

[MS-RTVPPF]: RTP Payload Format for RT Video Streams Extensions

Intellectual Property Rights Notice for Protocol Documentation

- **Copyrights.** This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the protocol documentation.
- This protocol documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, the protocols may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>). If you would prefer a written license, or if the protocols are not covered by the OSP, patent licenses are available by contacting protocol@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. This protocol documentation is intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it. A protocol specification does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them.

Revision Summary			
Author	Date	Version	Comments

Microsoft Corporation	April 4, 2008	0.1	Initial Availability
Microsoft Corporation	April 25, 2008	0.2	Revised and edited the technical content
Microsoft Corporation	June 27, 2008	1.0	Revised and edited the technical content
Microsoft Corporation	August 15, 2008	1.01	Revised and edited the technical content
Microsoft Corporation	December 12, 2008	2.0	Revised and edited the technical content
Microsoft Corporation	February 13, 2009	2.01	Revised and edited the technical content
Microsoft Corporation	March 13, 2009	2.02	Edited the technical content

Table of Contents

1	Introduction.....	6
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	7
1.3	Protocol Overview (Synopsis).....	7
1.4	Relationship to Other Protocols.....	8
1.5	Prerequisites/Preconditions.....	8
1.6	Applicability Statement.....	8
1.7	Versioning and Capability Negotiation.....	8
1.8	Vendor-Extensible Fields	9
1.9	Standards Assignments	9
2	Messages.....	9
2.1	Transport.....	9
2.2	Message Syntax.....	9
2.2.1	RTP Header Usage.....	10
2.2.2	RTVideo Basic RTP Payload Format	10
2.2.3	RTVideo Extended RTP Payload Format.....	11
2.2.4	RTVideo Extended 2 RTP Payload Format.....	13
2.2.5	RTVideo FEC RTP Payload Format.....	14
2.2.5.1	Forward Error Correction (FEC) Algorithm.....	15
2.2.5.2	FEC Metadata Packet Usage <2>.....	16
2.2.5.3	RTP Header Usage.....	16
2.2.6	Parsing RTVideo Packets.....	16
2.2.7	SP-Frame and Cached Frame mechanisms.....	17
3	Protocol Details	17
3.1	Sender Details.....	17
3.1.1	Abstract Data Model	17
3.1.1.1	Choice of RTP Payload Format.....	17
3.1.1.2	Fragmenting Video Frames	18
3.1.1.2.1	Maximum Video Fragment Size	18
3.1.1.2.2	Additional Requirement for FEC	18
3.1.1.3	Understanding the Sequence Header.....	18
3.1.1.4	Other Requirements.....	19
3.1.2	Timers	19
3.1.3	Initialization	19
3.1.4	Higher-Layer Triggered Events.....	19
3.1.4.1	Send a RTVideo Frame.....	19
3.1.5	Message Processing Events and Sequencing Rules.....	19
3.1.6	Timer Events.....	20
3.1.7	Other Local Events.....	20

3.2	Receiver Details	20
3.2.1	Abstract Data Model	20
3.2.2	Timers	20
3.2.3	Initialization	20
3.2.4	Higher-Layer Triggered Events	21
3.2.4.1	Receive a Video Packet.....	21
3.2.5	Message Processing Events and Sequencing Rules.....	21
3.2.6	Timer Events.....	21
3.2.7	Other Local Events	21
4	Protocol Examples.....	21
4.1	Basic RTP Payload Format Examples	21
4.1.1	I-Frame.....	21
4.1.1.1	First Packet.....	21
4.1.1.2	Second Packet.....	22
4.1.1.3	Last Packet.....	22
4.1.2	SP-Frame.....	22
4.1.2.1	First Packet.....	22
4.1.2.2	Second Packet.....	22
4.1.2.3	Last Packet.....	22
4.1.3	P-Frame or B-Frame.....	22
4.1.3.1	First Packet/LastPacket	23
4.2	Extended RTP Payload Format Examples.....	23
4.2.1	I-Frame.....	23
4.2.1.1	First Packet.....	23
4.2.1.2	Second Packet.....	23
4.2.1.3	Last Packet.....	23
4.2.2	P-Frame.....	24
4.2.2.1	First Packet/Last Packet	24
4.2.3	SP-Frame.....	24
4.2.3.1	First Packet.....	24
4.2.3.2	Second Packet.....	24
4.2.3.3	Last Packet.....	25
4.2.4	B-Frame	25
4.2.4.1	First Packet/Last Packet	25
4.3	FEC RTP Payload Format Examples.....	25
4.3.1	I-Frame.....	25
4.3.1.1	FEC Metadata Packet (FEC Version 0)	25
4.3.1.2	FEC Metadata Packet (FEC Version 1) <1>.....	26
4.3.2	SP-Frame.....	26
4.3.2.1	FEC Metadata Packet.....	26
5	Security	27
5.1	Security Considerations for Implementers.....	27
5.2	Index of Security Parameters.....	27

