Internet Engineering Task Force (IETF)
Request for Comments: 7701
Category: Standards Track
ISSN: 2070-1721

A. Niemi

M. Garcia-Martin
Ericsson

G. Sandbakken
Cisco Systems

December 2015

Multi-party Chat Using the Message Session Relay Protocol (MSRP)

Abstract

The Message Session Relay Protocol (MSRP) defines a mechanism for sending instant messages (IMs) within a peer-to-peer session, negotiated using the Session Initiation Protocol (SIP) and the Session Description Protocol (SDP). This document defines the necessary tools for establishing multi-party chat sessions, or chat rooms, using MSRP.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7701.

Table of Contents

1. Introduction

The Message Session Relay Protocol (MSRP) [RFC4975] defines a mechanism for sending a series of instant messages within a session. The Session Initiation Protocol (SIP) [RFC3261] in combination with the Session Description Protocol (SDP) [RFC4566] allows for two peers to establish and manage such sessions.

In another application of SIP, a User Agent (UA) can join in a multi-party conversation called a "conference" that is hosted by a specialized UA called a "focus" [RFC4353]. Such a conference can naturally involve MSRP sessions. It is the responsibility of an entity handling the media to relay IMs received from one participant to the rest of the participants in the conference.

Several such systems already exist in the Internet. Participants in a chat room can be identified with a pseudonym or nickname and can decide whether their real identifier is disclosed to other participants. Participants can also use a rich set of features such as the ability to send private instant messages to other participants.

Similar conferences supporting chat room functionality are already available today. For example, Internet Relay Chat (IRC) [RFC2810], Extensible Messaging and Presence Protocol (XMPP): Core [RFC6120], as well as many other proprietary systems. Specifying equivalent functionality for MSRP-based systems eases interworking between these systems.

This document defines requirements, conventions, and extensions for providing private messages and nickname management in centralized chat rooms with MSRP. Participants in a chat room can be identified by a pseudonym and decide if their real identifier should be disclosed to other participants. This memo uses the SIP Conferencing Framework [RFC4353] as a design basis. It also aims to be compatible with "A Framework for Centralized Conferencing" [RFC5239]. Should requirements arise, future mechanisms for providing similar functionality in generic conferences might be developed, for example, where the media is not only restricted to MSRP. The mechanisms described in this document provide a future compatible short-term solution for MSRP centralized chat rooms.

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 BCP 14 [RFC2119] and indicate requirement levels for compliant implementations.

This memo deals with "Tightly Coupled SIP Conferences" as defined in the SIP Conferencing Framework [RFC4353] and adopts the terminology from that document. In addition, we introduce some new terms:

Nickname: a pseudonym or descriptive name associated with a participant. See Section 7 for details.

Multi-party Chat: an instance of a tightly coupled conference, in which the media exchanged between the participants consist of MSRP-based IMs. Also known as a chat room.

Chat Room: a synonym for a multi-party chat.

Chat Room URI: a URI that identifies a particular chat room and that is a synonym of a "Conference URI" as defined in RFC 4353 [RFC4353].

Sender: the chat room participant who originally created an IM and sent it to the chat room server for further delivery.

Recipient: the destination chat room participant(s). This defaults to the full conference participant list minus the IM Sender.

MSRP Switch: a media-level entity that is an MSRP endpoint. It is a special MSRP endpoint that receives MSRP messages and delivers them to the other chat room participants. The MSRP switch has a similar role to a conference mixer with the exception that the MSRP switch does not actually "mix" together different input media streams; it merely relays the messages between chat room participants.

Private IM: an IM sent in a chat room intended for a single participant. Generally speaking, a private IM is seen by the MSRP switch, in addition to the sender and recipient. A private IM is usually rendered distinctly from the rest of the IMs, indicating that the message was a private communication.

Anonymous URI: a URI concealing the participant’s SIP address of record (AOR) from the other participants in the chat room. The allocation of such a URI is out of scope of this specification. An anonymous URI must be valid for the length of the chat room

session and will be utilized by the MSRP switch to forward messages to and from anonymous participants. Privacy and anonymity are discussed in greater detail in RFC 3323 [RFC3323] and RFC 3325 [RFC3325].

Conference Event Package: a notification mechanism that allows conference participants to learn conference information including roster and state changes in a conference. This would typically be the mechanisms defined in "A Session Initiation Protocol (SIP) Event Package for Conference State" [RFC4575] or "Conference Event Package Data Format Extension for Centralized Conferencing (XCON)" [RFC6502].

Identifier: a string used to recognize or establish as being a particular user.

To log in: to enter identifying data, as a name or password, into a chat room, so as to be able to do work with the chat room.

3. Motivations and Requirements

Although conference frameworks describing many types of conferencing applications already exist, such as the one in "A Framework for Centralized Conferencing" [RFC5239] and the SIP Conferencing Framework [RFC4353], the exact details of session-based instant messaging conferences (chat rooms) are not well-defined at the moment.

To allow interoperable chat implementations, for both conference-aware and conference-unaware UAs, certain conventions for MSRP chat rooms need to be defined. It also seems beneficial to provide a set of features that enhance the baseline multi-party MSRP in order to be able to create systems that have functionality on par with existing chat systems as well as to enable the building of interworking gateways to these existing chat systems.

We define the following requirements:

REQ-1: A basic requirement is the existence of a chat room, where participants can join and leave the chat room and exchange IMs with the rest of the participants.

