AVT D. Singer
Internet-Draft Apple Computer Inc.
Intended status: Standards Track H. Desineni
Expires: August 30, 2007 Qualcomm
February 26, 2007

A general mechanism for RTP Header Extensions
draft-ietf-avt-rtp-hdrext-12.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on August 30, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

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 de-centralized. The actual extensions in use in a session are signaled in the setup information for that session.

Table of Contents

1. Introduction

The RTP Specification [RFC3550] provides a capability to extend the RTP header. It defines the header extension format and rules for its use in section 5.3.1. 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 labelled by URNs, defines an IANA registry for extension elements defined in IETF specifications, and an SDP method for mapping between the URNs and the identifier values carried in the RTP packets.

This header extension applies to the RTP/AVP profile and its extensions.

2. Requirements notation

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].

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 collision.

This mechanism provides an alternative to the practice of burying associated metadata into the media format bit stream. This has often been done in media data sent over fixed-bandwidth channels. Once this is done, a decoder for the specific media format is required to extract the metadata. Also, depending on the media format, the metadata may need to be added at the time of encoding the media so that the bit-rate required for the metadata is taken into account. But the metadata may not be known at that time. Inserting metadata at a later time can require a decode and re-encode to meet bit-rate requirements.

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

4. Packet Design

The following design is fit into the "header extension" of the RTP extension, as described above. The 16-bit value required by the RTP specification for a header extension, labelled in the RTP specification as "defined by profile", takes the fixed bit pattern 0xBEDE (the first draft of this specification was written on the feast day of the Venerable Bede).

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

This specification inherits the requirement from the RTP specification that the header extension "is designed so that the header extension may be ignored". To be specific, header extensions using this specification MUST only be used for data that can safely be ignored by the recipient without affecting interoperability. Examples might include meta-data that is additional to the usual RTP information.

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. Since it is expected that (a) the number of extensions in any given RTP session is small and (b) the extensions themselves are small, only 4 bits are allocated to each of these. The local identifiers may be mapped to a larger namespace in the negotiation (e.g. session signaling).

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

```
 0
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|  ID   |  len  |
+-+-+-+-+-+-+-+-+
```

The 4-bit ID is the local identifier of this element in the range 1-14 inclusive. The values present in the stream MUST have been negotiated or defined out-of-band. There are no static allocations of identifiers. Each distinct extension MUST have a unique ID.

The value 0 is reserved for padding and MUST NOT be used as an

identifier.

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

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. The first four bits of a padding byte are not interpreted as the ID of an ID/length pair, nor are the second four bits interpreted as a length field. When a padding byte is found it is ignored and the parser moves on to interpreting the next byte.

A sequence of extension elements, possibly with padding, forms the header extension defined in the RTP specification. There are as many extension elements as fit into the length as indicated in 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 required), and parsing stops at the earlier of the end of the entire header extension, or on encountering an identifier with the reserved value of 15.

An example header extension, with three extension elements, some padding, and including the required RTP fields, follows:

```
 0                   1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0xBE    |    0xDE       |           length=3            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID   | L=0   |     data      |  ID   |  L=1  |   data...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      ...data   |    0 (pad)    |    0 (pad)    |  ID   | L=3   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          data                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 may be present in some packets, not that they are in fact present in all (or indeed, any) packets.

5. SDP Signalling 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 a SIP offer/answer exchange using SDP. This section defines such signaling in SDP.

A usable mapping MUST use IDs in the range 1-14, and each ID in this range MUST be used only once for each media (or only once if the mappings are session level). Mappings which do not conform to these rules MAY be presented, for instance during offer/answer 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, and precisely identifies the format and meaning of the extension. In general, the URI SHOULD also be de-referencable by any system that sees or receives the SDP containing it. 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, gives documentation on the extension. The URIs are not required to be de-referencable, in order to permit confidential or experimental use, and to cover the case when extensions continue to be used after the organization that defined them ceases to exist.

An extension URI 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 may, of course, be used for several streams.)

For extensions defined in RFCs, the URI used SHOULD be a URN starting "urn:ietf:params:rtp-hdrext:" and followed by a registered, descriptive name. These URNs are managed by IANA. An example (this is only 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 (this is only an example) 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 for each stream type for a given media, or for the session for session level declarations.

Each local identifier potentially used in the stream is mapped to a string using an attribute of the form:

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

where <URI> is a URI, as above, <value> is the local identifier (ID) of this extension, and is an integer in the range 1-14 inclusive (0 and 15 are reserved, as noted above), and <direction> is one of "sendonly", "recvonly", "sendrecv", "inactive" (without the quotes).

The formal BNF syntax is presented in a later section 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) which is diagnostic that this style of header extensions is used, not the magic number indicated above.

6. Offer/Answer

The simple signaling described above may be enhanced in an offer/answer context, to permit:

o asymmetric behavior (extensions sent in only one direction);

o the offer of mutually-exclusive alternatives;

o 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", "inactive". A "sendonly" direction indicates an ability to send; a "recvonly" direction indicates a desire to receive; a "sendrecv" direction indicates both. An "inactive" direction indicates neither, but later re-negotiation may make an extension active.

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 may be modified 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 which has no desire to receive the extension or does not understand the extension SHOULD remove it from the SDP answer.

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 which has no desire to, or is unable to, send the extension SHOULD remove it from the SDP answer.

Identifiers in the range 1-14 inclusive in an offer or answer must not be used more than once per media section (including the session level section). A session update MAY change the direction qualifiers of extensions under use. A session update MAY add or remove extension(s). Identifiers values in the range 1-14 MUST NOT be altered (remapped).

Note that, under this rule, the same 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 re-numbering is not permitted either).

If a party wishes to offer mutually exclusive alternatives, then multiple extensions with the same identifier in the (unusable) 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 range 1-14, for that extension to be usable.

Similarly, if more than 14 extensions are offered, 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 range 1-14.

It is always allowed to place the offered identifier value "as is" in the SDP answer (for example, due to lack of a free identifier value in the range 1-14). Extensions with an identifier outside the range 1-14 cannot, of course, be used. If required, the offerer or answerer can update the session to make space for such an extension.

Rationale: the range 4096-4351 for these negotiation identifiers is deliberately restricted to allow expansion of the range of valid identifiers in future (e.g. by using a full byte for an ID).

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 range 1-14 in SDP originated by the sender can be sent.

Example (port numbers, RTP profiles, payload IDs and 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. They are not interested in transmission offsets on audio, and do not understand the URI-obscure extension. They therefore move the extensions from session level to media level, and adjust 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
```