6	Appendix A: Product Behavior	27
7	Index	28

1 Introduction

This document specifies the RTP Payload Format for [RTVideo](#) Streams Extensions [MS-RTVPPF] protocol. It is a Microsoft® proprietary protocol describing the payload format for carrying the Microsoft real-time video streams (referred to as RTVideo video streams later in this document) in the payload of RTP. It is used to transmit and receive RTVideo streams in two-party peer-to-peer calls and in multi-party conference calls.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

big-endian
network byte order

The following terms are defined in [\[MS-OCSGLOS\]](#):

B-frame
forward error correction (FEC)
I-frame
P-frame
RTVC1
SP-frame

The following terms are specific to this document:

cached frame: A video frame that both the encoder and the decoder cache for later use. A cached frame acts as the reference frame for the next [SP-frame](#), [I-frames](#) and [SP-frames](#) are normally cached frames.

GOP: A group of pictures that starts with one **I-frame** and ends with the next **I-frame**, excluding the next **I-frame**. Refer to [\[SMPTE-VC-1\]](#) for more details.

RTVideo: A video stream that carries an **RTVC1** bitstream.

RTVideo FEC metadata packet: A packet generated using the **FEC** algorithm to provide redundancy. It is packetized in the **RTVideo FEC** RTP Payload Format.

RTVideo frame: A video frame encoded using the **RTVC1** codec.

sequence header: Encoding and display parameters that are placed before a group of pictures. Refer to [\[SMPTE-VC-1\]](#) for more details.

VC1: A Microsoft video codec. Also known as the SMPTE 421M standard [\[SMPTE-VC-1\]](#).

video data packet: A video data block that encapsulates a complete video frame or a fragment of a video frame. It contains the video payload header and the video payload.

video data packet list: A list of video data packets that belong to the same video frame.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn.microsoft.com/en-us/library/cc136647.aspx>, as an additional source.

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", October 2008.

[MS-OCSGLOS] Microsoft Corporation, "[Office Communications Server Master Glossary](#)", December 2008.

[MS-RTP] Microsoft Corporation, "[Real-time Transport Protocol \(RTP\) Extensions](#)", December 2008.

[RFC2119] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>.

[SMPTE-VC-1] Society of Motion Picture and Television Engineers, "VC-1 Compressed Video Bitstream Format and Decoding Process", SMPTE 421M-2006, 2006, <http://www.smpte.org/standards>. **Note** There is a charge to download the specification.

1.2.2 Informative References

None.

1.3 Protocol Overview (Synopsis)

[MS-RTVPF] specifies a payload format to transport an [RTVC1](#) bitstream using RTP.

[MS-RTVPF] accepts an RTVC1-encoded video frame. It fragments the video frame into one or more packets enumerated in a [data packet list containing RTVideo](#). Each RTVideo data packet contains an RTVideo payload header and a video payload. The RTVideo data packet list optionally has one or more (up to 31) [RTVideo FEC metadata packet](#) appended to the end of the list. Each RTVideo FEC metadata packet contains an RTVideo FEC payload header and FEC metadata.

1.4 Relationship to Other Protocols

[MS-RTVPF] carries the RTVC1 bitstream, defined in [\[SMPTE-VC-1\]](#) as a payload, and in turn is carried as a payload in RTP [\[MS-RTP\]](#).

1.5 Prerequisites/Preconditions

[MS-RTVPF] specifies only the payload format for RTVideo video streams. [MS-RTVPF] requires the establishment of an RTP stream, a mechanism to obtain RTVideo video frames for it to packetize, and a mechanism to render RTVideo video frames that it has depacketized.

Higher layers are required to provide [RTVideo frames](#) with the following added information about each frame:

- **I-frame Flag:** Specifies whether the frame is an **I-frame**.
- **SP Flag:** Specifies whether the frame is an **SP-frame**.
- **[cached frame](#) Flag:** Specifies whether the frame is a cached frame.
- **[sequence header](#):** This is required for each **I-frame**. It is not needed for other frame types.

Higher layers are required to provide video frames in referencing order. Frames being referenced are required to be provided earlier than frames referring to them.

Higher layers are also required to respect the following assumptions:

- An **I-frame** does not have any reference frame.
- **I-frames** and **SP-frames** are cached frames as well.
- An **SP-frame** refers to the previous cached frame.
- A **[P-frame](#)** refers to the previous **P-frame**, **SP-frame**, or **I-frame**.
- A **[B-frame](#)** refers to the previous **P-frame**, **SP-frame**, or **I-frame**. If the frame being referred to is not an **I-frame**, then the B-frame also refers to the reference frame of the reference frame.

1.6 Applicability Statement

[MS-RTVPF] is only applicable for transporting video frames encoded using the RTVC1 codec.

1.7 Versioning and Capability Negotiation

[MS-RTVPF] has the following versioning constraints:

- **Supported Transports:** This protocol uses RTP as its transport as discussed in [section 2.1](#).
- **Protocol Versions:** This protocol supports FEC version 0 and FEC version 1 as discussed in [section 2.2](#).