REQ-2: A recipient of an IM in a chat room must be able to determine the identifier of the sender of the message. Note that the actual identifier depends on the one that was used by the sender when joining the chat room.

REQ-3: A recipient of an IM in a chat room must be able to determine the identifier of the recipient of received messages. For instance, the recipient of the message might be the entire chat room or a single participant (i.e., a private message). Note that the actual identifier may depend on the one that was used by the recipient when he or she joined the chat room.

REQ-4: It must be possible to send a message to a single participant within the chat room (i.e., a private IM).

REQ-5: A chat room participant may have a nickname or pseudonym associated with their real identifier.

REQ-6: It must be possible for a participant to change their nickname during the progress of the chat room session.

REQ-7: It must be possible for a participant to be known only by an anonymous identifier and not their real identifier by the rest of the chat room.

REQ-8: It must be possible for chat room participants to learn the chat room capabilities described in this document.

4. Overview of Operation

Before a chat room can be entered, it must be created. Users wishing to host a chat room themselves can, of course, do just that; their UA simply morphs from an ordinary UA into a special purpose one called a "Focus UA". Another, commonly used setup is one where a dedicated node in the network functions as a Focus UA.

Each chat room has an identifier of its own: a SIP URI that participants use to join the chat room, e.g., by sending an INVITE request to it. The conference focus processes the invitations, and as such, maintains SIP dialogs with each participant. In a multi-party chat, or chat room, MSRP is one of the established media streams. Each chat room participant establishes an MSRP session with the MSRP switch, which is a special purpose MSRP application. The MSRP sessions can be relayed by one or more MSRP relays, which are specified in RFC 4976 [RFC4976]. This is illustrated in Figure 1.

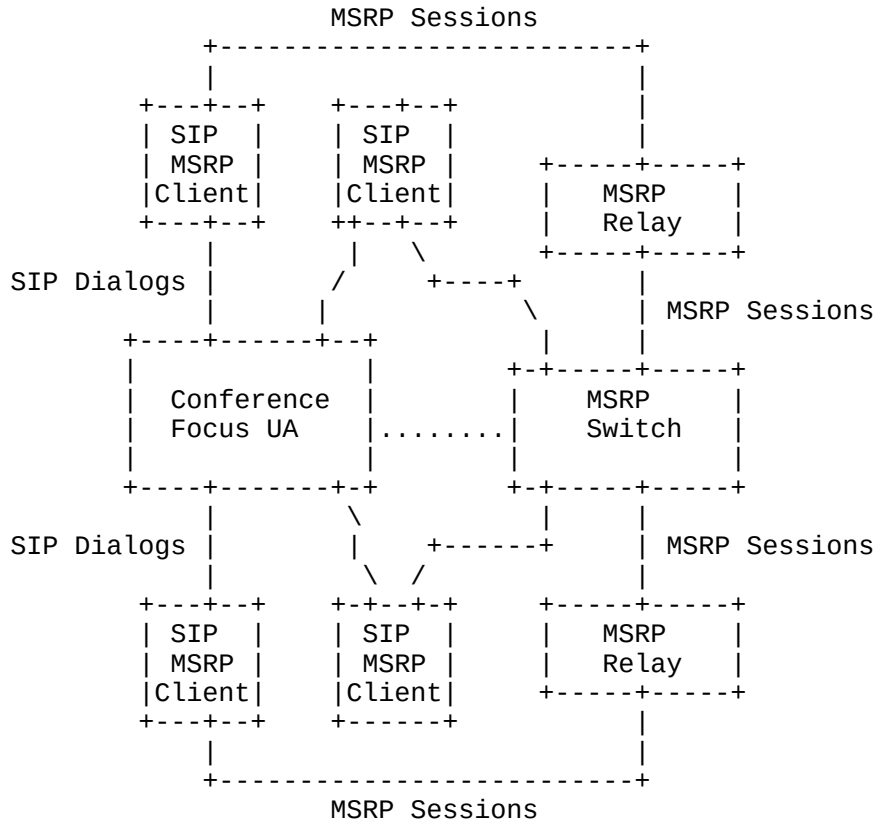

Figure 1: Multi-party Chat Overview Shown with MSRP Relays and a Conference Focus UA

The MSRP switch is similar to a conference mixer in that it both handles media sessions with each of the participants and bridges these streams together. However, unlike a conference mixer, the MSRP switch merely forwards messages between participants: it doesn't actually mix the streams in any way. The system is illustrated in Figure 2.

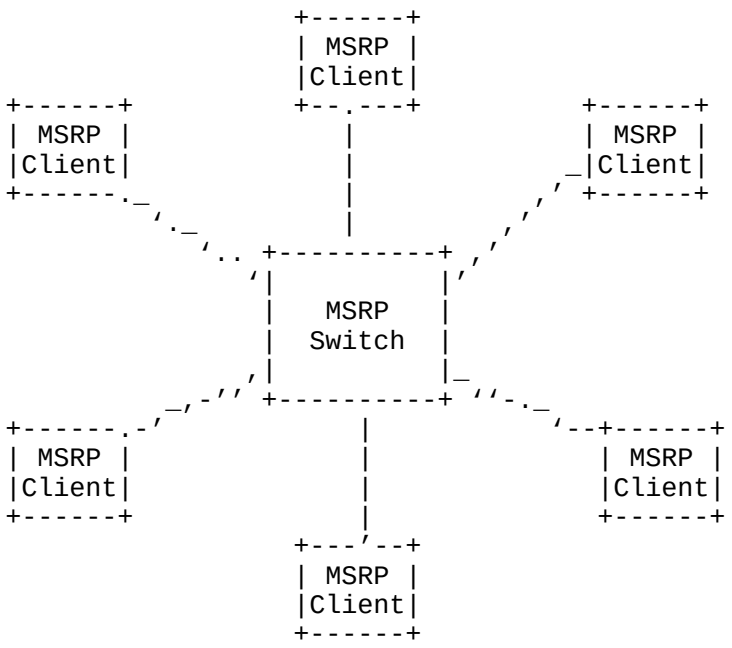

Figure 2: Multi-party Chat in a Centralized Chat Room

Typically, chat room participants also subscribe to a conference event package to gather information about the conference roster in the form of conference state notifications. For example, participants can learn about other participants' identifiers, including their nicknames.

