Advanced Instant Messaging Requirements for the Session Initiation Protocol (SIP)
draft-rosenberg-simple-messaging-requirements-00

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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 June 17, 2003.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This specification defines a set of requirements for new capabilities for instant messaging and presence in SIP-based systems.
Table of Contents

1. Introduction .................................................. 3
2. Delivery Status Reporting .................................... 4
3. Is-Composing .................................................. 7
4. Content Capabilities ........................................... 9
5. Page-Mode Groups ............................................. 10
5.1 Invitations to Non-Real-Time Sessions .................... 10
6. IANA Considerations .......................................... 12
7. Security Considerations ....................................... 13
8. Acknowledgments ............................................... 14
   Informative References ....................................... 15
   Author’s Address ............................................. 16
   Intellectual Property and Copyright Statements ............. 17
1. Introduction

The Session Initiation Protocol (SIP) defines several specifications that support Instant Messaging (IM). The SIP MESSAGE method [2] allows for "page-mode" messaging, offering a service similar to Short Message Service (SMS) in wireless networks. A more advanced capability, called session mode messaging, uses the SIP INVITE method to establish a session whose media type is messaging [8][9]. This allows for many SIP capabilities to be directly applied to instant messaging, such as conferencing [10].

However, there are many additional features that exist in current, proprietary IM systems. Some of these features do not require protocol extensions in order to be deployed (IM message archival, for example). However, others do.

This specification provides requirements for a number of advanced IM features which require additional standardization activity to support. It does not cover features which can be achieved with existing protocols and specifications. It is also limited to instant messaging only, and does not consider presence [5].
2. Delivery Status Reporting

In most cases, an IM is delivered immediately to the recipient. Indeed, this is the principal motivation behind the "Instant" in "Instant Messaging". However, in many systems, and instant message can be sent even when the recipient is not available. Indeed, even if they are available when the message is sent, the user may log off before the message can be delivered.

Typically, this is dealt with through a combination of two features. The first is message archival and retrieval. These features allow the intended recipient to retrieve their messages at a later time. To support this, the receiving domain stores the content of the IM, allowing the user to fetch it later. In this regard, it is very similar to existing email systems. Existing protocols, such as IMAP4 [7], can be used for the retrieval functions of IM [[[Note - is there any need for an "IM" profile for IMAP, similar to the "voice" profile for IMAP as specified in RFC 2421 [6]]].

The second feature is message delivery confirmation. This feature allows the sender to know that the receiver has received the message. This feature exists in SMS and in email [12]. A similar function is needed for IP-based instant messaging. Indeed, it is provided in several commercial IM systems, including Wireless Village.

It is important to note that, while delivery status notifications are needed to support storage-and-retrieval of IM, the notifications are needed even for real-time interactive conversations. Specifically, they are needed to support several of the requirements for IM outlined in RFC 2779 [4]. Specifically, requirement 4.2.1 calls out for delivery success and/or failure notifications. While SIP itself provides this basic function for immediately delivered page-mode messages, in session mode, an application layer mechanism is needed. Furthermore, in both page and session modes, group IM will require delivery status notifications in order for the sender to know which, if any, of the recipients got the message.

[[[EDITORS NOTE: The major open issue is the scope. IM is not email; how many features do we really need here? The only clear ones are the ones in the paragraph above which are needed even when the IM is not stored. Perhaps we can support two mechanisms; one, restricted to immediate deliveries, and the other, providing the full delivery status notifications ala [12].]]]

Certainly, much of the email specifications for message delivery confirmation can be reused for IM. However, much of it is email-specific, and IM introduces some new requirements. The following requirements apply to IM delivery status notifications:
REQ-DELNOT-1: It MUST be possible for the sender of an IM to request a delivery notification.

REQ-DELNOT-2: It MUST be possible for the sender of an IM to learn immediately that a delivery notification will, or will not, be sent.

REQ-DELNOT-3: It MUST be possible for the delivery notification to be sent at an arbitrary time in the future.

REQ-DELNOT-4: The delivery notification MUST be capable of indicating that the message was delivered to the intended target.

REQ-DELNOT-5: The delivery notification MUST be capable of indicating whether the message was delivered successfully, or whether, when it was delivered, the recipient generated an error. It MUST be possible for the sender to learn the specific error condition.

REQ-DELNOT-6: The delivery notification MUST indicate the time of message delivery.

REQ-DELNOT-7: The delivery notification MUST indicate the specific message which was delivered.

REQ-DELNOT-8: Delivery notifications MUST operate for both page mode and session mode.

