Internet Engineering Task Force (IETF)
Request for Comments: 8285
Obsoletes: 5285
Category: Standards Track
ISSN: 2070-1721

D. Singer
Apple, Inc.
H. Desineni
Qualcomm
R. Even, Ed.
Huawei Technologies

October 2017

A General Mechanism for RTP Header Extensions

Abstract

This document provides a general mechanism to use the header extension feature of RTP (the Real-time Transport Protocol). It provides the option to use a small number of small extensions in each RTP packet, where the universe of possible extensions is large and registration is decentralized. The actual extensions in use in a session are signaled in the setup information for that session. This document obsoletes RFC 5285.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8285.

Table of Contents

1. Introduction

The RTP specification [RFC3550] provides a capability to extend the RTP header. Section 5.3.1 of [RFC3550] defines the header extension format and rules for its use. The existing header extension method permits at most one extension per RTP packet, identified by a 16-bit identifier and a 16-bit length field specifying the length of the header extension in 32-bit words.

This mechanism has two conspicuous drawbacks. First, it permits only one header extension in a single RTP packet. Second, the specification gives no guidance as to how the 16-bit header extension identifiers are allocated to avoid collisions.

This specification removes the first drawback by defining a backward-compatible and extensible means to carry multiple header extension elements in a single RTP packet. It removes the second drawback by defining that these extension elements are named by URIs, defining an IANA registry for extension elements defined in IETF specifications, and providing a Session Description Protocol (SDP) method for mapping between the naming URIs and the identifier values carried in the RTP packets.

This header extension applies to RTP/AVP (the Audio/Visual Profile) and its extensions.

This document obsoletes [RFC5285] and removes a limitation from RFC 5285 that did not allow sending both one-byte and two-byte header extensions in the same RTP stream.

2. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Design Goals

The goal of this design is to provide a simple mechanism whereby multiple identified extensions can be used in RTP packets, without the need for formal registration of those extensions but nonetheless avoiding collisions.

This mechanism provides an alternative to the practice of burying associated metadata into the media format bitstream. This has often been done in media data sent over fixed-bandwidth channels. Once

this is done, a decoder for the specific media format needs to extract the metadata. Also, depending on the media format, the metadata can be added at the time of encoding the media so that the bit-rate used for the metadata is taken into account. But the metadata can be unknown at that time. Inserting metadata at a later time can cause a decode and re-encode to meet bit-rate requirements.

In some cases, a more appropriate and higher-level mechanism may be available, and if so, it can be used. For cases where a higher-level mechanism is not available, it is better to provide a mechanism at the RTP level than to have the metadata be tied to a specific form of media data.

4. Packet Design

4.1. General

The following design is fit into the "header extension" of the RTP extension, as described above.

The presence and format of this header extension and its contents are negotiated or defined out of band, such as through signaling (see below for SDP signaling). The 16-bit identifier for the two forms of the RTP extension defined here is only an architectural constant (e.g., for use by network analyzers); it is the negotiation/definition (e.g., in SDP) that is the definitive indication that this header extension is present.

The RTP specification [RFC3550] states that RTP "is designed so that the header extension may be ignored by other interoperating implementations that have not been extended." The intent of this restriction is that RTP header extensions MUST NOT be used to extend RTP itself in a manner that is backward incompatible with non-extended implementations. For example, a header extension is not allowed to change the meaning or interpretation of the standard RTP header fields or of the RTP Control Protocol (RTCP). Header extensions MAY carry metadata in addition to the usual RTP header information, provided the RTP layer can function if that metadata is missing. For example, RTP header extensions can be used to carry data that’s also sent in RTCP, as an optimization to lower latency, since they’ll fall back to the original non-optimized behavior if the header extension is not present. The use of header extensions to convey information that will, if missing, disrupt the behavior of a higher-layer application that builds on top of RTP is only acceptable if this doesn’t affect interoperability at the RTP layer. For example, applications that use the SDP BUNDLE extension with the Media Identification (MID) RTP header extension [SDP-BUNDLE] to correlate RTP streams with SDP "m=" lines likely won’t work with full

functionality if the MID is missing, but the operation of the RTP layer of those applications will be unaffected. Support for RTP header extensions based on this memo is negotiated using, for example, SDP Offer/Answer [RFC3264]; intermediaries aware of the RTP header extensions are advised to be cautious when removing or generating RTP header extensions. See Section 4.7 of [RFC7667].