All messages in the chat room use the Message/CPIM wrapper content type [RFC3862], to distinguish between private and regular messages. When a participant wants to send an instant message to the chat room, it constructs an MSRP SEND request and submits it to the MSRP switch including a regular payload (e.g., a Message/CPIM message that contains text, HTML, an image, etc.). The Message/CPIM To header is set to the chat room URI. The switch then fans out the SEND request to all of the other participants using their existing MSRP sessions.

A participant can also send a private IM addressed to a participant whose identifier has been learned, e.g., via a conference event package. In this case, the sender creates an MSRP SEND request with a Message/CPIM wrapper whose To header contains not the chat room URI but the recipient's URI. The MSRP switch then forwards the SEND request to that recipient. This specification supports the sending of private messages to one and only one recipient. However, if the

recipient is logged in from different endpoints, the MSRP switch will distribute the private message to each endpoint at which the recipient is logged in.

We extend the current MSRP negotiation that takes place in SDP [RFC4566] to allow participants to learn whether the chat room supports and is willing to accept (e.g., due to local policy restrictions) certain MSRP functions defined in this memo, such as nicknames or private messaging. This is achieved by a new 'chatroom' attribute in SDP (please refer to Section 8 for a detailed description).

Naturally, when a participant wishes to leave a chat room, it sends a SIP BYE request to the Focus UA and terminates the SIP dialog with the focus and MSRP sessions with the MSRP switch.

This document assumes that each chat room is allocated its own SIP URI. A user joining a chat room sends an INVITE request to that SIP URI, and, as a result, a new MSRP session is established between the user and the MSRP switch. It is assumed that an MSRP session is mapped to a chat room. If a user wants to join a second chat room, he creates a different INVITE request, through a different SIP dialog, which leads to the creation of a second MSRP session between the user and the MSRP switch. Notice that these two MSRP sessions can still be multiplexed over the same TCP connection as per regular MSRP procedures. However, each chat room is associated with a unique MSRP session and a unique SIP dialog.

4.1. Policy Attributes of the Chat Room

The Conference Framework with SIP [RFC4353] introduces the notion of a Conference Policy as "The complete set of rules governing a particular conference." A chat room is a specialized type of conference, and the conference policy is sometimes extended with new chat-specific rules. This section lists all the Conference Policy attributes used by the present document and refers to sections in the document where the usage of these attributes are described in greater detail.

Nicknames: Whether the chat room accepts users to be recognized with a nickname. See Sections 7, 7.1, and 8 for details. Also, the scope of uniqueness of the nickname: the chat room (conference instance), a realm or domain, a server, etc.

Nickname quarantine: The quarantine to be imposed on a nickname once it is not currently in use (e.g., because the participant holding this nickname abandons the chat room), prior to the wide availability of this nickname to other users. This allows the initial holder of the nickname to join the chat room during the quarantine period and claim the same nickname they were previously using. See Section 11 for details.

Private messaging: Whether the chat room allows users to send private messages to other users of the chat room through the MSRP switch. See Sections 6.2 and 8 for details.

Deletion of the chat room: Whether the chat room can be deleted when the creator leaves the chat room or through an out-of-band mechanism. See Section 5.3 for details.

Simultaneous access: Whether a user can log in from different endpoints using the same identity. See Sections 6.1 and 6.2 for details.

Force TLS transport: Whether the MSRP switch accepts only Transport Layer Security (TLS) as an MSRP transport, in an effort to guarantee confidentiality and privacy. See Section 11 for details.

Maximum message size in congested MSRP sessions: The maximum size of messages that can be distributed to a user over a congested MSRP session. See Section 6.4 for details.

Chunk reception timer: The value of a time that controls the maximum time that the MSRP switch is waiting for the reception of different chunks belonging to the same message. If the timer expires, the MSRP switch will discard the associated message state. See Section 6.1 for details.

Supported wrapped media types: The list of media types that the MSRP switch accepts in Message/CPIM wrappers sent from participants. This list is included in the 'accept-wrapped-types' attribute of the MSRP message media line in SDP. If the MSRP switch accepts additional media types to those explicitly listed, a "*" is added to the list. A single "*" indicates that the chat room accepts any wrapped media type.

5. Creating, Joining, and Deleting a Chat Room

5.1. Creating a Chat Room

Since we consider a chat room a particular type of conference having MSRP media, the methods defined by the SIP Conference Framework [RFC4353] for creating conferences are directly applicable to a chat room.

Once a chat room is created, it is identified by a SIP URI, like any other conference.

5.2. Joining a Chat Room

Participants usually join the chat room by sending an INVITE request to the chat room URI. The chat room then uses regular SIP mechanisms to authenticate the participant. This may include, e.g., client certificates, SIP Digest authentication [RFC3261], asserted network identity [RFC3325], SIP Identity header field [RFC4474], etc. As long as the user is authenticated, the INVITE request is accepted by the focus and the user is brought into the actual chat room.

