Internet Engineering Task Force
Internet Draft

Nomura/Schulzrinne
Fujitsu/Columbia U.

draft-nomura-cdi-mmusic-mupdate-00.txt
June 24, 2002
Expires: November 2002

SIP Event Notification for Metadata Update Protocol

Abstract

This document specifies an update notification protocol for metadata using SIP event notification. This protocol improves metadata coherency and efficiency between content providers and receivers by up-to-date notifications when metadata changes. This protocol can also be applied to several systems such as a streaming video distribution system and a web content ditsribution system.

1 Introduction

Metadata can describe content features and its structures of multimedia content and web pages. Metadata is a principal element for content providers or receivers to manage and distribute multimedia and web content [1]. It allows users to initiate streaming media sessions, schedule delivery of downloadable or multicast content or

listen to live multicast sessions [2].

Since content and its structure described by metadata changes as time elapses, a content receiver needs to be notified of changes in order to avoid stale state of metadata and content.

This document defines an update notification protocol for metadata using SIP Event Notification framework [3]. It satisfies the requirements for multimedia and web content management such as an update notification, session keep-alive, status confirmation and authentication of subscriber. In addition, using SIP framework, extensiblity and flexiblity is provided.

This document also gives a guideline for metadata description and update information formats used in the protocol. The guideline includes candidate description formats and delta description methods.

2 Terminology

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [4].

Notifier: A notifier is a user agent which generates NOTIFY requests and receives SUBSCRIBE requests.

Subscriber: A subscriber is a user agent which receives NOTIFY requests from notifiers and generates SUBSCRIBE requests.

Session: An identical metadata state between a notifier and a subscriber.

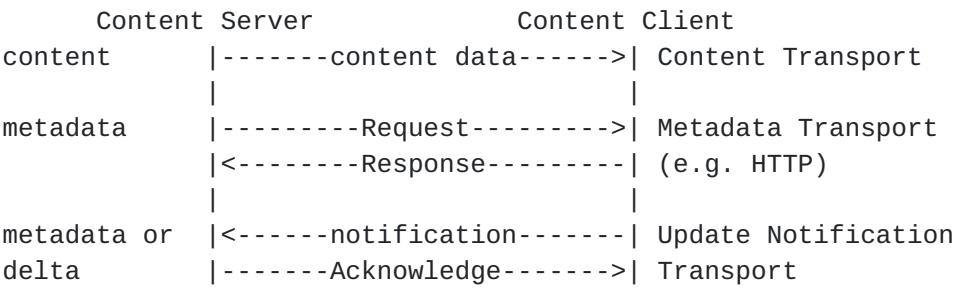
Delta: Differences between the latest metadata state and the previous state of metadata.

3 Overview of the content distribution system

Elements of the System: This protocol is used in a content distribution system consists of at least two hosts, one is a content server and the other is a content client.

Target Content: All content such as a multimedia streaming data, multimedia file, and Web content consists web pages.

Three Transport Model: A content distribution system has three transports, (1) content transport, (2) metadata transport and (3) update notification (UN) transport.



This protocol is used by Update Notification Transport but it can be used in Metadata Transport if required.

1. Content Transport

   This transport is used to transmit content in the system. Any existing transport mechanism can be applied. The transport is specified in metadata transmitted by Metadata Transport. This document does not require particular content transport.

2. Metadata Transport

   This transport is used to transmit metadata or delta information in the system. Any existing transport mechanism can be applied but the simplest transport is HTTP. The transport is specified and selected in UN transmitted by UN transport. This document assumes that Metadata transport does not provide Update Notification Transport.

3. Update Notification Transport

   This transport is used to transmit update notifications from a content server to a content client. When Request/Response protocol like HTTP is used in Metadata Transport and the system requires up-to-date content distribution, this transport is necessary.

   The problem of using HTTP for the metadata transport is an inefficient refresh mechanism by polling. The freshness of metadata has to be improved to send HTTP requests frequently but the frequent requests increase the load on the host receiving the unnecessary requests. The detail of this problem has been introduced in RUP requirements [5].

   The update notification like this protocol and RUP activities aims to solve this problem.

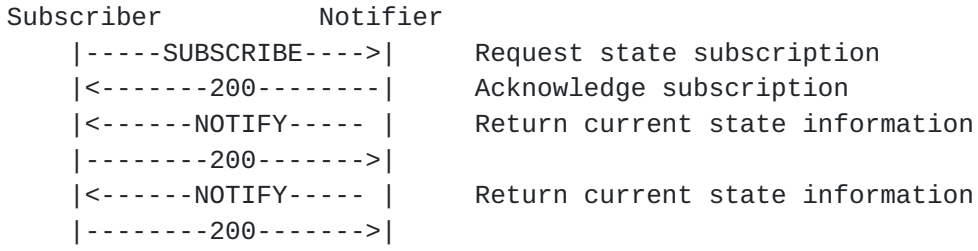
4 Overview of SIP Event Notification

4.1 Basic Protocol Operations

SIP [6] is a text-based request/response protocol. A SIP request consists of a request-line, header field and message body like HTTP. A SIP response also consists of a status-line, header field and body.