- The RTP Payload Format for RTVideo Streams Extensions protocol does not have any capability negotiation constraints.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

[MS-RTVPF] is a payload for the [\[MS-RTP\]](#) transport protocol and therefore relies on RTP for providing means to transport its payload over the network.

2.2 Message Syntax

[MS-RTVPF] defines four RTP payload formats:

- RTVideo Basic RTP Payload Format
The RTVideo Basic RTP Payload Format provides a basic scheme to packetize and transport RTVideo frames between a sender and receiver. It provides enough information to allow the receiver to reconstruct the video frames.
- RTVideo Extended RTP Payload Format
The RTVideo Extended RTP Payload Format extends the RTVideo Basic RTP Payload Format with extra fields. A video frame counter and the video frame counter of its reference frame are added. This provides a way for the receiver to actively drop frames received but not decodable due to packet loss. The receiver *MAY* consider a video frame as lost if one or more packets in the video frame are lost. It can then optionally drop the video frame that references the lost video frame, because the RTVideo decoder cannot decode the video frame.
- RTVideo Extended 2 RTP Payload Format
The RTVideo Extended 2 RTP Payload Format extends the RTVideo Extended RTP Payload Format. It carries four extra bytes.
- RTVideo FEC RTP Payload Format
The RTVideo FEC RTP Payload format is a special case of the RTVideo Extended 2 RTP Payload Format that provides a way to protect against a frame loss due to a packet loss. When [MS-RTVPF] is applied, one or more (up to 31) FEC metadata packets are added to the end of the video packet list for a video frame.

The C, SP, L, O, I, S, F, Sequence Header Length, and Sequence Header Bytes fields are the same as specified in [section 2.2.2](#).

M (1 bit): The meaning of this field is the same as specified in [section 2.2.2](#). It MUST be set to 1 in the RTVideo Extended RTP Payload Format.

M2 (1 bit): Payload format mode2 flag. This field MUST be set to 0 in the RTVideo Extended RTP Payload Format. The field is set to 1 in other RTP payload formats as specified in sections [2.2.4](#) and [2.2.5](#).

HiRFC (2 bits) and **RefFrameCounter** (8 bits):

The HiRFC field and RefFrameCounter field together specify the video frame counter (10 bits) for the reference frame or the reference frames (if the current frame is a B-frame) of this frame.

If the video frame is an **I-frame**, **P-frame**, or **SP-frame**, the whole 10-bit field specifies a frame counter.

The HiRFC field specifies the 2 high bits of the counter. The RefFrameCounter specifies the 8 low bits.

If the video frame is a B-frame, the HiRFC field MUST be set to 0. The 4 high bits of the RefFrameCounter specify one reference frame counter delta (RefFrameCounterDelta1) and the 4 low bits of the RefFrameCounter specify another reference frame counter delta (RefFrameCounterDelta2). These two reference counter delta values correspond to the frame counters (referred to as RefFrameCounter1 and RefFrameCounter2) of the two reference frames for the B-frame, respectively.

The two reference frame counters are calculated by subtracting the frame counter delta from the frame counter for the B-frame. These two reference frame counters MAY be the same, meaning the B-frame only references a single frame.

HiFC (2 bits) and **FrameCounter**(8 bits):

The HiFC field and FrameCounter field together specify the video frame counter (10 bits) for the video frame. The HiFC field specifies the 2 high bits. The FrameCounter specifies the 8 low bits.

DV (2 bits): Data version field. The value ranges from 0 to 3. This field SHOULD be ignored when receiving an RTVideo Extended RTP Payload Format packet and MUST be set to 0 when sending an RTVideo Extended RTP Payload Format packet.

E (1 bit): ExtraData field. In this document the FEC metadata is considered to be an extra data. A value of 1 specifies the packet is an FEC metadata packet. A value of 0 specifies it is

an RTVideo data packet. There is no extra data defined in the RTVideo Extended RTP Payload Format header. Due to this reason this field MUST be set to 0 in the RTVideo Extended RTP Payload Format header.

2.2.4 RTVideo Extended 2 RTP Payload Format

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
M	C	SP	L	O	I	S	F	M2	HiRFC	HiFC	DV	E	FrameCounter										RefFrameCounter								
Reserved																															
Sequence Header Length (optional)								Sequence Header Bytes (length varies)																							

The size of the RTVideo Extended 2 RTP Payload Format header varies. The minimal size is 8 bytes without a sequence header present. With a sequence header present, the maximum size is 72 bytes.

The C, SP, L, O, I, S, and F fields are the same as specified in [section 2.2.2](#).

The HiFC, HiRFC, M2, FrameCounter, and RefFrameCounter fields are the same as specified in [section 2.2.3](#).

M (1 bit): The meaning of this field is same as specified in [section 2.2.2](#). It MUST be set to 1 in the RTVideo Extended 2 RTP Payload Format.

M2 (1 bit): The meaning of this field is same as specified in [section 2.2.3](#). This field MUST be set to 1 in the RTVideo Extended 2 RTP Payload Format.

E (1 bit): The meaning of this field is same as specified in [section 2.2.3](#).

