Based on RTCP reports and feedback message providing information on delay, jitter and losses, the SFU can determine the bandwidth available to participants as well as aspects of endpoint connectivity (e.g. whether their network access is wireless/wired, satellite/terrestrial, etc.).

Even when the RTP payload is encrypted with a key unavailable to the SFU, the ability of the SFU to obtain and potentially forge signaling implies that end-to-end privacy is only guaranteed when media is authenticated end-to-end by an entity that can be considered trustworthy even when presented with an intercept order.

5. IANA Considerations

This document does not require actions by IANA.

6. References

6.1. Informative References

- [ABS-SEND] WebRTC site, "The Absolute Send Time extension",
<http://www.webrtc.org/experiments/rtp-hdrext/abs-send-time>,
retrieved July 6, 2015.
- [H.264] ITU-T Recommendation H.264, "Advanced video coding for
generic audiovisual services", March 2010.
- [Hellwagner] Hellwagner, H., Kuschnig, R., Stutz, T. and Andreas Uhl,
"Efficient In-Network Adaptation of Encrypted H.264/SVC
Content", Journal of Image Communication, Volume 24, Issue
9, October 2009 Pages 740-758, Esevier Science, retrieved
from <http://wavelab.at/papers/Hellwagner09a.pdf>
- [HEVC] ITU-T Recommendation H.265, "High efficiency video coding",
April 2013.
- [I-D.grange-vp9-bitstream]
Grange, A. and H. Alvestrand, "A VP9 Bitstream Overview",
[draft-grange-vp9-bitstream-00](#) (work in progress), February
2013.
- [I-D.ietf-avtcore-rtp-multi-stream]
Lennox, J., Westerlund, M., Wu, W., and C. Perkins,
"Sending Multiple Media Streams in a Single RTP Session",
[draft-ietf-avtcore-rtp-multi-stream-07](#) (work in progress),
March 2015.
- [I-D.ietf-avtcore-rtp-topologies-update]
Westerlund, M. and S. Wenger, "RTP Topologies", [draft-ietf-avtcore-rtp-topologies-update-10](#) (work in progress), July
2015.
- [I-D.ietf-avtext-rtp-grouping-taxonomy]
Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and

B. Burman, "A Taxonomy of Grouping Semantics and Mechanisms for Real-Time Transport", [draft-ietf-avtext-rtp-grouping-taxonomy-07](#) (work in progress), June 2015.

[I-D.ietf-avtext-rtp-stream-pause]

Burman, B., Akram, A., Even, R. and M. Westerlund, "RTP Stream Pause and Resume", [draft-ietf-avtext-rtp-stream-pause-08](#) (work in progress), July 2015.

[I-D.ietf-payload-rtp-h265]

Wang, Y.-K., Sanchez, Y., Schierl, T., Wenger, S. and M. M. Hannuksela, "RTP Payload Format for High Efficiency Video Coding", [draft-ietf-payload-rtp-h265-13](#) (work in progress), June 2015.

[I-D.ietf-payload-vp8]

Westin, P., Lundin, H., Glover, M., Uberti, J. and F. Galligan, "RTP Payload Format for VP8 Video", [draft-ietf-payload-vp8-16](#) (work in progress), June 2015.

[I-D.ietf-rtcweb-rtp-usage]

Perkins, C., Westerlund, M. and J. Ott, "Web Real-Time Communication (WebRTC): Media Transport and Use of RTP", [draft-ietf-rtcweb-rtp-usage-25](#) (work in progress), June 2015.

[I-D.uberti-payload-vp9]

Uberti, J., Holmer, S., Flodman, M. and J. Lennox, "RTP Payload Format for VP9 Video", [draft-uberti-payload-vp9-01](#) (work in progress), March 2015.

[LASTN]

Grozev, B., Marinov, L., Singh, V. and E. Ivov, "Last N: relevance-based selectivity for forwarding video in multimedia conferences", Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 19-24, 2015, ISBN: 978-1-4503-3352-8

[RFC2119]

Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

[RFC3264]

Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 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.

- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and Norrman, K., "The Secure Real-time Transport Protocol (SRTP)", [RFC 3711](#), March 2004.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and Rey, J., "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", [RFC 4585](#), July 2006.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", [RFC 4588](#), July 2006.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and Burman, B., "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", [RFC 5104](#), February 2008.
- [RFC5109] Li, A., "RTP Payload Format for Generic Forward Error Correction", [RFC 5109](#), December 2007.
- [RFC6184] Wang, Y.-K., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", [RFC 6184](#), May 2011.
- [RFC6190] Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", [RFC 6190](#), May 2011.
- [RFC6386] Bankoski, J., Koleszar, J., Quillio, L., Salonen, J., Wilkins, P., and Y. Xu, "VP8 Data Format and Decoding Guide", [RFC 6386](#), November 2011.
- [RFC6464] Lennox, J., Ivov, E. and E. Marocco, "A Real-time Transport Protocol (RTP) Header Extension for Client-to-Mixer Audio Level Indication", [RFC 6464](#), December 2011.

Acknowledgments

The author acknowledges discussions relating to the topic of this memo within the IETF PAYLOAD, AVTCORE and AVTEXT WGs, as well as discussions with the IMTC MANE Activity Group. In particular, Emil Ivov, Alex Eleftheriadis and Stephan Wenger have provided helpful comments relating to this topic.

Authors' Addresses

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

Email: bernard_aboba@hotmail.com