SIP Event has the same feature as SIP but SIP Event needs only two messages, SUBSCRIBE and NOTIFY, to notify an event to subscribers. Since the protocol intends to do this, it does not require SIP messages to establish a session.

The following diagram shows basic protocol operations in SIP Event. Protocol details are shown in section 5.



There are several reasons to use SIP Event for metadata update notification.

4.2 Features of SIP Event Notification

1. Simple: SIP Event consists of just two messages to satisfy RUP requirement and Program guide requirement.

2. Standard: Since SIP Event is used by many applications in IMPP [7] and call services, there are a lot of implementations based on it as a standard specification. SIP Event is carefully designed to make it secure.

3. Extensible: SIP Event can take advantage of SIP functions and services such as a proxy and so on.

4. Independent of transport protocols: SIP runs on top of several different transport protocols. Therefore, it is possible to use various transports such as UDP, TCP and multicast.

5 Protocol Details

5.1 Initialize

First of all, a subscriber of metadata MUST send a SUBSCRIBE request to a notifier in order to prepare to receive a subsequent update notification from the notifier.

The SUBSCRIBE request includes identity information in its headers defined by SIP Events framework. As defined in SIP, To, From, Call-ID, Event and Contact header can be used to identify a session between a subscriber and notifier. Each parameter in the headers depends on the implementations.

In this phase, the SUBSCRIBE request MAY contain a body indicating what metadata status will be subscribed.

The name of this package is "metadataupdate". This header also appears in NOTIFY requests.

If the SUBSCRIBE request is not related to existing sessions and the notifier can authenticate the request successfully, the notifier sends a 200 (OK) response to the subscriber. If the notifier can't authenticate or accept the request, the notifier sends a 4xx response and does not send a NOTIFY request.

A notifier MUST authenticate all subscription requests. This authentication can be done using any of the mechanisms defined in SIP and SIP Events.

After sending the SUBSCRIBE response, the notifier sends a NOTIFY request immediately to the same subscriber. The request contains an URI indicating a metadata location or metadata. Metadata indicated by URI or contained in the body MUST describe the latest full state when the NOTIFY request is sent. "Content-Type:" header in the NOTIFY request MUST indicate a data type of the body.

The body in NOTIFY request MUST contains a version number and a timestamp. The version number increases by exactly one for each NOTIFY message as defined in SIP Event. The time stamp indicates latest modified time of metadata status.

When the subscriber receives the request, the subscriber tries to obtain metadata specified in the body. If the subscriber gets it successfully, it sends 200 (OK) response to the notifier. If not, the subscriber sends 4xx response.

5.2 Update Notification

When metadata changes and it affects an existing subscription, notifier sends a NOTIFY request on a session related to the metadata status. The request body SHOULD contain metadata delta location or metadata delta. The format of metadata delta is discussed in section 6.

When the subscriber receives the request, the subscriber tries to obtain delta information specified in the body. If the subscriber successfully gets it and updates metadata, it sends 200 (OK) response to the notifier. If not, the subscriber sends 4xx response.

5.3 Session Keep Alive

At any time before a subscription expires, the subscriber may send a SUBSCRIBE request to the notifier. The notifier sends a SUBSCRIBE response and a NOTIFY request with delta information to communicate up-to-date metadata status.

If the subscriber receives the NOTIFY message which includes the same timestamp as the previous NOTIFY message, the subscriber does not have to obtain delta information. In this case, a notifier sends it just for a confirmation of the metadata status as an immediate response for a SUBSCRIBE request.

If the timestamp is updated, the subscriber MUST obtain new delta in the same way as receiving an update notification.

This mechanism can be applicable not only to refresh the timer but also to confirm the current metadata status.

5.4 Polling metadata status

If a subscriber does not want to receive an update notification, a subscriber may poll metadata status to send a SUBSCRIBE with an "Expires" of 0.

5.5 Confirming the status of Subscribers

If a notifier needs to confirm the current status of a subscriber which subscribes metadata status, it may send NOTIFY request including a body which indicates a current delta information and wait a NOTIFY response from the subscriber. When the notifier receives the response with 200 (OK), the notifier confirms that the subscriber's status is up-to-date. If not, the notifier can decide that the subscriber has stale metadata.