This specification requires all IMs to be wrapped in a Message/CPIM wrapper [RFC3862]. Therefore, the 'accept-types' attribute for the MSRP message media in both the SDP offer and answer need to include at least the value 'Message/CPIM' (notice that RFC 4975 [RFC4975] mandates this 'accept-types' attribute in SDP). If the 'accept-types' attribute does not contain the value 'Message/CPIM', the conference focus will reject the request. The actual instant message payload type is negotiated in the 'accept-wrapped-types' attribute in SDP (see RFC 4975 [RFC4975] for details). There is no default wrapped type. Typical wrapped type values can include text/plain, text/html, image/jpeg, image/png, audio/mp3, etc. It is RECOMMENDED that participant endpoints add an 'accept-wrapped-types' attribute to the MSRP 'message' media line in SDP, where the supported wrapped types are declared, as per RFC 4975 procedures [RFC4975].

The MSRP switch needs to be aware of the URIs of the participant (SIP, tel, or IM URIs) in order to validate messages sent from this participant prior to their forwarding. This information is known to the focus of the conference. Therefore, an interface between the focus and the MSRP switch is assumed. However, the interface between the focus and the MSRP switch is outside the scope of this document.

Conference-aware participants will detect that the peer is a focus due to the presence of the "isfocus" feature tag [RFC3840] in the Contact header field of the 200-class response to the INVITE request. Conference-unaware participants will not notice it is a focus, and

cannot apply the additional mechanisms defined in this document. Participants are also aware that the mixer is an MSRP switch due to the presence of a 'message' media type and either TCP/MSRP or TCP/TLS/MSRP as the protocol field in the media line of SDP [RFC4566].

The conference focus of a chat room MUST only use a Message/CPIM [RFC3862] top-level wrapper as a payload of MSRP messages, and the focus MUST declare it in the SDP offer or answer as per regular procedures in RFC 4975 [RFC4975]. This implies that if the conference focus receives, from a participant's endpoint, an SDP offer that does not include the value 'Message/CPIM' in the 'accept-types' attribute for the MSRP message media line, the conference focus SHOULD either reject the MSRP message media stream or reject the complete SDP offer by using regular SIP or SDP procedures (e.g., creating an SDP answer that sets to zero the port of the MSRP message media line, responding the INVITE with a 488 response, etc.).

If the conference focus accepts the participant's SDP offer, when the conference focus generates the SDP answer, it MUST set the 'accept-types' attribute for the MSRP message media line to a value of 'Message/CPIM'. This specification requires all IMs to be wrapped in a Message/CPIM wrapper, therefore, the 'accept-types' attribute in this SDP body contains a single value of 'Message/CPIM'. The actual IM payload type is negotiated in the 'accept-wrapped-types' attribute in SDP (see RFC 4975 [RFC4975] for details). The conference focus MAY also add an 'accept-wrapped-types' attribute to the MSRP message media line in SDP containing the supported wrapped types, according to the supported wrapped media types policy.

> Note that the Message/CPIM wrapper is used to carry the sender information that, otherwise, it will not be available to the recipient. Additionally, the Message/CPIM wrapper carries the recipient information (e.g., To and Cc headers).

If the UA supports anonymous participation and the user chooses to use it, the participant's UA SHOULD do at least one of these options:

(a) provide an anonymous URI in SIP headers that otherwise reveal identifiers. Please refer to RFC 3323 [RFC3323] for a detailed description of which headers are subject to reveal identifiers and how to populate them; or

(b) trust the conference focus and request privacy of their URI, e.g., by means of the SIP Privacy header field [RFC3323], network asserted identity [RFC3325], or a similar privacy mechanism.

If the participant has requested privacy, the conference focus MUST expose a participant's anonymous URI through the conference event package [RFC4575].

The conference focus of a chat room learns the supported chat room capabilities in the endpoint by means of the 'chatroom' attribute exchanged in the SDP offer/answer (please refer to Section 8 for a detailed description). The conference focus MUST inform the MSRP switch of the chat room capabilities of each participant that joins the chat room (note that the interface defined between the conference focus and the MSRP switch is outside the scope of this specification). This information allows the MSRP switch, e.g., to avoid the distribution of private messages to participants whose endpoints do not support private messaging.

5.3. Deleting a Chat Room

As with creating a conference, the methods defined by the SIP Conference Framework [RFC4353] for deleting a conference are directly applicable to a chat room. The MSRP switch will terminate the MSRP sessions with all the participants.

Deleting a chat room is an action that heavily depends on the policy of the chat room. For example, the policy can determine whether the chat room is deleted when the creator leaves the room or whether an out-of-band mechanism is responsible for the deletion.

6. Sending and Receiving Instant Messages

6.1. Regular Messages

This section describes the conventions used to send and receive IMs that are addressed to all the participants in the chat room. These are sent over a regular MSRP SEND request that contains a Message/CPIM wrapper [RFC3862] that, in turn, contains the desired payload (e.g., text, image, video clip, etc.).

When a chat room participant wishes to send an instant message to all the other participants in the chat room, it constructs an MSRP SEND request according to the procedures specified in RFC 4975 [RFC4975]. The sender MAY choose the desired MSRP report model (e.g., populate the Success-Report and Failure-Report MSRP header fields).