7. BNF Syntax

The syntax element 'URI-reference' is as defined in [RFC3986], except that only absolute URIs are permitted here. The syntax element 'extmap' is an attribute as defined in [RFC4566]. Extensionattributes are not defined here, but by the specification that defines a specific extension name; there may be several.

```
digit = "0"/"1"/"2"/"3"/"4"/"5"/"6"/"7"/"8"/"9"

integer = 1*digit

space = " "

extensionname = URI-reference

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

mapentry = "extmap:" integer ["/" direction]

mapattrs = [space extensionattributes]

extmap = mapentry space extensionname mapattrs
```

8. Security Considerations

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

Care should be taken when defining extensions. Clearly, they should be solely informative, but even when the information is extracted, should not cause security concerns.

Header extensions have the same security coverage as the RTP header itself. When SRTP [RFC3711] is used to protect RTP sessions, the RTP payload may be both encrypted and integrity protected, while the RTP header is either unprotected or integrity protected. Therefore, it is inappropriate to place information in header extensions which cause security problems if disclosed, unless the entire RTP packet is protected by a lower-layer security protocol providing both confidentiality and integrity capability.

9. IANA Considerations

9.1. New space for IANA to manage

The rtp-hdrext namespace under urn:ietf:params: needs to be created for management, referenced to RFCxxxx. Additions in this namespace shall be made on the basis of "Specification Required".

Note: Names drawn from other spaces than the IETF are managed outside both the IETF and IANA, and the handling of registration and documentation is the responsibility of the owner of the internet domain name as of the date specified in the registration; no IANA action is required for these names.

The IANA will also maintain a server available via at least HTTP and FTP that contains all of the registered elements in some publicly accessible space in the same way that all of the IANA's registered elements are available via http://www.iana.org/assignments/. The suggested path is
http://www.iana.org/assignments/RTP-header-extensions/.

Here is the formal declaration required by the IETF URN Sub-namespace specification [RFC3553].

- o Registry name: urn:ietf:params:rtp-hdrext: (RTP header extensions defined by the IETF)

- o Specification: RFCxxxx and RFCs updating RFCxxxx.

- o Repository: see above.

- o Index value: -- The index value is an absolute URI, chosen for uniqueness within the parameter space.

9.2. Registration of the SDP extmap attribute

This section contains the information required by [RFC4566] for an SDP attribute.

- o contact name, email address and telephone number: D. Singer, singer@apple.com, +1 408-974-3162

- o attribute-name (as it will appear in SDP): extmap

- o long-form attribute name in English: generic header extension map definition

- o type of attribute (session level, media level, or both): both

- o whether the attribute value is subject to the charset attribute: not subject to the charset attribute

- o a one paragraph explanation of the purpose of the attribute: This attribute defines the mapping from the extension numbers used in packet headers into extension names as documented in specifications and appropriately registered.

- o a specification of appropriate attribute values for this attribute: see RFCxxxx.

10. RFC Editor Considerations

RFCxxxx in the IANA considerations needs to be replaced with the RFC number (two places).

11. Acknowledgments

Both Brian Link and John Lazzaro provided helpful comments on an initial draft. 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 final review.

12. Normative References

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

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, STD 0064, July 2003.

[RFC3553] Mealling, T., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", RFC 3553, June 2003.

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

[RFC3986] Berners-Lee, MT., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005.

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

Authors' Addresses

David Singer
Apple Computer Inc.
1 Infinite Loop
Cupertino, CA 95014
US

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

Harikishan Desineni
Qualcomm
5775 Morehouse Drive
San Diego, CA 92126
USA

Phone: +1 858 845 8996
Email: hd@qualcomm.com
URI: http://www.qualcomm.com

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).