The RTP header extension is formed as a sequence of extension elements, with possible padding. Each extension element has a local identifier and a length. The local identifiers MAY be mapped to a larger namespace in the negotiation (e.g., session signaling).

4.1.1. Transmission Considerations

As is good network practice, data should only be transmitted when needed. The RTP header extension SHOULD only be present in a packet if that packet also contains one or more extension elements, as defined here. An extension element SHOULD only be present in a packet when needed; the signaling setup of extension elements indicates only that those elements can be present in some packets, not that they are in fact present in all (or indeed, any) packets.

Some general considerations for getting the header extensions delivered to the receiver are as follows:

1. The probability for packet loss and burst loss determines how many repetitions of the header extensions will be required to reach a targeted delivery probability, and if burst loss is likely, what distribution would be needed to avoid losing all repetitions of the header extensions in a single burst.

2. If a set of packets are all needed to enable decoding, there is commonly no reason for including the header extension in all of these packets, as they share fate. Instead, at most one instance of the header extension per independently decodable set of media data would be a more efficient use of the bandwidth.

3. How early the header extension item information is needed, from the first received RTP data or only after some set of packets are received, can guide whether the header extension(s) should be (1) in all of the first N packets or (2) included only once per set of packets -- for example, once per video frame.

4. The use of RTP-level robustness mechanisms, such as RTP retransmission [RFC4588] or Forward Error Correction (e.g., [RFC5109]) may treat packets differently from a robustness perspective, and header extensions should be added to packets that get a treatment corresponding to the relative importance of receiving the information.

As a summary, the number of header extension transmissions should be tailored to a desired probability of delivery, taking the receiver population size into account. For the very basic case, N repetitions of the header extensions should be sufficient but may not be optimal. N is selected so that the header extension target delivery probability reaches 1-P^N, where P is the probability of packet loss. For point-to-point or small receiver populations, it might also be possible to use feedback, such as RTCP, to determine when the information in the header extensions has reached all receivers and stop further repetitions. Feedback that can be used includes the RTCP Extended Report (XR) Loss RLE Report Block [RFC3611], which will indicate successful delivery of particular packets. If the RTP/AVPF transport-layer feedback messages for generic NACK [RFC4585] are used, they can indicate failure to deliver an RTP packet with the header extension, thus indicating the need for further repetitions. The normal RTCP report blocks can also provide an indicator of successful delivery, if no losses are indicated for a reporting interval covering the RTP packets with the header extension. Note that loss of an RTCP packet reporting on an interval where RTP header extension packets were sent does not necessarily mean that the RTP header extension packets themselves were lost.

4.1.2. Header Extension Type Considerations

Each extension element in a packet has a local identifier (ID) and a length. The local identifiers present in the stream MUST have been negotiated or defined out of band. There are no static allocations of local identifiers. Each distinct extension MUST have a unique ID. The ID value 0 is reserved for padding and MUST NOT be used as a local identifier.

An extension element with an ID value equal to 0 MUST NOT have an associated length field greater than 0. If such an extension element is encountered, its length field MUST be ignored, processing of the entire extension MUST terminate at that point, and only the extension elements present prior to the element with ID 0 and a length field greater than 0 SHOULD be considered.

There are two variants of the extension: one-byte and two-byte headers. Since it is expected that (a) the number of extensions in any given RTP session is small and (b) the extensions themselves are

small, the one-byte header form is preferred and MUST be supported by all receivers. A stream MUST contain only one-byte headers or only two-byte headers unless it is known that all recipients support mixing, by either SDP Offer/Answer [RFC3264] negotiation (see Section 6) or out-of-band knowledge. Each RTP packet with an RTP header extension following this specification will indicate whether it contains one-byte or two-byte header extensions through the use of the "defined by profile" field. Extension element types that do not match the header extension format, i.e., one-byte or two-byte, MUST NOT be used in that RTP packet. Transmitters SHOULD NOT use the two-byte header form when all extensions are small enough for the one-byte header form. Transmitters that intend to send the two-byte form SHOULD negotiate the use of IDs above 14 if they want to let the receivers know that they intend to use the two-byte form -- for example, if the RTP header extension is longer than 16 bytes. A transmitter may be aware that an intermediary may add RTP header extensions; in this case, the transmitter SHOULD use the two-byte form.

A sequence of extension elements, possibly with padding, forms the header extension defined in the RTP specification. There are as many extension elements as will fit in the RTP header extension, as indicated by the RTP header extension length. Since this length is signaled in full 32-bit words, padding bytes are used to pad to a 32-bit boundary. The entire extension is parsed byte by byte to find each extension element (no alignment is needed), and parsing stops (1) at the end of the entire header extension or (2) in the "one-byte headers only" case, on encountering an identifier with the reserved value of 15 -- whichever happens earlier.