DV (2 bits): The meaning of this field is same as specified in [section 2.2.3](#). This field SHOULD be ignored when receiving an RTVideo Extended 2 RTP Payload Format packet and MUST be set to 0 when sending an RTVideo Extended 2 RTP Payload Format packet.

Reserved (32 bits): Reserved field. This field MUST be set to 0 when the E field is set to 0. This field is redefined when the E field is set to 1 as specified in the RTVideo FEC RTP Payload Format header.

2.2.5 RTVideo FEC RTP Payload Format

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
M	C	SP	L	O	I	S	F	M2	HiRFC	HiFC	DV	E	FrameCounter										RefFrameCounter								
M3	HiPN	Reserved		PacketNumberLo								HiLPL	EndOffset	LastPacketLengthLo																	
		FEC PacketsNumber <2>																													

The RTVideo FEC RTP Payload Format header can be considered as a special case of the RTVideo Extended 2 RTP Payload Format, where the S field MUST be set to 0 and E field MUST be set to 1.

This RTVideo FEC Payload Format header has a fixed size of 8 bytes.

The C, SP, L, O, I, and F fields are the same as specified in [section 2.2.2](#).

The HiFC, HiRFC, and RefFrameCounter fields MUST be set to 0.

S (1 bit): The meaning of this field is same as specified in [section 2.2.2](#). The S field MUST be set to 0 in the RTVideo FEC RTP Payload Format. This means the sequence header size and the sequence header data fields MUST NOT be present in the RTVideo FEC RTP Payload Format.

M (1 bit): The meaning of this field is same as specified in [section 2.2.2](#). This field MUST be set to 1 in the RTVideo FEC RTP Payload Format.

M2 (1 bit): The meaning of this field is same as specified in [section 2.2.3](#). This field MUST be set to 1 in the in the RTVideo FEC RTP Payload Format.

E (1 bit): The meaning of this field is same as specified in [section 2.2.3](#). This field MUST be set to 1 in the RTVideo FEC RTP Payload Format.

DV (2 bits): FEC version number. The value ranges from 0 to 3. The version number MUST be 00 or 01 in the RTVideo FEC RTP Payload Format. Currently it is set to 00 for the first FEC packet and equal to 01 for subsequent FEC packets of the same frame. <2>

Reserved/FECPacketsNumber (5 bits): The semantics of these bits depends on the value of DV. If DV \neq '01' then this is a field reserved for future use. If DV = '01' then this field represents an FECPacketsNumber field specifying the total number of contiguous FEC packets, generated by the FEC algorithm, associated with the packets conveying the video frame. The maximum number of FEC packets is limited to 31. if DV = '01' then **FECPacketsNumber** MUST NOT be set to 0. <2>

HiPN (2 bits) and **PacketNumberLo** (8 bits): The HiPN field and PacketNumberLo field specify the number of video packets (10 bits) the video frame is fragmented into. The HiPN field specifies the 2 high bits. The PacketNumberLo field specifies the 8 low bits.

M3(1 bits): Payload format mode3 flag. This field MUST be set to 0 in the RTVideo FEC Payload Format.

HiLPL (3 bits) and **LastPacketLengthLo** (8 bits): The HiLPL field and LastPacketLengthLo field specify the size (11 bits) of the last [video data packet](#). Both the video payload header size and the video payload size are counted. The HiLPL field specifies the 3 high bits. The LastPacketLengthLo field specifies the 8 low bits.

EndOffset(5 bits): Indicates the FEC metadata packet distance from the last video data packet, minus 1. For example, a video frame is fragmented into five video data packets. One FEC metadata packet is added after the five video data packets. These six packets are indexed starting at 0. The index of the first data packet is 0. The index of the last data packet is 4. The index of the FEC metadata packet is 5. The EndOffset field MUST be set to $5-4-1 = 0$.

2.2.5.1 Forward Error Correction (FEC) Algorithm

The FEC implementation is capable of detecting and correcting errors in the video payload header as well as the video payload data. It achieves this by constructing an FEC metadata packet(s), using the video payload header and video payload data, and sending it alongside video packets. Note only one FEC packet is constructed if DV = 00 .

Assume that a video frame is fragmented and packetized into N data packets. The RTP payload header size plus the payload data size MUST be the same for all data packets, except for the last data packet.

Each video data packet is considered as a FEC data block. The size of all the FEC data blocks MUST be the same. If the size of the video frame is not the multiple of the FEC data block size, then 0s MUST be padded to the end to make it the FEC data block size. The padding 0s are only used for calculating the FEC metadata packet and reconstructing the lost video data packet and are not sent over the network. The actual size of the last video data packet without padding might be smaller than the FEC data block size.

The actual size of the last video packet without padding is set in the HiLPL field and the LastPacketLengthLo field of the FEC RTP payload header. When the last video packet is

reconstructed, the receiver MUST strip out the padding 0s using the HiLPL and the LastPacketLengthLo fields.

The FEC data block size MUST be smaller than the MTU size. The most common MTU size is the Ethernet MTU, which is 1500 bytes.