On sending a regular message, the sender MUST populate the To header of the Message/CPIM wrapper with the URI of the chat room. The sender MUST also populate the From header of the Message/CPIM wrapper with a proper identifier by which the user is recognized in the chat room. Identifiers that can be used (among others) are:

- A SIP URI [RFC3261] representing the participant's address-of-record
- A tel URI [RFC3966] representing the participant's telephone number
- An IM URI [RFC3860] representing the participant's instant messaging address
- An anonymous URI representing the participant's anonymous address

If the participant wants to remain anonymous, the participant's endpoint MUST populate an anonymous URI in the From header of the Message/CPIM wrapper. Other participants of the chat room will use this anonymous URI in the To header of the Message/CPIM wrapper when sending private messages. Notice that in order for the anonymity mechanism to work, the anonymous URI MUST NOT reveal the participant's SIP AOR. The mechanism for acquiring an anonymous URI is outside the scope of this specification.

An MSRP switch that receives a SEND request from a participant SHOULD first verify that the From header field of the Message/CPIM wrapper is correctly populated with a valid URI of a participant. This imposes a requirement for the focus of the conference to inform the MSRP switch of the URIs by which the participant is known, in order for the MSRP switch to validate messages. Section 6.3 provides further information with the actions to be taken in case this validation fails.

Then the MSRP switch should inspect the To header field of the Message/CPIM wrapper. If the MSRP switch receives a message containing several To header fields in the Message/CPIM wrapper the MSRP switch MUST reject the MSRP SEND request with a 403 response, as per procedures in RFC 4975 [RFC4975]. Then, if the To header field of the Message/CPIM wrapper contains the chat room URI and there are no other To header fields, the MSRP switch can generate a copy of the SEND request to each of the participants in the chat room except the sender. The MSRP switch MUST NOT modify the content received in the SEND request. However, the MSRP switch MAY re-chunk any of the outbound MSRP SEND requests.

When generating a copy of the SEND request to each participant in the chat room, the MSRP switch MUST evaluate the wrapped media types that the recipient is able to accept. This was learned through the 'accept-wrapped-types' attribute of the MSRP message media line in SDP. If the MSRP switch is aware that the media type of the wrapped content is not acceptable to the recipient, the MSRP switch SHOULD NOT forward this message to that endpoint. Note that this version of

the specification does not require the MSRP switch to notify the sender about this failure. Extensions to this specification may improve handling of unknown media types.

Note that the MSRP switch does not need to wait for the reception of the complete MSRP chunk or MSRP message before it starts the distribution to the rest of the participants. Instead, once the MSRP switch has received the headers of the Message/CPIM wrapper, it SHOULD start the distribution process. But, bear in mind that the MSRP switch SHOULD still implement some sanity checking. Please refer to the security considerations in Section 11 for further details.

When forwarding chunked messages as soon as they are received, the Message/CPIM wrapper is only present at the beginning of the message, typically within the first chunk. Subsequent chunks will contain the rest of the message, but not the Message/CPIM headers. Therefore, an MSRP switch that receives a subsequent message may face challenges in determining the correct list of recipients of the message. An MSRP switch that uses this fast forwarding procedure MUST temporarily store the Message-ID of the MSRP message to correlate the different chunks; it MUST also temporarily store the list of recipients to which the initial chunks were delivered. The MSRP switch SHOULD forward subsequent chunks only to those recipients who were sent the initial chunks, except if the MSRP switch has knowledge that one of the recipients of the initial chunks has dropped from the chat room. This behavior also avoids new participants who had joined the chat room when the first chunk was distributed from receiving subsequent chunks that would otherwise need to be discarded.

Once the MSRP switch receives the last chunk of a message, and that chunk is successfully sent to each of the recipients, the MSRP switch discards the temporary storage of MSRP Message-ID and the associated list of recipients.

In some occasions, a sender might suffer a transport error condition (such as loss of connectivity or depletion of battery) that makes the sending of a message incomplete, e.g., some chunks were received by the MSRP switch, but not all of them. This is a behavior already considered in the core MSRP specification (see RFC 4975 [RFC4975] Section 5.4). The problem in the context of a chat room lies with the use of temporary storage for fast forwarding. In order to prevent attacks related to the exhaustion of temporary storage of chunked messages, on receiving a first chunk of a message, where the MSRP switch is using the fast forward method, the MSRP switch MUST set a chunk reception timer for controlling the reception of the remaining chunks.

This chunk reception timer can be reset every time a new chunk of the same message is received. When this timer expires, the MSRP switch MUST consider that the sending of the message was aborted, and it MAY discard all the message state associated with it, including the Message-ID and the list of recipients. Additionally, if this chunk reception timer expires, the MSRP switch MAY choose to send an abort chunk (i.e., one with the "#" flag set) to each to the recipients. This is just an optimization, since MSRP endpoints need to be able to handle incomplete messages as per regular MSRP.

The specific value of this chunk reception timer is not standardized; it is subject of local policy. However, it is recommended not to be a short value. For example, a time interval on the order of a normal TCP timeout (i.e., around 540 seconds) would be reasonable. A value on the order of a few seconds would not.

An MSRP endpoint that receives a SEND request from the MSRP switch containing a Message/CPIM wrapper SHOULD first inspect the To header field of the Message/CPIM wrapper. If the To header field is set to the chat room URI, it should render it as a regular message that has been distributed to all the participants in the chat room. Then, the MSRP endpoint SHOULD inspect the From header field of the Message/CPIM wrapper to identify the sender. The From header field will include a URI that identifies the sender. The endpoint might have also received further identifier information through a subscription to a conference event package.