In both forms, padding bytes have the value of 0 (zero). They MAY be placed between extension elements, if desired for alignment, or after the last extension element, if needed for padding. A padding byte does not supply the ID of an element, nor does it supply the length field. When a padding byte is found, it is ignored, and the parser moves on to interpreting the next byte.

Note carefully that the one-byte header form allows for data lengths between 1 and 16 bytes, by adding 1 to the signaled length value (thus, 0 in the length field indicates that one byte of data follows). This allows for the important case of 16-byte payloads. This addition is not performed for the two-byte headers, where the length field signals data lengths between 0 and 255 bytes.

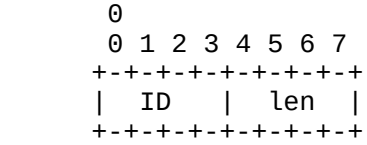
Use of RTP header extensions will reduce the efficiency of RTP header compression, since the header extension will be sent uncompressed unless the RTP header compression module is updated to recognize the extension header. If header extensions are present in some packets

but not in others, this can also reduce compression efficiency by requiring an update to the fixed header to be conveyed when header extensions start or stop being sent. The interactions of the RTP header extension and header compression are explored further in [RFC2508] and [RFC3095].

4.2. One-Byte Header

In the one-byte header form of extensions, the 16-bit value required by the RTP specification for a header extension, labeled in the RTP specification as "defined by profile", MUST have the fixed bit pattern 0xBEDE (the pattern was picked for the trivial reason that the first version of this specification was written on May 25th -- the feast day of the Venerable Bede).

Each extension element MUST start with a byte containing an ID and a length:

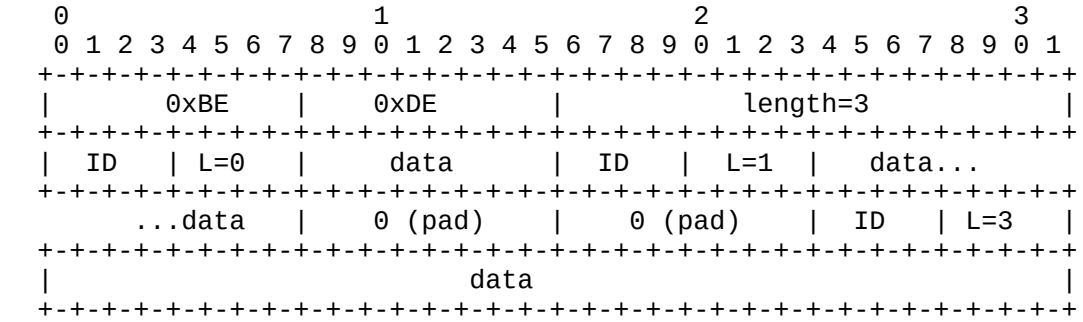


The 4-bit ID is the local identifier of this element in the range 1-14 inclusive. In the signaling section, this is referred to as the valid range.

The local identifier value 15 is reserved for a future extension and MUST NOT be used as an identifier. If the ID value 15 is encountered, its length field MUST be ignored, processing of the entire extension MUST terminate at that point, and only the extension elements present prior to the element with ID 15 SHOULD be considered.

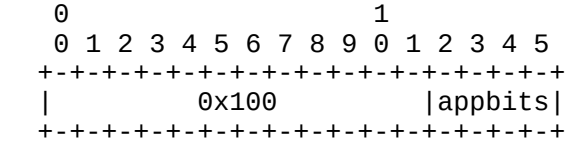
The 4-bit length is the number, minus one, of data bytes of this header extension element following the one-byte header. Therefore, the value zero (0) in this field indicates that one byte of data follows, and a value of 15 (the maximum) indicates element data of 16 bytes. (This permits carriage of 16-byte values, which is a common length of labels and identifiers, while losing the possibility of zero-length values, which would often be padded anyway.)

An example header extension, with three extension elements and some padding, follows:



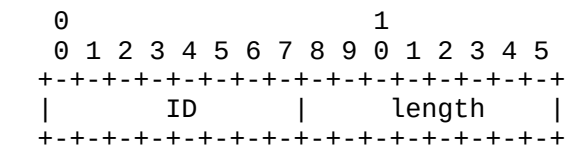
4.3. Two-Byte Header

In the two-byte header form, the 16-bit value defined by the RTP specification for a header extension, labeled in the RTP specification as "defined by profile", is defined as shown below.