5.6 Timer Expiration

Expiration time depends on the system. If the system requires short duration to guarantee metadata coherency, it should be a small value. In some system, a connection between a subscriber and notifier is not persistent but sporadic. In this case, subscriber may require the long expiration time such as 1 day or 1 week.

If a subscriber receives NOTIFY request on the session which already has been terminated because the timer has expired, the subscriber SHOULD try to subscribe it again.

5.7 Invalid update notification

If a subscriber receives invalid NOTIFY message such as a discontinuous version number or CSeq, the subscriber SHOULD close the session and restart the session from Initialize step. As a result, the subscriber can obtain latest full metadata states.

6 Descriptions of Metadata and Update information

6.1 Candidate Metadata Description

MPEG-7 [8] can describe metadata of content by using XML. It defines XML schema representing general multimedia content and structure of content group.

WCIP [9] provides a description format of a data structure for web content group in order to achieve cache coherency between an origin web notifier and a surrogate.

These two approaches are very similar in terms of describing a structure of content group.

If metadata appears in a body of SIP Event, a data type MUST be specified in Content-Type in header field.

6.2 Metadata based Invalidation

Metadata can describe availability information for content. If a subscriber recognizes that the availability has been invalid, the subscriber can invalidate content spontaneously without being notified by a NOTIFY request from the notifier.

Availability information in metadata can contribute to reduce NOTIFY messages indicating the invalidation operation. In this case, update notifications are just used to substitute invalid metadata for updated metadata. Removing content due to updating metadata may be required on a system provides a content coherency between two hosts.

6.3 Delta Description

Separate description between metadata and delta is RECOMMENDED.

Metadata may describe delta information between two metadata. However, describing delta information using metadata makes metadata description complex and less extensible.

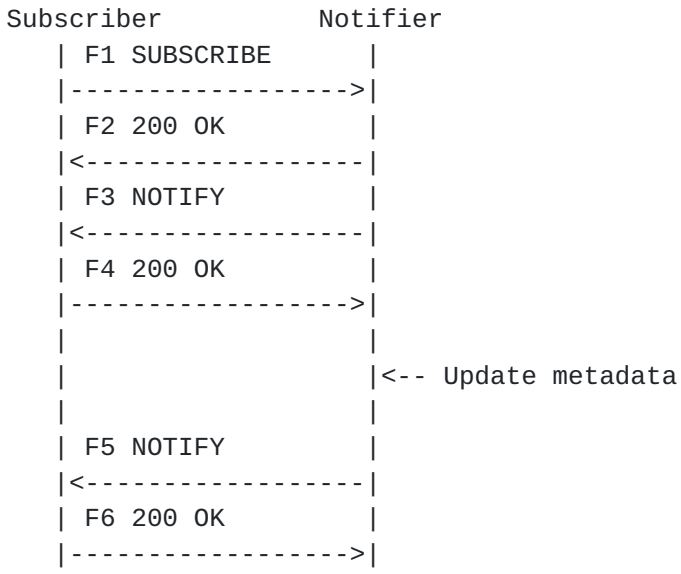
Let's assume that one text notation in metadata should be substituted another text and metadata described by XML. In terms of a general tree structure such a XML, to identify child node information requires all of parent node information to which child node belongs because child node itself does not have identity information.

In this case, delta information must contain all of parent node information in a document from a root node to a leaf node in order to describe the only one text to be substituted.

Since metadata schema should provide identity information for every child nodes in order to describe delta information, it results that metadata schema becomes complex and difficult to extensible.

To avoid the problem, it is useful to keep metadata from delta information and introduce differencing algorithm and description such as the UNIX "diff" command described in delta encoding in HTTP [10]. Even though additional "diff" mechanism is required, this function is enough simple and make metadata schema simple and extensible.

7 Example Message Flow



F1 SUBSCRIBE subscriber.com->notifier.com

```
SUBSCRIBE sip:metadata@notifier.com SIP/2.0
Via: SIP/2.0/TCP host1.subscriber.com;branch=z9hG4bKnashds7
To: <sip:metadata@notifier.com>
From: <sip:metadata@subscriber.com>;tag=xfg9
Call-ID: 7070@host1.subscriber.com
CSeq: 15024 SUBSCRIBE
Max-Forwards: 70
Event: metadataupdate
Contact: <sip:metadata@host.subscriber.com>
Expires: 300
Content-Length: 0
```

F2 200 OK notifier.com->subscriber.com

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP host1.subscriber.com;branch=z9hG4bKnashds7
  ;received=192.0.2.2
To: <sip:metadata@notifier.com>;tag=ffd2
From: <sip:metadata@subscriber.com>;tag=xfg9
Call-ID: 7070@host1.subscriber.com
CSeq: 15024 SUBSCRIBE
Event: metadataupdate
Expires: 300
Contact: <sip:metadata@host2.notifier.com>
Content-Length: 0
```

F3 NOTIFY notifier.com-> subscriber.com

```
NOTIFY metadata@host.subscriber.com SIP/2.0
Via: SIP/2.0/TCP host2.notifier.com;branch=z9hG4bKna998sk
From: <sip:metadata@notifier.com>;tag=ffd2
To: <sip:metadata@subscriber.com>;tag=xfg9
Call-ID: 7070@host1.subscriber.com
Event: metadataupdate
Subscription-State: active;expires=599
Max-Forwards: 70
CSeq: 3487 NOTIFY
Content-Type: text/xml
Content-Length: ...

