Network Working Group
Request for Comments: 3311
Category: Standards Track

J. Rosenberg
dynamicsoft

September 2002

The Session Initiation Protocol (SIP) UPDATE Method

Status of this Memo

Copyright Notice

Abstract

This specification defines the new UPDATE method for the Session Initiation Protocol (SIP). UPDATE allows a client to update parameters of a session (such as the set of media streams and their codecs) but has no impact on the state of a dialog. In that sense, it is like a re-INVITE, but unlike re-INVITE, it can be sent before the initial INVITE has been completed. This makes it very useful for updating session parameters within early dialogs.

Table of Contents

1 Introduction

The Session Initiation Protocol (SIP) [1] defines the INVITE method for the initiation and modification of sessions. However, this method actually affects two important pieces of state. It impacts the session (the media streams SIP sets up) and also the dialog (the state that SIP itself defines). While this is reasonable in many cases, there are important scenarios in which this coupling causes complications.

The primary difficulty is when aspects of the session need to be modified before the initial INVITE has been answered. An example of this situation is "early media", a condition where the session is established, for the purpose of conveying progress of the call, but before the INVITE itself is accepted. It is important that either caller or callee be able to modify the characteristics of that session (putting the early media on hold, for example), before the call is answered. However, a re-INVITE cannot be used for this purpose, because the re-INVITE has an impact on the state of the dialog, in addition to the session.

As a result, a solution is needed that allows the caller or callee to provide updated session information before a final response to the initial INVITE request is generated. The UPDATE method, defined here, fulfills that need. It can be sent by a UA within a dialog (early or confirmed) to update session parameters without impacting the dialog state itself.

2 Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [2] and indicate requirement levels for compliant SIP implementations.

3 Overview of Operation

Operation of this extension is straightforward. The caller begins with an INVITE transaction, which proceeds normally. Once a dialog is established, either early or confirmed, the caller can generate an UPDATE method that contains an SDP offer [3] for the purposes of updating the session. The response to the UPDATE method contains the answer. Similarly, once a dialog is established, the callee can send an UPDATE with an offer, and the caller places its answer in the 2xx to the UPDATE. The Allow header field is used to indicate support for the UPDATE method. There are additional constraints on when UPDATE can be used, based on the restrictions of the offer/answer model.

4 Determining Support for this Extension

The initiation of a session operates as specified in RFC 3261 [1]. However, a UAC compliant to this specification SHOULD also include an Allow header field in the INVITE request, listing the method UPDATE, to indicate its ability to receive an UPDATE request.

When a UAS compliant to this specification receives an INVITE request for a new dialog, and generates a reliable provisional response containing SDP, that response SHOULD contain an Allow header field that lists the UPDATE method. This informs the caller that the callee is capable of receiving an UPDATE request at any time. An unreliable provisional response MAY contain an Allow header field listing the UPDATE method, and a 2xx response SHOULD contain an Allow header field listing the UPDATE method.

Responses are processed normally as per RFC 3261 [1], and in the case of reliable provisional responses, according to [4]. It is important to note that a reliable provisional response will always create an early dialog at the UAC. Creation of this dialog is necessary in order to receive UPDATE requests from the callee.

If the response contains an Allow header field containing the value "UPDATE", the UAC knows that the callee supports UPDATE, and the UAC is allowed to follow the procedures of Section 5.1.

5 UPDATE Handling

5.1 Sending an UPDATE

The UPDATE request is constructed as would any other request within an existing dialog, as described in Section 12.2.1 of RFC 3261. It MAY be sent for both early and confirmed dialogs, and MAY be sent by either caller or callee. Although UPDATE can be used on confirmed dialogs, it is RECOMMENDED that a re-INVITE be used instead. This is because an UPDATE needs to be answered immediately, ruling out the possibility of user approval. Such approval will frequently be needed, and is possible with a re-INVITE.

The UAC MAY add optional headers for the UPDATE request, as defined in Tables 1 and 2.

UPDATE is a target refresh request. As specified in RFC 3261 [1], this means that it can update the remote target of a dialog. If a UA uses an UPDATE request or response to modify the remote target while an INVITE transaction is in progress, and it is a UAS for that INVITE transaction, it MUST place the same value into the Contact header field of the 2xx to the INVITE that it placed into the UPDATE request or response.