The appbits field is 4 bits that are application dependent and MAY be defined to be any value or meaning; this topic is outside the scope of this specification. For the purposes of signaling, this field is treated as a special extension value assigned to the local identifier 256. If no extension has been specified through configuration or signaling for this local identifier value (256), the appbits field SHOULD be set to all 0s (zeros) by the sender and MUST be ignored by the receiver.

Each extension element starts with a byte containing an ID and a byte containing a length:

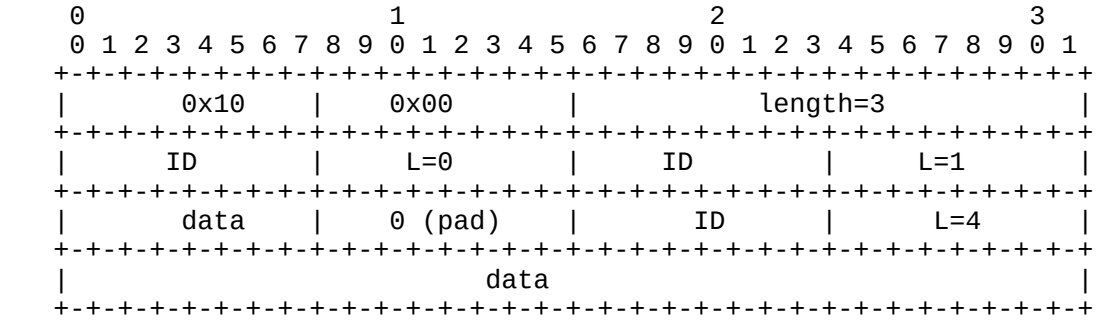


The 8-bit ID is the local identifier of this element in the range 1-255 inclusive. In the signaling section, the range 1-256 is referred to as the valid range, with the values 1-255 referring to

extension elements and the value 256 referring to the 4-bit appbits field (above). Note that there is one ID space for both the one-byte form and the two-byte form. This means that the lower values (1-14) can be used in the 4-bit ID field in the one-byte header format with the same meanings.

The 8-bit length field is the length of extension data in bytes, not including the ID and length fields. The value zero (0) indicates that there is no subsequent data.

An example header extension, with three extension elements and some padding, follows:



5. SDP Signaling Design

The indication of the presence of this extension, and the mapping of local identifiers used in the header extension to a larger namespace, MUST be performed out of band -- for example, as part of an SDP Offer/Answer [RFC3264]. This section defines such signaling in SDP.

A usable mapping MUST use IDs in the valid range, and each ID in this range MUST be used only once for each media section (or only once if the mappings are session level). Mappings that do not conform to these rules MAY be presented, for instance, during SDP Offer/Answer [RFC3264] negotiation as described in the next section, but remapping to conformant values is necessary before they can be applied.

Each extension is named by a URI. That URI MUST be absolute; it precisely identifies the format and meaning of the extension. URIs that contain a domain name SHOULD also contain a month-date in the form mmyyyy. The definition of the element and assignment of the URI MUST have been authorized by the owner of the domain name on or very close to that date. (This avoids problems when domain names change ownership.) If the resource or document defines several extensions,

then the URI MUST identify the actual extension in use, e.g., using a fragment or query identifier (characters after a "#" or "?" in the URI).

Rationale: The use of URIs provides for a large, unallocated space and gives documentation on the extension. The URIs do not have to be dereferencable, in order to permit confidential or experimental use, or to cover the case when extensions continue to be used after the organization that defined them ceases to exist.

An extension URI with the same attributes MUST NOT appear more than once applying to the same stream, i.e., at session level or in the declarations for a single stream at media level. (The same extension can, of course, be used for several streams and can appear with different <extensionattributes> for the same stream.)

For extensions defined in RFCs, the URI used SHOULD be a URN starting with "urn:ietf:params:rtp-hdrext:" followed by a registered, descriptive name.

The registration requirements are detailed in Section 10 ("IANA Considerations").

An example where "avt-example-metadata" is the hypothetical name of a header extension might be:

urn:ietf:params:rtp-hdrext:avt-example-metadata

An example name not from the IETF might be:

http://example.com/082005/ext.htm#example-metadata

The mapping MAY be provided per media stream (in the media-level section(s) of SDP, i.e., after an "m=" line) or globally for all streams (i.e., before the first "m=" line, at session level). The definitions MUST be either all session level or all media level; it is not permitted to mix the two styles. In addition, as noted above, the IDs used MUST be unique in each media section of the SDP or unique in the session for session-level SDP declarations.