To calculate the FEC metadata packet of FEC version 0. DV= 00 :

1. Initialize an FEC result buffer with all zeros. The size of the buffer MUST be the FEC data block size.
2. For each video data packet and for each byte in the correspondent FEC data block, do a bitwise XOR with the correspondent byte in the FEC result buffer and save the result into the result buffer.
3. After completing the calculation for all the data packets, the FEC result buffer is the resulting FEC metadata.

Reconstructing a lost video data packet is done by performing a byte-by-byte XOR operation on all the received video data packets and the FEC metadata packet. The result is the lost video data packet. If this data packet is the last data packet, only the first N bytes (N is specified by HiLPL field and LastPacketLengthLo field in the FEC RTP payload header) are actual data. The rest of the buffer is the padding and MUST be thrown away.

For FEC Version 1 (DV = '01'), the algorithm allows multiple FEC packets to protect the data packets of a video frame. All FEC packets are appended and transmitted immediately after the data packets pertaining to the video frame they protect. The first FEC packet is compatible with the default RTVideo FEC RTP Payload used when DV = '00'. In this case, the FEC packet is a simple XOR packet of all the data packets pertaining to the video frame it protects described in the algorithm above. . The subsequent FEC packets are generated/decoded by a Microsoft proprietary algorithm. <2>

2.2.5.2 FEC Metadata Packet Usage <2>

The FEC metadata packet MUST follow the video data packets.

2.2.5.3 RTP Header Usage

The FEC metadata packet is also encapsulated into an RTP packet. The RTP packet MUST use the same numbering space as the rest of the video data packets for the RTP sequence number field. Refer to [\[MS-RTP\]](#) for more details about RTP sequence number field.

2.2.6 Parsing RTVideo Packets

The M, M2, M3, E, and DV fields can be used to determine the RTP payload format type of a RTVideo packet.

Field Values	RTP Payload Format Type
--------------	-------------------------

M=0	Basic RTP Payload Format
M=1, M2=0	Extended RTP Payload Format
M=1, M2=1, E=0	Extended 2 RTP Payload Format (non FEC)
M=1, M2=1, M3=0, E=1, DV=00 or 01 <2>	FEC RTP Payload Format

2.2.7 SP-Frame and Cached Frame mechanisms

Both encoder and decoder may periodically cache decoded video frames. The cached video frame is stored in a dedicated memory location and in addition to the current reference decoded I and P frames. The caching mechanism is done in a synchronized fashion meaning that the video encoder and the video decoder must have a copy of the same decoded video frame in their cache whenever the encoder has completed encoding a video frame and the decoder is about to decode the same video frame. The cache frame is signaled in the packetized video bitstream by means of the ‘C’ field in the Packet Payload Format as described in sections [2.2.2](#), [2.2.3](#) and [2.2.4](#) above. The encoder and decoder use only one cached frame at a time so the data of the previous cached frame may be discarded whenever a new frame is cached. When the encoder receives a packet loss event reported by the decoder, it can choose to encode the next P frame using the cached frame as reference as opposed to the previous P frame or the previous I frame. In this case, the P frame is called a Super-P (SP) frame as it is predicted from the latest cached frame and not from the previous P frame or the previous I frame. The presence of an **SP-frame** in the packetized video bitstream is signaled by the ‘SP’ field in the Packet Payload Format as described in section [2.2.2](#), [2.2.3](#) and [2.2.4](#) above. Upon receiving an SP frame, the decoder should decode the video frame using the cached frame reference.

3 Protocol Details

3.1 Sender Details

This section covers the role of the sender of RTVideo frames.

3.1.1 Abstract Data Model

The RTVideo sender specifies the packetization process. When the sender receives a video frame from higher layers, the RTVideo sender fragments the video frame into multiple data fragments and then adds the RTP payload format header before each packet.

3.1.1.1 Choice of RTP Payload Format

The RTVideo receiver MAY not be able to decode a video frame if the reference frame is missing, or the receiver MAY manage to decode the frame, but with undesirable video artifacts.

The RTVideo Extended RTP Payload Format provides a way to reduce video artifacts by detecting and dropping the video frames for which reference frames are missing. This not only reduces the undesirable artifacts but also reduces CPU usage. The RTVideo Extended RTP Payload Format **SHOULD** be used.

If protection against packet loss is desired, and the application can afford the cost of extra bandwidth, the RTVideo FEC RTP Payload Format **MAY** be used together with the RTVideo Extended RTP Payload Format.

When minimizing resource use is desired, or if reducing the video artifacts is desired, the RTVideo Basic RTP Payload Format **MAY** be used.

The RTVideo Extended 2 RTP Payload Format is reserved for future extensions. It **MUST NOT** be used.

3.1.1.2 Fragmenting Video Frames

3.1.1.2.1 Maximum Video Fragment Size

The video frame **MUST** be fragmented in such a way that the size of a fragment plus the overhead of all layers does not exceed the MTU size.

The overhead **MUST** include at least the:

- RTP payload format header.
- RTP header.
- transport protocol header. For example, UDP header (8 bytes) or TCP header (20 bytes).
- IP protocol header.