The rules for inclusion of offers and answers in SIP messages as defined in Section 13.2.1 of RFC 3261 still apply. These rules exist to guarantee a consistent view of the session state. This means that, for the caller:

- If the UPDATE is being sent before completion of the initial INVITE transaction, and the initial INVITE contained an offer, the UPDATE can contain an offer if the callee generated an answer in a reliable provisional response, and the caller has received answers to any other offers it sent in either PRACK or UPDATE, and has generated answers for any offers it received in an UPDATE from the callee.

- If the UPDATE is being sent before completion of the initial INVITE transaction, and the initial INVITE did not contain an offer, the UPDATE can contain an offer if the callee generated an offer in a reliable provisional response, and the UAC generated an answer in the corresponding PRACK. Of course, it can’t send an UPDATE if it has not received answers to any other offers it sent in either PRACK or UPDATE, or has not generated answers for any other offers it received in an UPDATE from the callee.

- If the UPDATE is being sent after the completion of the initial INVITE transaction, it cannot contain an offer if the caller has generated or received offers in a re-INVITE or UPDATE which have not been answered.

and for the callee:

- If the UPDATE is being sent before the completion of the INVITE transaction, and the initial INVITE contained an offer, the UPDATE cannot be sent with an offer unless the callee has generated an answer in a reliable provisional response, has received a PRACK for that reliable provisional response, has not received any requests (PRACK or UPDATE) with offers that it has not answered, and has not sent any UPDATE requests containing offers that have not been answered.

- If the UPDATE is being sent before completion of the INVITE transaction, and the initial INVITE did not contain an offer, the UPDATE cannot be sent with an offer unless the callee has sent an offer in a reliable provisional response, received an answer in a PRACK, and has not received any UPDATE requests with offers that it has not answered, and has not sent any UPDATE requests containing offers that have not been answered.

- If the UPDATE is being sent after the completion of the initial INVITE transaction, it cannot be sent with an offer if the callee has generated or received offers in a re-INVITE or UPDATE which have not been answered.

5.2 Receiving an UPDATE

The UPDATE is processed as any other mid-dialog target refresh request, as described in Section 12.2.2 of RFC 3261 [1]. If the request is generally acceptable, processing continues as described below. This processing is nearly identical to that of Section 14.2 of RFC 3261 [1], but generalized for the case of UPDATE.

A UAS that receives an UPDATE before it has generated a final response to a previous UPDATE on the same dialog MUST return a 500 response to the new UPDATE, and MUST include a Retry-After header field with a randomly chosen value between 0 and 10 seconds.

If an UPDATE is received that contains an offer, and the UAS has generated an offer (in an UPDATE, PRACK or INVITE) to which it has not yet received an answer, the UAS MUST reject the UPDATE with a 491 response. Similarly, if an UPDATE is received that contains an offer, and the UAS has received an offer (in an UPDATE, PRACK, or INVITE) to which it has not yet generated an answer, the UAS MUST

reject the UPDATE with a 500 response, and MUST include a Retry-After header field with a randomly chosen value between 0 and 10 seconds.

If a UA receives an UPDATE for an existing dialog, it MUST check any version identifiers in the session description or, if there are no version identifiers, the content of the session description to see if it has changed. If the session description has changed, the UAS MUST adjust the session parameters accordingly and generate an answer in the 2xx response. However, unlike a re-INVITE, the UPDATE MUST be responded to promptly, and therefore the user cannot generally be prompted to approve the session changes. If the UAS cannot change the session parameters without prompting the user, it SHOULD reject the request with a 504 response. If the new session description is not acceptable, the UAS can reject it by returning a 488 (Not Acceptable Here) response for the UPDATE. This response SHOULD include a Warning header field.

5.3 Processing the UPDATE Response

Processing of the UPDATE response at the UAC follows the rules in Section 12.2.1.2 of RFC 3261 [1] for a target refresh request. Once that processing is complete, it continues as specified below. This processing is nearly identical to the processing of Section 14.1 of RFC 3261 [1], but generalized for UPDATE.