Each local identifier potentially used in the stream is mapped to an extension identified by a URI using an attribute of the form:

```
a=extmap:<value>["/"<direction>] <URI> <extensionattributes>
```

where

- <value> is the local identifier (ID) of this extension and is an integer in the valid range (0 is reserved for padding in both forms, and 15 is reserved in the one-byte header form, as noted above).
- <direction> is one of "sendonly", "recvonly", "sendrecv", or "inactive" (without the quotes) with relation to the device being configured.
- <URI> is a URI, as above.

The formal BNF syntax is presented in Section 8 of this specification.

Example:

```
a=extmap:1 http://example.com/082005/ext.htm#ttime

a=extmap:2/sendrecv http://example.com/082005/ext.htm#xmeta short
```

When SDP signaling is used for the RTP session, it is the presence of the "extmap" attribute(s) that is diagnostic that this style of header extensions is used, not the magic number ("BEDE" or "100") indicated above.

6. SDP Signaling for Support of Mixed One-Byte and Two-Byte Header Extensions

In order to allow for backward interoperability with systems that do not support the mixing of one-byte and two-byte header extensions, this document defines the "a=extmap-allow-mixed" Session Description Protocol (SDP) [RFC4566] attribute to indicate if the participant is capable of supporting this new mode. The attribute takes no value. This attribute can be used at the session level or the media level. A participant that proposes the use of this mode SHALL itself support the reception of mixed one-byte and two-byte header extensions.

If SDP Offer/Answer [RFC3264] is supported and used, the negotiation for mixed one-byte and two-byte extensions MUST be negotiated using SDP Offer/Answer per [RFC3264]. In the absence of negotiations using

SDP Offer/Answer -- for example, when declarative SDP is used -- mixed headers MUST NOT occur unless the transmitter has some (out-of-band) knowledge that all potential recipients support this mode.

The formal definition of this attribute is:

Name: extmap-allow-mixed

Value: None

Usage Level: session, media

Charset Dependent: No

Example:

a=extmap-allow-mixed

When doing SDP Offer/Answer [RFC3264], an offering client that wishes to use both one-byte and two-byte extensions MUST include the attribute "a=extmap-allow-mixed" in the SDP offer. If "a=extmap-allow-mixed" is present in the SDP offer, the answerer that supports this mode and wishes to use it SHALL include the "a=extmap-allow-mixed" attribute in the answer. In the cases where the attribute has been excluded, both clients SHALL NOT use mixed one-byte and two-byte extensions in the same RTP stream but MAY use the one-byte or two-byte form exclusively (see Section 4.1.2).

When used per [SDP-BUNDLE], this attribute is specified as the IDENTICAL category [SDP-MUX].

7. SDP Offer/Answer

The simple signaling described above for the "extmap" attribute MAY be enhanced in an SDP Offer/Answer [RFC3264] context, to permit:

- asymmetric behavior (extensions sent in only one direction),
- the offer of mutually exclusive alternatives, or
- the offer of more extensions than can be sent in a single session.

A direction attribute MAY be included in an "extmap"; without it, the direction implicitly inherits, of course, from the stream direction or is "sendrecv" for session-level attributes or extensions of "inactive" streams. The direction MUST be one of "sendonly", "recvonly", "sendrecv", or "inactive" as specified in [RFC3264].

Extensions, with their directions, MAY be signaled for an "inactive" stream. It is an error to use an extension direction incompatible with the stream direction (e.g., a "sendonly" attribute for a "recvonly" stream).

If an offer or answer contains session-level mappings (and hence no media-level mappings) and different behavior is desired for each stream, then the entire set of extension map declarations MAY be moved into the media-level section(s) of the SDP. (Note that this specification does not permit mixing global and local declarations, to make identifier management easier.)

If an extension map is offered as "sendrecv", explicitly or implicitly, and asymmetric behavior is desired, the SDP answer MAY be changed to modify or add direction qualifiers for that extension.

If an extension is marked as "sendonly" and the answerer desires to receive it, the extension MUST be marked as "recvonly" in the SDP answer. An answerer that has no desire to receive the extension or does not understand the extension SHOULD remove it from the SDP answer. An answerer MAY want to respond that he supports the extension and does not want to receive it at the moment, but he may indicate a desire to receive it in a future offer and will mark the extension as "inactive".