REQ-DELNOT-9: In order to support interaction conversations where the sender can re-type their message if it is not received, the delivery notifications MUST be sent rapidly when the message can be immediately delivered. In this case, rapidly is loosely defined, but generally, fast enough to support an interactive conversation.

REQ-DELNOT-10: It MUST be possible for the message sender (the recipient of the notification) to authenticate the sender of the notification. There is no explicit requirement for confidentiality of the notification.

REQ-DELNOT-11: As it is anticipated that this mechanism will be used frequently from wireless devices, it SHOULD keep overhead to a minimum, and in particular, SHOULD NOT provide extraneous information not relevant to the above requirements.

REQ-DELNOT-12: When an IM is sent to a group, there MUST be delivery notifications generated about the delivery of the IM to each group participant.
REQ-DELNOT-13: REQ-DELNOT-12 MUST support recursive groups.

REQ-DELNOT-14: The identity of the ultimate recipient MUST be made known to the message sender.

REQ-DELNOT-15: The notification MAY contain the content of the original IM [[EDITORS NOTE: is this really needed for IM?]]

REQ-DELNOT-16: Any error condition reported by the notification MAY contain a textual description of the error meant for human consumption [[EDITORS NOTE: Do we need this?]]

REQ-DELNOT-17: If an IM is being relayed through a gateway, for example, to SMS, the delivery report SHOULD indicate such a condition [[EDITORS NOTE: Do we need this?]]

REQ-DELNOT-18: The delivery notification MUST indicate the Content-Type of the message that was delivered.

REQ-DELNOT-19: The delivery notification MUST indicate the Content-Length of the message that was delivered.

REQ-DELNOT-20: The delivery notification MUST indicate the To header field from the message that was delivered.

REQ-DELNOT-21: The delivery notification MUST indicate the Expires header field of the message that was delivered.
3. Is-Composing

Many commercial instant messaging and presence systems provide a feature generally referred to as "is-typing". This feature is used during instant messaging chat sessions. Whenever one user is in the process of typing a message to another user, the recipient-to-be can see that a message is in progress. This avoids a common problem where both participants are typing replies at the same time, so that the resulting stream of conversation is not well synchronized.

Generalizing this concept, the idea is really to allow one participant to inform another participant that they are composing a message of some type. By conveying a type, a broader set of features can be enabled. For example, if one user indicates that they are composing a message of type audio/basic, the other user will know that a voice IM is coming.

REQ-COMP-1: The is-composing feature must work with instant messaging sessions [8].

REQ-COMP-2: Either side in the session should be able to indicate that they can receive the indicators. The indicators must not be sent from A to B if B does not explicitly indicate that they can receive them.

REQ-COMP-3: It must be possible for indicators to be sent in only one direction, i.e., A sends them to B, but B does not send them to A.

REQ-COMP-4: It must be possible for usage of the indicators to be added or removed to any IM session after the session has begun.

REQ-COMP-5: The indicator must be able to inform the recipient that the sender has begun composing a message.

REQ-COMP-6: The indicator must be able to inform the recipient that the sender has stopped composing a message.

REQ-COMP-7: The indicator must be able to convey the MIME type of the message that is being composed.

REQ-COMP-8: The indicator must be able to convey the content-disposition of the message that is being composed. [Do we want this requirement?]

REQ-COMP-9: The indicator must be synchronized with the message stream itself. That is, if a recipient gets an indicator that a user has stopped composing a message, and they also get a message,
the recipient must be able to know which came first.

REQ-COMP-10: It must be possible to provide end-to-end message integrity and authentication over the indicators.

REQ-COMP-11: It must be possible to associate the is-composing indicator with a particular instant messaging session.

REQ-COMP-12: It should be possible to interwork is-composing indicators between CPIM compliant systems, possibly with some loss of functionality, but with integrity and authentication intact.

REQ-COMP-13: It should be possible for is-composing indicators to work, possibly with loss of functionality, in page mode. [Do we want this requirement?]

REQ-COMP-14: The is-composing indicator should not result in an increase on the load of proxies.

REQ-COMP-15: It must be possible to receive delivery confirmation reports for is-composing indicators [Do we need this requirement?]

REQ-COMP-16: The overhead of the indicators should be minimal.
4. Content Capabilities

Although traditionally used with text, an IM can contain any kind of content. There is an increasing trend to send multimedia content, including audio, video, and even applications, over IM. However, recipients may not wish to receive content that they do not understand, or is over a particular size limit.