If a UA receives a non-2xx final response to a UPDATE, the session parameters MUST remain unchanged, as if no UPDATE had been issued. Note that, as stated in Section 12.2.1 of RFC 3261 [1], if the non-2xx final response is a 481 (Call/Transaction Does Not Exist), or a 408 (Request Timeout), or no response at all is received for the UPDATE (that is, a timeout is returned by the UPDATE client transaction), the UAC will terminate the dialog.

If a UAC receives a 491 response to a UPDATE, it SHOULD start a timer with a value T chosen as follows:

1. If the UAC is the owner of the Call-ID of the dialog ID (meaning it generated the value), T has a randomly chosen value between 2.1 and 4 seconds in units of 10 ms.

2. If the UAC is not the owner of the Call-ID of the dialog ID, T has a randomly chosen value between 0 and 2 seconds in units of 10 ms.

When the timer fires, the UAC SHOULD attempt the UPDATE once more, if it still desires for that session modification to take place. For example, if the call was already hung up with a BYE, the UPDATE would not take place.

6 Proxy Behavior

Proxy processing of the UPDATE request is identical to any other non-INVITE request.

7 Definition of the UPDATE method

The semantics of the UPDATE method are described in detail above. This extension adds another value to the Method BNF described in RFC 3261:

```
UPDATEm  =  %x55.50.44.41.54.45 ; UPDATE in caps
Method   =  INVITEm / ACKm / OPTIONSm / BYEm
            / CANCELm / REGISTERm / UPDATEm
            / extension-method
```

Table 1 extends Table 2 of RFC 3261 for the UPDATE method.

Table 2 updates Table 3 of RFC 3261 for the UPDATE method.

8 Example Call Flow

This section presents an example call flow using the UPDATE method. The flow is shown in Figure 1. The caller sends an initial INVITE (1) which contains an offer. The callee generates a 180 response (2) with an answer to that offer. With the completion of an offer/answer exchange, the session is established, although the dialog is still in the early state. The caller generates a PRACK (3) to acknowledge the 180, and the PRACK is answered with a 200 OK (4). The caller decides to update some aspect of the session - to put it on hold, for example. So, they generate an UPDATE request (5) with a new offer. This offer is answered in the 200 response to the UPDATE (6). Shortly thereafter, the callee decides to update some aspect of the session, so it generates an UPDATE request (7) with an offer, and the answer is sent in the 200 response (8). Finally, the callee answers the call, resulting in a 200 OK response to the INVITE (9), and then an ACK (10). Neither the 200 OK to the INVITE, nor the ACK, will contain SDP.

Header field	where	proxy	UPDATE
Accept	R		o
Accept	2xx		o
Accept	415		c
Accept-Encoding	R		o
Accept-Encoding	2xx		o
Accept-Encoding	415		c
Accept-Language	R		o
Accept-Language	2xx		o
Accept-Language	415		c
Alert-Info			-
Allow	R		o
Allow	2xx		o
Allow	r		o
Allow	405		m
Allow-Events	(1)		-
Authentication-Info	2xx		o
Authorization	R		o
Call-ID	c	r	m
Call-Info		ar	o
Contact	R		m
Contact	1xx		o
Contact	2xx		m
Contact	3xx	d	o
Contact	485		o
Content-Disposition			o
Content-Encoding			o
Content-Language			o
Content-Length		ar	t
Content-Type			*
CSeq	c	r	m
Date		a	o
Error-Info	300-699	a	o
Event	(1)		-
Expires			-
From	c	r	m
In-Reply-To			-
Max-Forwards	R	amr	m
Min-Expires			-
MIME-Version			o
Organization		ar	o

Table 1: Summary of header fields, A--O ; (1) defined in [5].

Header field	where	proxy	UPDATE
Priority			-
Proxy-Authenticate	407	ar	m
Proxy-Authenticate	401	ar	o
Proxy-Authorization	R	dr	o
Proxy-Require	R	ar	o
RAck	R		-
Record-Route	R	ar	o
Record-Route	2xx,18x	mr	o
Reply-To			-
Require		ar	c
Retry-After	404,413,480,486		o
	500,503		o
	600,603		o
Route	R	adr	c
RSeq	-		-
Server	r		o
Subject	-		-
Subscription-State	(1)		-
Supported	R		o
Supported	2xx		o
Timestamp			o
To	c	r	m
Unsupported	420		m
User-Agent			o
Via	R	amr	m
Via	rc	dr	m
Warning	r		o
WWW-Authenticate	401	ar	m
WWW-Authenticate	407	ar	o

Table 2: Summary of header fields, P--Z.