If an extension is marked as "recvonly" and the answerer desires to send it, the extension MUST be marked as "sendonly" in the SDP answer. An answerer that has no desire to, or is unable to, send the extension SHOULD remove it from the SDP answer. An answerer MAY want to respond that he supports this extension but has no intention of sending it now; he may indicate a desire to send it in a future offer by marking the extension as "inactive".

Local identifiers in the valid range inclusive in an offer or answer must not be used more than once per media section (including the session-level section). The local identifiers MUST be unique in an RTP session, and the same identifier MUST be used for the same offered extension in the answer. A session update MAY change the direction qualifiers of extensions being used. A session update MAY add or remove extension(s). Identifier values in the valid range MUST NOT be altered (remapped).

Note that, under this rule, the same local identifier cannot be used for two extensions for the same media, even when one is "sendonly" and the other "recvonly", as it would then be impossible to make either of them "sendrecv" (since renumbering is not permitted either).

If a party wishes to offer mutually exclusive alternatives, then multiple extensions with the same identifier in the extended range 4096-4351 MAY be offered. The answerer SHOULD select, at most, one of the offered extensions with the same identifier and remap it to a free identifier in the valid range for that extension to be usable.

Similarly, if more extensions are offered than can be fit in the valid range, identifiers in the range 4096-4351 MAY be offered; the answerer SHOULD choose those that are desired and remap them to a free identifier in the valid range.

An answerer may copy an "extmap" for an identifier in the extended range into the answer to indicate to the offerer that it supports that extension. Of course, such an extension cannot be used, since there is no way to specify it in an extension header. If needed, the offerer or answerer can update the session to assign a valid identifier to that extension URI.

Rationale: The range 4096-4351 for these negotiation identifiers is deliberately restricted to allow expansion of the range of valid identifiers in the future.

Either party MAY include extensions in the stream other than those negotiated, or those negotiated as "inactive" (for example, for the benefit of intermediate nodes). Only extensions that appeared with an identifier in the valid range in SDP originated by the sender can be sent.

Example (port numbers, RTP profiles, payload IDs, rtpmaps, etc. all omitted for brevity):

The offer:

```
a=extmap:1 URI-toffset
a=extmap:14 URI-obscure
a=extmap:4096 URI-gps-string
a=extmap:4096 URI-gps-binary
a=extmap:4097 URI-frametype
m=video
a=sendrecv
m=audio
a=sendrecv
```

The answerer is interested in receiving GPS in string format only on video but cannot send GPS at all. It is not interested in transmission offsets on audio and does not understand the URI-obscure extension. It therefore moves the extensions from session level to media level and adjusts the declarations:

```
m=video
a=sendrecv
a=extmap:1 URI-toffset
a=extmap:2/recvonly URI-gps-string
a=extmap:3 URI-frametype
m=audio
a=sendrecv
a=extmap:1/sendonly URI-toffset
```

When using [SDP-BUNDLE] to bundle multiple "m=" lines, the "extmap" attribute falls under the SPECIAL category of [SDP-MUX]. All the "m=" lines in a BUNDLE group are considered to be part of the same local identifier (ID) space. If an RTP header extension, i.e., a particular extension URI and configuration using <extensionattributes>, is offered in multiple "m=" lines that are part of the same BUNDLE group, it MUST use the same ID in all of these "m=" lines. Each "m=" line in a BUNDLE group can include different RTP header extensions allowing, for example, audio and video sources to use different sets of RTP header extensions. A difference in configuration using any of the <extensionattributes> is important. Unless an RTP header extension explicitly states otherwise, any such difference SHALL be communicated to all receivers and SHALL cause assignment of different IDs. An RTP header extension that does not follow this rule MUST explicitly define what would constitute compatible configurations that can be sent with the same ID. The directionality of the RTP header extensions in each "m=" line of the BUNDLE group is handled in the same way as handling for non-bundled "m=" lines. This allows for specifying different directionality for each of the repeated extension URIs in a BUNDLE group.

8. BNF Syntax

The syntax definition below uses ABNF according to [RFC5234]. The syntax element "URI" is defined in [RFC3986] (only absolute URIs are permitted here). The syntax element "extmap" is an attribute as defined in [RFC4566], i.e., "a=" precedes the "extmap" definition. Specific <extensionattributes> are defined by the specification that defines a specific extension name; there can be several.