Handling these "content capabilities" is done differently for page mode and session mode. In session mode, the initial offer/answer exchange [3] can be used to convey content capabilities. Indeed, the messaging sessions mechanism allows for negotiation of supported content types. However, some additional aspects of negotiation are required:

REQ-CONTENT-1: A UA MUST be able to indicate the maximum message size it is willing to receive.

In page mode messaging, a 413 response can be sent if a MESSAGE request is too large. However, there is no way to indicate what the largest allowed size is:

REQ-CONTENT-2: A 413 response MUST be capable of indicating the maximum allowed message size.

Note that, there is no requirement to support transcoding of content at an intermediary in order to reduce the size of a sent message to match that of a recipient.
5. Page-Mode Groups

There is no explicit support for groups in page mode. Supporting groups in session mode is trivial, and is completely handled through the SIP conferencing specifications [10]. While there is no expectations that the same level of features will be available in page mode conferencing as session mode, some minimal features are desirable.

REQ-GROUP-1: It MUST be possible to address a page-mode IM to a group.

REQ-GROUP-2: Each recipient of a group page IM MUST be capable of determining the set of other recipients that got the request.

REQ-GROUP-3: It MUST be possible for a user to send to an ad-hoc group, where the identities of the recipients are carried in the message itself.

REQ-GROUP-4: It MUST be possible for the recipient of a group IM to send a message to all other participants that received the same group IM (i.e., Reply-To-All).

[[Editors NOTE: It is not clear at all that we want to support any of this. Session mode provides a much more comprehensive conferencing story. Do we really want to add features for page mode??]]

5.1 Invitations to Non-Real-Time Sessions

SIP fundamentally deals with real-time sessions. That is, it allows users to invite other users to communicate using some kind of interactive media. However, there are many other types of "sessions", in the broad sense of the word, that one can be invited to. As an example, one user could invite another user to join an email mailing list, to join a conference call occurring next week, or to view a web site.

Specific desired capabilities are:

REQ-NRT-1: It MUST be possible to send a user a message requesting that they perform some specific action.

REQ-NRT-2: The set of possible actions MUST include the ability to request that a user add another user to their buddy list.

REQ-NRT-3: The set of possible actions MUST include the ability to request the user to join a meeting scheduled at a specific time.
REQ-NRT-4: The set of possible actions MUST include the ability to request the user to view a certain URL.

REQ-NRT-5: The set of possible actions MUST include the ability to request the user to join a specific group (such as a page-mode group IM list).

REQ-NRT-6: The set of actions MUST be easily extensible.

REQ-NRT-7: It MUST be possible for the sender to cancel the request, i.e., ask that the user not bother to perform the previous requested action. Of course, there is no expectation that the request is honored, and no way to enforce it. The only requirement is the ability to convey this desire.

REQ-NRT-8: It MUST be possible for the sender to learn when the recipient has performed the desired action.

REQ-NRT-9: It MUST be possible for the sender to learn that the recipient has received the request to perform the desired action.

REQ-NRT-10: It MUST be possible for the recipient to indicate that they accept the invitation, reject it, or will defer it (consider it later).

REQ-NRT-11: It MUST be possible for REQ-NRT-10, REQ-NRT-9 and REQ-NRT-8 to occur at an arbitrarily long period of time after the invitation was issued.

This capability is a hybrid between a traditional SIP INVITE and a SIP MESSAGE. It is like INVITE in that it is accepted or rejected, and can be cancelled. It is not like INVITE in that the acceptances or rejections can come at any time, even days after the invitation. In that sense, it is more like a MESSAGE.

One solution to these requirements is to use IM, and for the content to indicate an invitation of sorts. As an example, an invitation to a meeting can be done using an iCal object [11] carried in a MESSAGE request. However, cancelling the invitation, and indicating acceptances, would all be specified using iCal specific parameters. They would not be reusable for other invitation types.

As such, an alternative solution is to define several general primitives for this operation.

[[EDITORS NOTE: It is far from clear that we want to do any standards work here. This section was included because this feature is supported in one of the commercial systems.]]
6. IANA Considerations

There are no IANA Considerations associated with this specification.
7. Security Considerations

Security requirements are discussed above where relevant.
8. Acknowledgments

This draft includes requirements contributed by Aki Niemi [13].
Informative References


progress), October 2002.

Author’s Address

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

Phone: +1 973 952-5000
EMail: jdrosen@dynamicsoft.com
URI: http://www.jdrosen.net
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property 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; neither does it represent that it has made any effort to identify any such rights. Information on the IETF’s procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Acknowledgement

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