It is possible that a participant, identified by a SIP AoR or other valid URI, joins a chat room simultaneously from two or more different SIP UAs. It is recommended that the MSRP switch implements means to map a URI to two or more MSRP sessions. If the policy of the chat room allows simultaneous access, the MSRP switch MUST copy all regular messages intended to the recipient through each MSRP session mapped to the recipient’s URI.

6.2. Private Messages

This section describes the conventions used to send and receive private IMs, i.e., IMs that are addressed to one participant of the chat room rather than to all of them. The chat room has a local policy that determines whether or not private messages are supported. A chat room can signal support for private messages using the ’chatroom’ attribute in SDP (please refer to Section 8 for a detailed description).

When a chat room participant wishes to send a private IM to a participant in the chat room, it follows the same procedures to create a SEND request as for regular messages (Section 6.1). The

only difference is that the MSRP endpoint MUST populate a single To header of the Message/CPIM wrapper with the identifier of the intended recipient. The identifier can be SIP, tel, and im URIs typically learned from the information received in notifications of a conference event package.

> This version of the specification does not support sending a private message to multiple recipients, i.e., the presence of multiple To headers in the Message/CPIM wrapper of the MSRP SEND request. This is due to added complexity, for example, with the need to determine whether a message was not delivered to some of the intended recipients. Implementations that still want to recreate this function can send a series of single private messages, one private message per intended recipient. The endpoint can correlate this series of messages and create the effect of a private message addressed to multiple recipients.

As for regular messages, an MSRP switch that receives a SEND request from a participant SHOULD first verify that the From header field of the Message/CPIM wrapper is correctly populated with a valid URI (i.e., the URI is a participant of this chat room). Section 6.3 provides further information regarding the actions to be taken in case this validation fails.

Then, the MSRP switch inspects the To header field of the Message/CPIM wrapper. If the MSRP switch receives a message containing several To header fields in the Message/CPIM wrapper, the MSRP switch MUST reject the MSRP SEND request with a 403 response, as per procedures in RFC 4975 [RFC4975]. Then, the MSRP switch verifies that the To header of the Message/CPIM wrapper matches the URI of a participant of the chat room. If this To header field does not contain the URI of a participant of the chat room or if the To header field cannot be resolved (e.g., caused by a mistyped URI), the MSRP switch MUST reject the request with a 404 response. This new 404 status code indicates a failure to resolve the recipient URI in the To header field of the Message/CPIM wrapper.

> Notice the importance of the From and To headers in the Message/CPIM wrapper. If an intermediary modifies these values, the MSRP switch might not be able to identify the source or intended destination of the message, resulting in a rejection of the message.

Finally, the MSRP switch verifies that the recipient supports private messages. If the recipient does not support private messages, the MSRP switch MUST reject the request with a 428 response. This new 428 response indicates that the recipient does not support private messages. Any potential REPORT request that the MSRP switch sends to

the sender MUST include a Message/CPIM wrapper containing the original From header field included in the SEND request and the To header field of the original Message/CPIM wrapper. The MSRP switch MUST NOT forward private messages to a recipient that does not support private messaging.

If successful, the MSRP switch should search its mapping table to find the MSRP sessions established toward the recipient. If a match is found, the MSRP switch MUST create a SEND request and MUST copy the contents of the sender's message to it.

An MSRP endpoint that receives a SEND request from the MSRP switch does the same validations as for regular messages (Section 6.1). If the To header field is different from the chat room URI, the MSRP endpoints know that this is a private message. The endpoint should render who it is from based on the value of the From header of the Message/CPIM wrapper. The endpoint can also use the sender's nickname, possibly learned via a conference event package, to render the sender of the message, instead of using the sender's actual URI.

As with regular messages, if the policy of the chat room allows simultaneous access, the MSRP switch MUST copy all private messages intended to the recipient through each MSRP session mapped to the recipient's URI.

6.3. MSRP Reports and Responses

This section discusses the common procedures for regular and private messages with respect to MSRP reports and responses. Any particular procedure affecting only regular messages or only private messages is discussed in the previous sections (Sections 6.1 or 6.2, respectively).

MSRP switches MUST follow the success report and failure report handling described in Section 7 of RFC 4975 [RFC4975], complemented with the procedures described in this section. The MSRP switch MUST act as an MSRP endpoint receiver of the request, according to Section 5.3 of RFC 4975 [RFC4975].

If the MSRP switch receives an MSRP SEND request that does not contain a Message/CPIM wrapper, the MSRP switch MUST reject the request with a 415 response (specified in RFC 4975 [RFC4975]).

If the MSRP switch receives an MSRP SEND request where the URI included in the From header field of the Message/CPIM wrapper is not valid, (e.g., because it does not "belong" to the sender of the message or is not a valid participant of the chat room), the MSRP

switch MUST reject the request with a 403 response. In cases without error, the MSRP switch MUST construct responses according to Section 7.2 of RFC 4975 [RFC4975].

When the MSRP switch forwards a SEND request, it MAY use any report model in the copies intended for the recipients. The receiver reports from the recipients MUST NOT be forwarded to the originator of the original SEND request. This could lead to having the sender receiving multiple reports for a single MSRP request.