Name: extmap

Value: extmap-value

Syntax:

```
extmap-value = mapentry SP extensionname
               [SP extensionattributes]

mapentry = "extmap:" 1*5DIGIT ["/" direction]

extensionname = URI

extensionattributes = byte-string

direction = "sendonly" / "recvonly" / "sendrecv" / "inactive"

URI = <Defined in RFC 3986>

byte-string = <Defined in RFC 4566>

SP = <Defined in RFC 5234>

DIGIT = <Defined in RFC 5234>
```

9. Security Considerations

This document defines only a place to transmit information; the security implications of each of the extensions must be discussed with those extensions.

Extension usage is negotiated using [RFC3264], so integrity protection and end-to-end authentication MUST be implemented. The security considerations of [RFC3264] MUST be followed to prevent, for example, extension-usage blocking.

Header extensions have the same security coverage as the RTP header itself. When the Secure Real-time Transport Protocol (SRTP) [RFC3711] is used to protect RTP sessions, the RTP payload can be

both encrypted and integrity protected, while the RTP header is either unprotected or integrity protected. In order to prevent DoS attacks (for example, by changing the header extension) integrity protection SHOULD be used. Lower-layer security protection such as Datagram Transport Layer Security (DTLS) [RFC6347] MAY be used. RTP header extensions can carry sensitive information for which participants in multimedia sessions want confidentiality. RFC 6904 [RFC6904] provides a mechanism that extends the mechanisms of SRTP to selectively encrypt RTP header extensions in SRTP.

The RTP application designer needs to consider their security needs, that includes cipher strength for SRTP packets in general and what that means for the integrity and confidentiality of the RTP header extensions. As defined by RFC 6904 [RFC6904], the encryption stream cipher for the header extension is dependent on the chosen SRTP cipher.

Other options for securing RTP are discussed in [RFC7201].

10. IANA Considerations

This document updates the references in three IANA registries to point to this document instead of RFC 5285, and updates and adds new SDP attributes in Sections 10.2 and 10.3, respectively.

10.1. Identifier Space for IANA to Manage

The mapping from the naming URI form to a reference to a specification is managed by IANA. Insertion into this registry is under the requirements of "Expert Review" as defined in [RFC8126].

IANA will also maintain a server that contains all of the registered elements in a publicly accessible space.

Here is the formal declaration to comply with the IETF URN sub-namespace specification [RFC3553].

- Registry name: RTP Compact Header Extensions
- Specification: RFC 5285 and RFCs updating RFC 5285
- Information required:

  A. The desired extension naming URI

  B. A formal reference to the publicly available specification

C. A short phrase describing the function of the extension

D. Contact information for the organization or person making the registration

For extensions defined in RFCs, the URI SHOULD be of the form urn:ietf:params:rtp-hdrext:, and the formal reference is the RFC number of the RFC documenting the extension.

- Review process: Expert Review is REQUIRED. The expert reviewer SHOULD check the following requirements:

  1. that the specification is publicly available;

  2. that the extension complies with the requirements of RTP, and this specification, for header extensions (specifically, that the header extension can be ignored or discarded without breaking the RTP layer);

  3. that the extension specification is technically consistent (in itself and with RTP), complete, and comprehensible;

  4. that the extension does not duplicate functionality in existing IETF specifications (including RTP itself) or other extensions already registered;

  5. that the specification contains a security analysis regarding the content of the header extension;

  6. that the extension is generally applicable -- for example, point-to-multipoint safe -- and the specification correctly describes limitations if they exist;

  7. that the suggested naming URI form is appropriately chosen and unique; and

  8. that for multiplexed "m=" lines [SDP-BUNDLE], any RTP header extension with differences in configurations of <extensionattributes> that do not require assignment of different IDs MUST explicitly indicate this and provide rules for what would constitute compatible configurations that can be sent with the same ID.

- Size and format of entries: A mapping from a naming URI string to a formal reference to a publicly available specification, with a descriptive phrase and contact information.

- Initial assignments: None

10.2. Registration of the SDP "extmap" Attribute

IANA has updated the registration of the "extmap" SDP attribute [RFC4566] in the "att-field (both session and media level)" subregistry of the "Session Description Protocol (SDP) Parameters" registry.