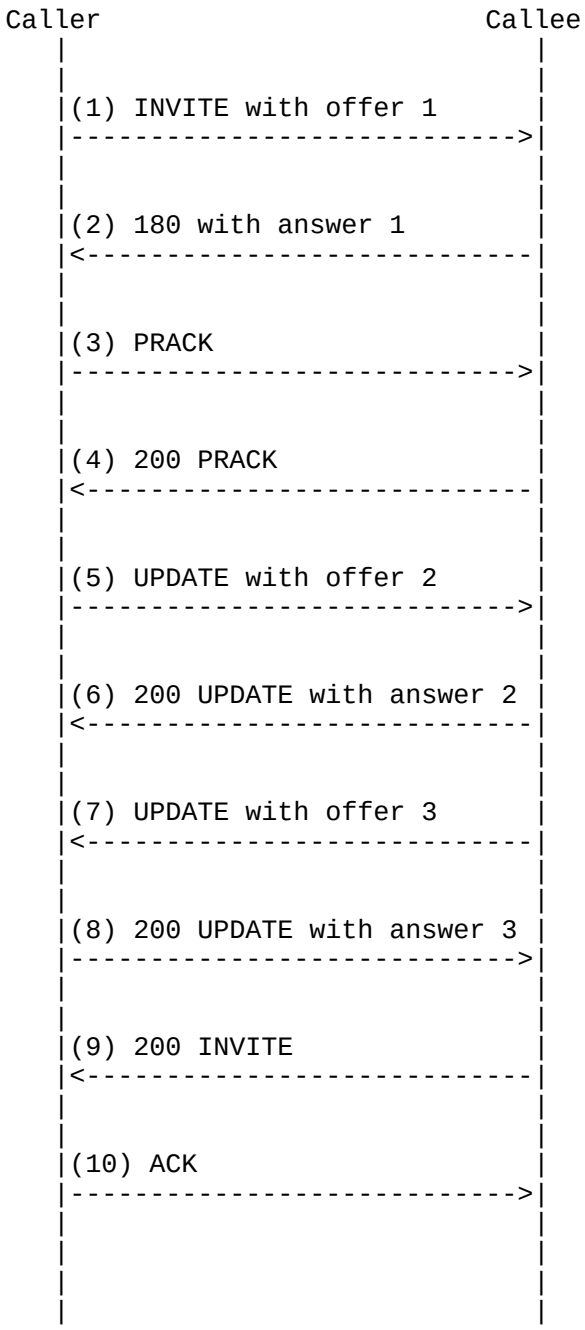

Figure 1: UPDATE Call Flow

9 Security Considerations

The security considerations for UPDATE are identical to those for re-INVITE. It is important that the UPDATE be integrity protected and authenticated as coming from the same source as the entity on the other end of the dialog. RFC 3261 [1] discusses security mechanisms for achieving these functions.

10 IANA Considerations

As per Section 27.4 of RFC 3261 [1], this specification serves as a registration for the SIP UPDATE request method. The information to be added to the registry is:

RFC 3311: This specification serves as the RFC for registering the UPDATE request method.

Method Name: UPDATE

Reason Phrase: Not applicable.

11 Notice Regarding Intellectual Property Rights

12 Normative References

[1] 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.

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

[3] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264, June 2002.

[4] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in the Session Initiation Protocol (SIP)", RFC 3262, June 2002.

[5] Roach, A.B., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

13 Acknowledgements

The author would like to thank Jo Hornsby, Markus Isomaki, Rohan Mahy, and Bob Penfield for their comments.

14 Author's Address

Jonathan Rosenberg
dynamicsoft
72 Eagle Rock Avenue
First Floor
East Hanover, NJ 07936

EMail: jdrosen@dynamicsoft.com

15 Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.