Upper protocol layers **MAY** have different limits on the payload size. To prevent payload size from exceeding any limit on any upper layer, [MS-RTVPPF] requires the video fragment size **MUST** be smaller than 1200 bytes.

3.1.1.2.2 Additional Requirement for FEC

The RTVideo FEC RTP Payload Format requires that all the video data packets except the last one **MUST** have the same size.

3.1.1.3 Understanding the Sequence Header

The sequence header carries enough information for the RTVideo receiver to decode a group of video frames. Because of this, [MS-RTVPPF] requires that the sequence header **MUST** be present for the first video data packet of an RTVideo **I-frame**.

In addition to the semantics described in the SMPTE 421M standard [\[SMPTE-VC-1\]](#), the semantics of the MAX_CODED_WIDTH and MAX_CODED_HEIGHT fields in the Sequence Header is extended to represent the original sample aspect ratio (pixel aspect ratio) of the video frames. Once a video frame has been received and decoded using CODED_WIDTH and CODED_HEIGHT information in the Entry Point Header, the display process in the receiving end-point SHOULD use the MAX_CODED_WIDTH and MAX_CODED_HEIGHT field values to reconstruct a video frame with horizontal and vertical dimensions agreeing with the original sample aspect ratio. [<1>](#)

3.1.1.4 Other Requirements

The following rules MUST be followed when packetizing a RTVideo frame:

- The RTP sequence numbers MUST be continuous across video frames.
- The first packet of a video frame MUST have the F (first packet flag) bit set to 1.
- The last data packet of a video frame MUST have the L (last packet flag) bit set to 1.
- The last packet of a video frame (either the data packet or the FEC metadata packet) MUST have the RTP M bit set to 1.
- The PacketNumber, HiPN, LastPacketLength, HiLPL, DV and EndOffset fields for all FEC metadata packets (up to 31) MUST be set correctly. For each FEC packet, If 'DV=01', then FECPacketNumber MUST be set correctly .
- The sequence header MUST be present for the first packet of an **I-frame**. In other words, the S bit MUST be set to 1.
- The HiFC, FrameCounter, HiRFC, and RefFrameCounter fields MUST be set to the correct frame counter and reference frame counter for the data packet and MUST be set to 0 for the FEC metadata packet.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Send a RTVideo Frame

Whenever higher layers send an RTVideo frame, the video frame MUST be fragmented if it does not fit in one video data packet and the appropriate RTP Payload Format MUST be used to packetize the video fragments.

3.1.5 Message Processing Events and Sequencing Rules

An RTP payload format MUST be selected to send a video frame as specified in [section 3.1.1.1](#).

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Receiver Details

This section covers the role of the receiver of RTVideo packets.

3.2.1 Abstract Data Model

1. When receiving each video packet, validation SHOULD be done considering the following factors:
 - The RTP sequence numbers MUST be continuous.
 - The first packet of a video frame MUST have the F (first packet flag) bit set to 1.
 - The last data packet MUST have the L (last packet flag) bit set to 1.
 - The last packet of the frame (either the data packet or the FEC metadata packet) MUST have the RTP M bit set to 1.
 - If FEC metadata packet(s) are present, the PacketNumber, HiPN, LastPacketLength, HiLPL, and EndOffset fields MUST be set correctly. For each FEC packet, If 'DV=01', then FECPacketNumber MUST be set correctly.
 - If the I bit is set to 1 (**I-frame**) and the F bit is set to 1 (the first packet), then the sequence header MUST be present; in other words, the S bit MUST be set to 1.
2. If one or more video data packets for the video frame are not received and the lost video data packet cannot be reconstructed by FEC or FEC is not used, all the video packets of the video frame MUST be dropped.
3. If the RTVideo Extended RTP Payload Format is used, the HiFC, FrameCounter, HiRFC, and RefFrameCounter fields MAY be used to reduce video artifact.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Receive a Video Packet

Whenever a higher-layer component receives a video packet, the video packet **MUST** be parsed as specified in 2.2. Validation **SHOULD** be done as specified in 3.2.1.

If the video packet is the last data packet and all the data packets are received, the video frame can be constructed by concatenating all video payloads in the video packets.

If video data packet loss is detected, and an FEC metadata packet is present, the lost data packet can be reconstructed by using the FEC algorithm.

If this video packet belongs to a new frame and the previous video frame cannot be constructed through the previous steps, all video packets of the previous video frame **SHOULD** be dropped.

3.2.5 Message Processing Events and Sequencing Rules

When receiving a video packet, the video packet **SHOULD** be processed as specified in [section 3.2.4.1](#).

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Basic RTP Payload Format Examples

4.1.1 I-Frame

The frame is packetized as four data packets.

4.1.1.1 First Packet

Payload header bytes in network byte order:

0x4F, 0x16, 0x25, 0x00, 0x00, 0x01, 0x0F, 0xC2, 0x86, 0x0A, 0xF0, 0x8F, 0x88, 0x80,
0x00, 0x00, 0x01, 0x0E, 0x48, 0x04, 0x2B, 0xC2, 0x3C, 0x80

The payload header contains fields of the following values:

M=0, C=1, SP=0, L=0, O=1, I=1, S=1, F=1

Sequence Header Length=0x16

Sequence Header Bytes=0x25, 0x00 ...

