

Network Working Group
Request for Comments: 5168
Category: Informational

O. Levin
Microsoft Corporation
R. Even
Polycom
P. Hagendorf
RADVISION
March 2008

XML Schema for Media Control

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This document defines an Extensible Markup Language (XML) Schema for video fast update in a tightly controlled environment, developed by Microsoft, Polycom, Radvision and used by multiple vendors. This document describes a method that has been deployed in Session Initiation Protocol (SIP) based systems over the last three years and is being used across real-time interactive applications from different vendors in an interoperable manner. New implementations are discouraged from using the method described except for backward compatibility purposes. New implementations are required to use the new Full Intra Request command in the RTP Control Protocol (RTCP) channel.

Table of Contents

| | |
|--|---|
| 1. Introduction | 2 |
| 2. Conventions | 2 |
| 3. Background | 3 |
| 4. The Video Control Commands | 3 |
| 5. The Schema Definition | 4 |
| 6. Error Handling | 5 |
| 7. Examples | 5 |
| 7.1. The Fast Update Command for the Full Picture | 5 |
| 7.2. Reporting an Error | 5 |
| 8. Transport | 6 |
| 9. IANA Considerations | 6 |
| 9.1. Application/media_control+xml Media Type Registration | 6 |
| 10. Security Considerations | 7 |
| 11. References | 8 |
| 11.1. Normative References | 8 |
| 11.2. Informative References | 8 |

1. Introduction

This document defines an Extensible Markup Language (XML) Schema for video fast update request in a tightly controlled environment, developed by Microsoft, Polycom, Radvision and used by multiple vendors. Implementation of this schema for interactive video applications in Session Initiation Protocol (SIP) [5] environments was designed in order to improve user experience. This mechanism is being used by both end user video conferencing terminals and conferencing servers in shipping products. This document describes the current method, but new implementations are discouraged from using this method, except for backward compatibility with legacy systems. Shipping products and new products SHOULD use the Full Intra Request, described in [7].

Sending video fast update using the SIP signaling path, as described in this document, is inferior to using the RTP Control Protocol (RTCP) feedback method [7], since the command flows through all the proxies in the signaling path adding delay to the messages and causing unnecessary overload to the proxies. RTCP messages flow end-to-end and not through the signaling proxies. The RTCP feedback document [7] also adds other required control functions, such as the flow control command, which is missing from this document.

2. Conventions

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 RFC 2119 [2].

3. Background

SIP typically uses the Real-time Transport Protocol (RTP) [6] for the transferring of real-time media.

RTP is augmented by a control protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks. The RTCP feedback mechanism [8] has been introduced in order to improve basic RTCP feedback time in case of loss conditions across different coding schemes. This technique addresses signaling of loss conditions and the recommended recovery steps.

Just recently, an extension to the feedback mechanism has been proposed [7] to express control operations on media streams as a result of application logic rather than a result of loss conditions. Note that in the decomposed systems, the implementation of the new mechanism will require proprietary communications between the applications/call control components and the media components.

This document describes a technology that has been deployed in SIP-based systems over the last three years and is being used across real-time interactive applications from different vendors in an interoperable manner. This memo documents this technology for the purpose of describing current practice and new implementation MUST use the RTCP Full Intra Request command specified in the RTCP-based codec control messages document[7].

4. The Video Control Commands

Output of a video codec is a frame. The frame can carry complete information about a picture or about a picture segment. These frames are known as "Intra" frames. In order to save bandwidth, other frames can carry only changes relative to previously sent frames. Frames carrying relative information are known as "Inter" frames.

Based on application logic (such as need to present a new video source), the application needs to have an ability to explicitly request from a remote encoder the complete information about a "full" picture.

An "Intra" frame may be of large size. In order to prevent causing network congestion, the current media capacity and network conditions MUST be validated before sending an "Intra" frame when receiving a fast update command, defined in this document.

In order to meet the presented requirements, a video primitive is defined by this document.