- Contact Name and email address: IETF, contacted via <mmusic@ietf.org> (or a successor address designated by the IESG)
- Attribute Name: extmap
- Attribute Syntax: See Section 8 of RFC 8285.
- Attribute Semantics: The details of appropriate values are given in RFC 8285.
- Usage Level: Media or session level
- Charset Dependent: No
- Purpose: Defines the mapping from the extension numbers used in packet headers into extension names.
- Offer/Answer (O/A) Procedures: See Section 7 of RFC 8285.
- MUX Category: SPECIAL
- Reference: RFC 8285

10.3. Registration of the SDP "extmap-allow-mixed" Attribute

IANA has registered one new SDP attribute in the "att-field (both session and media level)" subregistry of the "Session Description Protocol (SDP) Parameters" registry:

- Contact Name and email address: IETF, contacted via <mmusic@ietf.org> (or a successor address designated by the IESG)
- Attribute Name: extmap-allow-mixed
- Attribute Syntax: See Section 6 of RFC 8285.
- Attribute Semantics: See Section 6 of RFC 8285.
- Attribute Value: None
- Usage Level: Media or session level

- Charset Dependent: No
- Purpose: Negotiate the use of one byte and two bytes in the same RTP stream.
- O/A Procedures: See Section 6 of RFC 8285.
- MUX Category: IDENTICAL
- Reference: RFC 8285

11. Changes from RFC 5285

The major motivation for updating [RFC5285] was to allow having one-byte and two-byte RTP header extensions in the same RTP stream (but not in the same RTP packet). The support for this case is negotiated using a new SDP attribute, "extmap-allow-mixed", specified in this document.

The other major change is to update the requirement from the RTP specifications [RFC3550] and [RFC5285] that the header extension "is designed so that the header extension may be ignored." This is described in Section 4.1.

More text was added to Section 4.1.1 ("Transmission Considerations") to clarify when and how many times to send the RTP header extension to provide a higher probability of delivery.

The Security Considerations section was expanded.

The rest of the changes are editorial.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2508] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, DOI 10.17487/RFC2508, February 1999, <https://www.rfc-editor.org/info/rfc2508>.

[RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, DOI 10.17487/RFC3095, July 2001, <https://www.rfc-editor.org/info/rfc3095>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <https://www.rfc-editor.org/info/rfc3264>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <https://www.rfc-editor.org/info/rfc3711>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

[RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC6904] Lennox, J., "Encryption of Header Extensions in the Secure Real-time Transport Protocol (SRTP)", RFC 6904, DOI 10.17487/RFC6904, April 2013, <https://www.rfc-editor.org/info/rfc6904>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

12.2. Informative References

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", BCP 73, RFC 3553, DOI 10.17487/RFC3553, June 2003, <https://www.rfc-editor.org/info/rfc3553>.

[RFC3611] Friedman, T., Ed., Caceres, R., Ed., and A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, DOI 10.17487/RFC3611, November 2003, <https://www.rfc-editor.org/info/rfc3611>.

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <https://www.rfc-editor.org/info/rfc4585>.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <https://www.rfc-editor.org/info/rfc4588>.

[RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, DOI 10.17487/RFC5109, December 2007, <https://www.rfc-editor.org/info/rfc5109>.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, DOI 10.17487/RFC5285, July 2008, <https://www.rfc-editor.org/info/rfc5285>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <https://www.rfc-editor.org/info/rfc7201>.

[RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667, DOI 10.17487/RFC7667, November 2015, <https://www.rfc-editor.org/info/rfc7667>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[SDP-BUNDLE] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", Work in Progress, draft-ietf-mmusic-sdp-bundle-negotiation-39, August 2017.

[SDP-MUX] Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", Work in Progress, draft-ietf-mmusic-sdp-mux-attributes-16, December 2016.

Acknowledgments

Both Brian Link and John Lazzaro provided helpful comments on an initial draft of this document. Colin Perkins was helpful in reviewing and dealing with the details. The use of URNs for IETF-defined extensions was suggested by Jonathan Lennox, and Pete Cordell was instrumental in improving the padding wording. Dave Oran provided feedback and text in the review. Mike Dolan contributed the two-byte header form. Magnus Westerlund and Tom Taylor were instrumental in managing the registration text.

Authors' Addresses

David Singer
Apple, Inc.
1 Infinite Loop
Cupertino, CA 95014
United States of America

Phone: +1 408 996 1010
Email: singer@apple.com
URI: https://support.apple.com/quicktime

Harikishan Desineni
Qualcomm
10001 Pacific Heights Blvd.
San Diego, CA 92121
United States of America

Phone: +1 858 845 8996
Email: h3dnvb@gmail.com

Roni Even (editor)
Huawei Technologies
Tel Aviv
Israel

Email: Roni.even@huawei.com