4.1.1.2 Second Packet

Payload header bytes in network byte order:
0x4C

The payload header contains fields of the following values:
M=0, C=1, SP=0, L=0, O=1, I=1, S=0, F=0

4.1.1.3 Last Packet

Payload header bytes in network byte order:
0x5C

The payload header contains fields of the following values:
M=0, C=1, SP=0, L=1, O=1, I=1, S=0, F=0

4.1.2 SP-Frame

The frame is packetized as 4 data packets.

4.1.2.1 First Packet

Payload header bytes in network byte order:
0x69

The payload header contains fields of the following values:
M=0, C=1, SP=1, L=0, O=1, I=0, S=0, F=1

4.1.2.2 Second Packet

Payload header bytes in network byte order:
0x68

The payload header contains fields of the following values:
M=0, C=1, SP=1, L=0, O=1, I=0, S=0, F=0

4.1.2.3 Last Packet

Payload header bytes in network byte order:
0x78

The payload header contains fields of the following values:
M=0, C=1, SP=1, L=1, O=1, I=0, S=0, F=0

4.1.3 P-Frame or B-Frame

The frame is packetized as a single data packet.

4.1.3.1 First Packet/LastPacket

Payload header bytes in network byte order:
0x19

The payload header contains fields of the following values:
M=0, C=0, SP=0, L=1, O=1, I=0, S=0, F=1

4.2 Extended RTP Payload Format Examples

4.2.1 I-Frame

The frame is packetized as three or more data packets.

4.2.1.1 First Packet

Payload header bytes in network byte order:
0xCF, 0x00, 0x00, 0x00, 0x16, 0x25, 0x00, 0x00, 0x01, 0x0F, 0xC2, 0x86, 0x0A, 0xF0,
0x8F, 0x88, 0x80, 0x00, 0x00, 0x01, 0x0E, 0x48, 0x04, 0x2B

The payload header contains fields of the following values:
M=1, C=1, SP=0, L=0, O=1, I=1, S=1, F=1; M2=0, HiFEC=0, HiFC=0, DV=0, E=0,

FrameCounter=0x0

RefFrameCounter=0x0

Sequence Header Length=0x16

Sequence Header Bytes=0x27, 0x00 ...

4.2.1.2 Second Packet

Payload header bytes in network byte order:
0xCC, 0x00, 0x00, 0x00

The payload header contains fields of the following values:
M=1, C=1, SP=0, L=0, O=1, I=1, S=0, F=0; M2=0, HiFEC=0, HiFC=0, DV=0, E=0,

FrameCounter=0x0

RefFrameCounter=0x0

4.2.1.3 Last Packet

Payload header bytes in network byte order:
0xDC, 0x00, 0x00, 0x00

The payload header contains fields of the following values:

M=1, C=1, SP=0, L=1, O=1, I=1, S=0, F=0; M2=0, HiFEC=0, HiFC=0, DV=0, E=0,

FrameCounter=0x0

RefFrameCounter=0x0

4.2.2 P-Frame

The frame is packetized as a single data packet.

4.2.2.1 First Packet/Last Packet

Payload header bytes in network byte order:

0x99, 0x00, 0x01, 0x00

The payload header contains fields of the following values:

M=1, C=0, SP=0, L=1, O=1, I=0, S=0, F=1; M2=0, HiFEC=0, HiFC=0, DV=0, E=0,

FrameCounter=0x1

RefFrameCounter=0x0

4.2.3 SP-Frame

The frame is packetized as three or more data packets.

4.2.3.1 First Packet

Payload header bytes in network byte order:

0xE9, 0x00, 0x0F, 0x00

The payload header contains fields of the following values:

M=1, C=1, SP=1, L=0, O=1, I=0, S=0, F=1; M2=0, HiFEC=0, HiFC=0, DV=0, E=0,

FrameCounter=0x0F

RefFrameCounter=0x0

4.2.3.2 Second Packet

Payload header bytes in network byte order:

0xE8, 0x00, 0x0F, 0x00

The payload header contains fields of the following values:

M=1, C=1, SP=1, L=0, O=1, I=0, S=0, F=0; M2=0, HiFEC=0, HiFC=0, DV=0, E=0,

FrameCounter=0x0F

RefFrameCounter=0x0

4.2.3.3 Last Packet

Payload header bytes in network byte order:

0xF8, 0x00, 0x0F, 0x00

The payload header contains fields of the following values:

M=1, C=1, SP=1, L=1, O=1, I=0, S=0, F=0; M2=0, HiFEC=0, HiFC=0, DV=0, E=0,

FrameCounter=0xF

RefFrameCounter=0x0

4.2.4 B-Frame

The frame is packetized as a single data packet.