6.4. Congestion Avoidance

Congestion can occur when multiple heterogeneous interfaces are used by a number of users who are participating in a chat room, and, in particular, when paths become overloaded by any application. Some of these users might have fast paths capable of high throughputs while other users might be slow paths with constrained throughputs. Some paths might become congested only by the chat application; other paths gets congested by other applications. Therefore, it is possible that a subset of the participants of the chat room are able to send and receive a large number of messages in a short time or with large contents (e.g., pictures), whereas others are not able to keep up the pace.

Additionally, since MSRP uses a connection-oriented transport protocol such as TCP, it is expected that TCP congestion control mechanisms be activated if congestion occurs. Details on congestion control are specified in RFC 5681 [RFC5681].

While this document does not mandate a particular MSRP-specific mechanism to avoid congestion in any of the paths, something that is deemed outside the scope of this document, this document provides some recommendations for implementors to consider.

It is RECOMMENDED that MSRP switches implement one or more MSRP-specific strategies to detect and avoid congestion. Possible strategies (but definitely not a comprehensive list) include:

- If the MSRP switch is writing data to a send buffer and detects that the send buffer associated with that TCP connection is getting full (e.g., close to 80% of its capacity), the MSRP switch marks the associated MSRP sessions making use of that TCP connection as "congested".

- Prior to sending a new MSRP message to a user, the MSRP switch verifies the congested flag associated to that MSRP session. If the MSRP session is marked as congested, the MSRP switch can apply a congestion avoidance mechanism, such as:

    * The MSRP switch MAY discard regular MSRP messages sent to that user while the congestion flag is raised for the user's TCP connection. In order to inform the user of the congestion, the MSRP switch MAY send a regular MSRP message to the user whose congestion flag is raised. This message indicates that some other messages are being discarded due to network congestion. However, it should be noted that this message can get stuck at MSRP switch, if the path is fully congested, i.e., it may not be delivered anyhow.

    * The MSRP can implement a temporary policy to disallow the distribution of messages larger than a certain size to MSRP sessions marked as congested. Similarly, the user should be informed of this fact by the MSRP switch sending a regular MSRP message indicating this condition.

- If the MSRP switch determines that the congestion flag associated with a given TCP connection has been raised for quite some time (on the order of a few minutes), or if the interface is down, this may be considered an indication that the TCP connection has not been able to recover from a congestion state. The MSRP switch MAY close this congested TCP connection as well as the MSRP session and SIP session.

7. Nicknames

A common characteristic of existing chat room services is that participants have the ability to present themselves with a nickname to the rest of the participants of the chat room. It is used for easy reference of participants in the chat room and can also provide anonymous participants with a meaningful descriptive name.

A nickname is a useful construct in many use cases, of which MSRP chat is but one example. A nickname is associated with a URI; the focus knows the participant by its association to this URI. Therefore, if a user joins the chat room under the same URI from multiple devices, he or she may request the same nickname across all these devices.

A nickname is a user-selectable moniker by which the participant wants to be known to the other participants. It is not a 'display-name', but it is used somewhat like a display name. A main difference is that a nickname is unique inside a chat room to allow an unambiguous reference to a participant in the chat. Nicknames may be long lived, or they may be temporary. Users also need to reserve a nickname prior to its utilization.

This memo specifies the nickname as a string. The nickname string MUST unambiguously be associated with a single user in the scope of the chat room (conference instance). This scope is similar to having a nickname unique per user inside a chat room from "Extensible Messaging and Presence Protocol (XMPP): Core" [RFC6120]. The chat room may have policies associated with nicknames. It may not accept nickname strings at all, or it may provide a wider unambiguous scope like a domain or server, similar to IRC [RFC2810].

7.1. Using Nicknames within a Chat Room

This memo provides a mechanism to reserve a nickname for a participant for as long as the participant is logged into the chat room. The mechanism is based on a NICKNAME MSRP method (see below) and a new "Use-Nickname" header. Note that other mechanisms may exist (for example, a web page reservation system), although they are outside the scope of this document.

A chat room participant who has established an MSRP session with the MSRP switch, where the MSRP switch has indicated the support and availability of nicknames with the 'nicknames' token in the 'chatroom' SDP attribute, MAY send a NICKNAME request to the MSRP switch. The NICKNAME request MUST include a new Use-Nickname header that contains the nickname string that the participant wants to reserve. This nickname string MUST NOT be zero octets in length and MUST NOT be more than 1023 octets in length. Finally, MSRP NICKNAME requests MUST NOT include Success-Report or Failure-Report header fields.

> Bear in mind that nickname strings, like the rest of the MSRP message, use the UTF-8 transformation format [RFC3629]. Therefore, a character may be encoded in more than one octet.

An MSRP switch that receives a NICKNAME request containing a Use-Nickname header field SHOULD first verify whether the policy of the chat room allows the nickname functionality. If not allowed, the MSRP switch MUST reject the request with a 403 response, as per RFC 4975 [RFC4975].

If the policy of the chat room allows the usage of nicknames, any new nickname requested MUST be prepared and compared with nicknames already in use or reserved following the rules defined in "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Nicknames" [RFC7700].

This mitigates the problem of nickname duplication, but it does not solve a problem whereby users can choose similar (but different) characters to represent two different nicknames. For example, "BOY"

and "B0Y" are different nicknames that can mislead users. The former uses the capital letter "O" while the latter uses the number zero "0". In many fonts, the letter "O" and the number zero "0" might be quite similar and difficult to perceive as different characters. Chat rooms MAY provide a mechanism to mitigate confusable nicknames.