The following command is sent to the remote encoder:

- o Video Picture Fast Update

5. The Schema Definition

```
<?xml version="1.0" encoding="utf-8" ?>

<xs:schema id="TightMediaControl"
  elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="media_control">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="vc_primitive"
          type="vc_primitive"
          minOccurs="0"
          maxOccurs="unbounded" />
        <xs:element name="general_error"
          type="xs:string"
          minOccurs="0"
          maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Video control primitive. -->

  <xs:complexType name="vc_primitive">
    <xs:sequence>
      <xs:element name="to_encoder" type="to_encoder" />
      <xs:element name="stream_id"
        type="xs:string"
        minOccurs="0"
        maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <!-- Encoder Command:
    Picture Fast Update
  -->

  <xs:complexType name="to_encoder">
    <xs:choice>
      <xs:element name="picture_fast_update"/>
    </xs:choice>
  </xs:complexType>
```

</xs:schema>

6. Error Handling

Currently, only a single general error primitive is defined. It MAY be used for indicating errors in free-text format. The general error primitive MAY report problems regarding XML document parsing, inadequate level of media control support, inability to perform the requested action, etc.

The general error primitive MUST NOT be used for the indication of errors other than those related to media control parsing or to resultant execution. The general error primitive MUST NOT be sent back as a result of getting an error primitive.

When receiving the general error response, the user agent client (UAC) that sent the request SHOULD NOT send further fast update requests in the current dialog.

According to RFC 2976 [3], the only allowable final response to a SIP INFO containing a message body is a 200 OK. If SIP INFO is used to carry the request, the error message should be carried in a separate INFO request.

7. Examples

7.1. The Fast Update Command for the Full Picture

In the following example, the full picture "Fast Update" command is issued towards the remote video decoder(s).

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<media_control>
```

```
  <vc_primitive>
```

```
    <to_encoder>
```

```
      <picture_fast_update/>
```

```
    </to_encoder>
```

```
  </vc_primitive>
```

```
</media_control>
```

7.2. Reporting an Error

If an error occurs during the parsing of the XML document, the following XML document would be sent back to the originator of the original Media Control document.

```
<?xml version="1.0" encoding="utf-8" ?>

<media_control>

  <general_error>
    Parsing error: The original XML segment is:...
  </general_error>

</media_control>
```

8. Transport

The defined XML document is conveyed using the SIP INFO method [3] with the "Content-Type" set to "application/media_control+xml". This approach benefits from the SIP built-in reliability.

9. IANA Considerations

This document registers a new media type.

9.1. Application/media_control+xml Media Type Registration

Type name: application
Subtype name: media_control+xml
Required parameters: None
Optional parameters: charset

Indicates the character encoding of enclosed XML.

Encoding considerations: 8bit
Uses XML, which can employ 8-bit characters, depending on the character encoding used. See RFC 3023 [4], Section 3.2.

Security considerations: Security considerations specific to uses of this type are discussed in RFC 5168. RFC 1874 [1] and RFC 3023 [4] discuss security issues common to all uses of XML.

Interoperability considerations: None.

Published specification: RFC 5168

Applications that use this media type: This media type is used to convey information regarding media control commands and responses between SIP endpoints particularly for allowing a Video Fast Update intra-frame request.

Additional information:

Magic Number(s): None.
File Extension(s): None.
Macintosh File Type Code(s): None.

Person and email address to contact for further information:

Name: Roni Even
E-Mail: even.roni@gmail.com

Intended usage: LIMITED USE

Restrictions on usage: None.

Author: Roni Even. <even.roni@gmail.com>

Change Controller: Roni Even. <even.roni@gmail.com>

10. Security Considerations

The video control command transported using the method described in the document may cause the sender of the video data to send more data within the allowed bandwidth, as described in Section 4.

This document defines one control message; changing the content of the message will cause the video sender to ignore the request and send an error response. This may prevent the display of a video stream. The control message itself does not carry any sensitive information.

An eavesdropper may inject messages or change them, which may lead to either more data on the network or loss of video image. Using data integrity validation will prevent adding or changing of messages.

If the video media is sent over a secure transport, it is recommended to secure the signaling using TLS as explained in [5]. In most cases, securing the media will require a secure signaling path.

The security considerations of [3] and [5] apply here.

11. References

11.1. Normative References

- [1] Levinson, E., "SGML Media Types", RFC 1874, December 1995.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [3] Donovan, S., "The SIP INFO Method", RFC 2976, October 2000.
- [4] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.
- [5] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [6] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [7] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008.

11.2. Informative References

- [8] 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, July 2006.

Authors' Addresses

Orit Levin
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
USA

EMail: oritl@microsoft.com

Roni Even
Polycom
94 Derech Em Hamoshavot
Petach Tikva, 49130
Israel

EMail: roni.even@polycom.co.il

Pierre Hagendorf
RADVISION
24, Raul Wallenberg St.
Tel-Aviv, 69719
Israel

EMail: pierre@radvision.com

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

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.