4.2.4.1 First Packet/Last Packet

Payload header bytes in network byte order:

0x99, 0x00, 0x01, 0x11

The payload header contains fields of the following values:

M=1, C=0, SP=0, L=1, O=1, I=0, S=0, F=1; M2=0, HiFEC=0, HiFC=0, DV=0, E=0,

FrameCounter=0x1

RefFrameCounter=0x11, or RefFrameCounterDelta1=0x1, RefFrameCounterDelta2=0x1

RefFrameCounter1=0x0, RefFrameCounter2=0x0

4.3 FEC RTP Payload Format Examples

4.3.1 I-Frame

The frame is packetized as 4 data packets and 1 FEC metadata packet.

4.3.1.1 FEC Metadata Packet (FEC Version 0)

Payload header bytes in network byte order:

0xCC, 0x 81, 0x00, 0x00, 0x00, 0x04, 0x60, 0x84

The payload header contains fields of the following values:

M=1, C=1, SP=0, L=0, O=1, I=1, S=0, F=0; M2=1, HiFEC=0, HiFC=0, DV=0, E=1

FrameCounter=0x0

RefFrameCounter=0x0

M3=0, HiPN=0, Reserved=0

PacketNumberNo=0x4

HiLPL=3, EndOffset=0

LastPacketLengthLo=0x84

4.3.1.2 FEC Metadata Packet (FEC Version 1) <1>

Payload header bytes in network byte order:

0xCC, 0x 83, 0x00, 0x00, 0x03, 0x04, 0x60, 0x84

The payload header contains fields of the following values:

M=1, C=1, SP=0, L=0, O=1, I=1, S=0, F=0; M2=1, HiFEC=0, HiFC=0, DV=1, E=1

FrameCounter=0x0

RefFrameCounter=0x0

M3=0, HiPN=0, FEC PacketsNumber = 3

PacketNumberNo=0x4

HiLPL=3, EndOffset=0

LastPacketLengthLo=0x84

4.3.2 SP-Frame

The frame is packetized as three data packets and one FEC metadata packet.

4.3.2.1 FEC Metadata Packet

Payload header bytes in network byte order:

0xE8, 0x81, 0x10, 0x00, 0x00, 0x03, 0x60, 0xDF

The payload header contains fields of the following values:

M=1, C=1, SP=1, L=0, O=1, I=0, S=0, F=0; M2=1, HiFEC=0, HiFC=0, DV=0, E=1

FrameCounter=0x10

RefFrameCounter=0x0

M3=0, HiPN=0, Reserved=0

PacketNumberNo=0x3

HiLPL=3, EndOffset=0

LastPacketLengthLo=0xDF

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following versions of the Microsoft product:

- Microsoft® Office Communications Server 2007
- Microsoft® Office Communications Server 2007 R2
- Microsoft® Office Communicator 2007
- Microsoft® Office Communicator 2007 R2

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies the behavior of any of the previously listed Microsoft products in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that any of the previously listed Microsoft products do not follow the prescription.

<1> [Section 3.1.1.3](#), [Section 4.3.1.2](#): Microsoft Office Communications Server 2007, Microsoft Office Communicator 2007: This behavior is not supported.

<2> [Section 2.2.5](#), [Section 2.2.5.1](#), [Section 2.2.5.2](#), [Section 2.2.6](#): Microsoft Office Communications Server 2007, Microsoft Office Communicator 2007: **DV** (2 bits): FEC version number. The value ranges from 0 to 3. The version number **MUST** be 0 in the RTVideo FEC RTP Payload Format.. **Reserved/FECPacketsNumber** (5 bits): Reserved field for future use. It **MUST** be set to 0. **The sender MUST NOT send** more than one FEC metadata packets for each RTVideo frame

7 Index

A

Abstract data model, 17, 20

Applicability statement, 8

B

Basic RTP Payload Format examples, 21

C

Choice of RTP payload format, 17

E

Extended RTP Payload Format examples,
23

F

FEC RTP Payload Format examples, 25

Fragmenting video frames, 18

G

Glossary, 6

H

Higher-layer triggered events, 19, 21

I

Index of security parameters, 27

Informative references, 7

Initialization, 19, 20

M

Message processing events and sequencing
rules, 19, 21

Message syntax, 9

Messages, 9

Microsoft Office Communications Server
2007 behavior, 27

N

Normative references, 7

O

Other local events, 20, 21

Other requirements, 19

P

Prerequisites/preconditions, 8

Protocol details, 17

Protocol examples, 21

Protocol overview (synopsis), 7

R

Receive a video packet, 21

Receiver details, 20

References, 7

Relationship to other protocols, 8

RTP header usage, 10

RTVideo Basic RTP Payload Format, 10

RTVideo Extended 2 RTP Payload
Format, 13

RTVideo Extended RTP Payload Format,
11

RTVideo FEC RTP Payload Format, 14

S

Security, 27

Security considerations for implementers,
27

Send a RTVideo frame, 19

Sender details, 17

Standards assignments, 9

T

Timer events, 20, 21

Timers, 19, 20

Transport, 9

U

Understanding the sequence header, 18

V

Vendor-extensible fields, 9

Versioning and capability negotiation, 8