Version: 1
Last-Modified: Mon, 24 Jun 2002 08:03:55 GMT
Metadata-URI: http://metadata.notifier.com/program1/02080355.xml
```

F4 200 OK subscriber.com-> notifier.com

```
      SIP/2.0 200 OK
      Via: SIP/2.0/TCP host2.notifier.com;branch=z9hG4bKna998sk
        ;received=192.0.2.2
      From: <sip:metadata@subscriber.com>;tag=ffd2
      To: <sip:metadata@notifier.com>;tag=xfg9
      Call-ID: 7070@host1.subscriber.com
      CSeq: 3487 NOTIFY
```

F5 NOTIFY notifier.com -> subscriber.com

```
      NOTIFY sip:metadata@host.subscriber.com SIP/2.0
      Via: SIP/2.0/TCP host2.notifier.com;branch=z9hG4bKna998sl
      From: <sip: metadata@notifier.com>;tag=ffd2
      To: <sip:metadata@subscriber.com>;tag=xfg9
      Call-ID: 7070@host1.subscriber.com
      CSeq: 3488 NOTIFY
      Event: metadataupdate
      Subscription-State: active;expires=543
      Max-Forwards: 70
      Content-Type: text/plain
      Content-Length: ...

      Version: 2
      Last-Modified: Mon, 24 Jun 2002 09:19:31 GMT
      Delta-URI: http://delta.notifier.com/program1/02091931.txt
```

F6 200 OK subscriber.com-> notifier.com

```
      SIP/2.0 200 OK
      Via: SIP/2.0/UDP notifier.example.com;branch=z9hG4bKna998sl
       ;received=192.0.2.2
      From: <sip:metadata@subscriber.com>;tag=ffd2
      To: <sip:metadata@notifier.com>;tag=xfg9
      Call-ID: 7070@host1.subscriber.com
      CSeq: 3488 NOTIFY
      Content-Length: 0
```

8 Security Considerations

TBD

9 Bibliography

[1] L. Amini et al., "Distribution requirements for content internetworking," Internet Draft, Internet Engineering Task Force, Feb. 2002. Work in progress.

[2] Y. Nomura and H. Schulzrinne, "Protocol requirements for internet program guides," Internet Draft, Internet Engineering Task Force, Feb. 2002. Work in progress.

[3] A. Roach, "SIP-specific event notification," Internet Draft, Internet Engineering Task Force, Mar. 2002. Work in progress.

[4] S. Bradner, "Key words for use in RFCs to indicate requirement levels," RFC 2119, Internet Engineering Task Force, Mar. 1997.

[5] I. Cooper, D. Li, M. Dahlin, and M. Hamilton, "Requirements for a resource update protocol," Internet Draft, Internet Engineering Task Force, Mar. 2002. Work in progress.

[6] J. Rosenberg, H. Schulzrinne, et al., "SIP: Session initiation protocol," Internet Draft, Internet Engineering Task Force, Feb. 2002. Work in progress.

[7] J. Rosenberg, "Session initiation protocol (SIP) extensions for presence," Internet Draft, Internet Engineering Task Force, May 2002. Work in progress.

[8] ISO (International Organization for Standardization), "Overview of the MPEG-7 standard," ISO Standard ISO/IEC JTC1/SC29/WG11 N4509, International Organization for Standardization, Geneva, Switzerland, Dec. 2001.

[9] D. Li, P. Cao, and M. Dahlin, "WCIP: Web cache invalidation protocol," Internet Draft, Internet Engineering Task Force, Mar. 2001. Work in progress.

[10] J. Mogul, B. Krishnamurthy, F. Douglis, A. Feldmann, Y. Goland, A. van Hoff, and D. Hellerstein, "Delta encoding in HTTP," RFC 3229, Internet Engineering Task Force, Jan. 2002.

10 Author's Addresses

Yuji Nomura
Fujitsu Laboratories Ltd.
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki 211-8588
Japan
Email: nom@flab.fujitsu.co.jp

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027

USA
Email: schulzrinne@cs.columbia.edu