In addition to preparing and comparing following the rules above, the MSRP switch SHOULD only allow the reservation of an already-used nickname if the same user (e.g., identified by the SIP AOR) that is currently using the nickname is making this subsequent request. This may include, e.g., allowing the participant's URI to use the same nickname when the participant has joined the chat room from different devices under the same URI. The participant's authenticated identifier can be derived after a successful SIP Digest Authentication [RFC3261], included in a trusted SIP P-Asserted-Identity header field [RFC3325], included in a valid SIP Identity header field [RFC4474], or derived from any other present or future SIP authentication mechanism. Once the MSRP switch has validated that the participant is entitled to reserve the requested nickname, the MSRP switch verifies if the suggested nickname can be accepted (see below).

The reservation of a nickname can fail in several cases. If the NICKNAME request contains a malformed value in the Use-Nickname header field, the MSRP switch MUST answer the NICKNAME request with a 424 response code. This can be the case when the value of the Use-Nickname header field does not conform to the syntax.

The reservation of a nickname can also fail if the value of the Use-Nickname header field of the NICKNAME request is a reserved word (not to be used as a nickname by any user) or that particular value is already in use by another user. In these cases, the MSRP switch MUST answer the NICKNAME request with a 425 response code.

In both error conditions (receiving a 424 or 425 response code), the nickname usage is considered failed; the nickname is not allocated to this user. The user can select a different nickname and retry another NICKNAME request.

If the MSRP switch is able to accept the suggested nickname to be used by this user, the MSRP switch MUST answer the NICKNAME request with a 200 response as per regular MSRP procedures.

As indicated earlier, this specification defines a new MSRP header field: Use-Nickname. The Use-Nickname header field carries a nickname string. This specification defines the usage of the Use-Nickname header field in NICKNAME requests. If need arises, usages of the Use-Nickname header field in other MSRP methods should be specified separately.

According to RFC 4975 [RFC4975], MSRP uses the UTF-8 transformation format [RFC3629]. The syntax of the MSRP NICKNAME method and the Use-Nickname header field is built upon the MSRP formal syntax [RFC4975] using the Augmented Backus-Naur Form (ABNF) [RFC5234].

```
other-method =/ NICKNAMEm
         ; other-method defined in RFC 4975
NICKNAMEm = %x4E.49.43.4B.4E.41.4D.45 ; NICKNAME in caps
ext-header =/ Use-Nickname
         ; ext-header defined in RFC 4975
Use-Nickname = "Use-Nickname:" SP nickname
nickname = DQUOTE 1*1023(qdtext / qd-esc) DQUOTE
         ; qdtext and qd-esc defined in RFC 4975
```

Note that, according to RFC 4975 [RFC4975], "quoted-string" admits a subset of UTF-8 characters [RFC3629]. Please refer to Section 9 of RFC 4975 [RFC4975] for more details.

Once the MSRP switch has reserved a nickname and has bound it to a URI (e.g., a SIP AoR), the MSRP server MAY allow the usage of the same nickname by the same user (identified by the same URI, such as a SIP AoR) over a second MSRP session. This might be the case if the user joins the same chat room from a different SIP UA. In this case, the user MAY request a nickname that is the same or different than that used in conjunction with the first MSRP session; the MSRP server MAY accept the usage of the same nickname by the same user. The MSRP switch MUST NOT automatically assign the same nickname to more than one MSRP session established from the same URI, because this can create confusion to the user as whether the same nickname is bound to the second MSRP session.

7.2. Modifying a Nickname

Typically, a participant will reserve a nickname as soon as the participant joins the chat room. But it is also possible for a participant to modify his/her own nickname and replace it with a new one at any time during the duration of the MSRP session. Modification of the nickname is not different from the initial reservation and usage of a nickname; thus, the NICKNAME method is used as described in Section 7.1.

If a NICKNAME request that attempts to modify the current nickname of the user fails for some reason, the current nickname stays in effect. A new nickname comes into effect and the old one is released only after a NICKNAME request is accepted with a 200 response.

7.3. Removing a Nickname

If the participant no longer wants to be known by a nickname in the chat room, the participant can follow the method described in Section 7.2. The nickname element of the Use-Nickname header MUST be set to an empty quoted string.

7.4. Nicknames in Conference Event Packages

Typically the conference focus acts as a notifier of the conference event package, RFC 4575 [RFC4575]. It is RECOMMENDED that conference foci and endpoints support RFC 6502 [RFC6502] for providing information regarding the conference and, in particular, supplying information of the roster of the conference. It is also RECOMMENDED that conference foci and endpoints support RFC 6501 [RFC6501], which extends the <user> element originally specified in RFC 4575 [RFC4575] with a new 'nickname' attribute. This allows endpoints to learn the nicknames of participants of the chat room.

8. The SDP 'chatroom' Attribute

There are a handful of use cases where a participant would like to learn the chat room capabilities supported by the local policy of the MSRP switch and the chat room. For example, a participant would like to learn if the MSRP switch supports private messaging; otherwise, the participant may send what he believes is a private IM addressed to a participant, but since the MSRP switch does not support the functions specified in this memo, the message would eventually be distributed to all the participants of the chat room.

The reverse case also exists. A participant, say Alice, whose UA does not support the extensions defined by this document joins the chat room. The MSRP switch learns that Alice's application does not support private messaging nor nicknames. If another participant, say Bob, sends a private message to Alice, the MSRP switch does not distribute it to Alice, because Alice is not able to differentiate it from a regular message sent to the whole roster. Furthermore, if Alice replied to this message, she would do it to the whole roster. Because of this, the MSRP switch also keeps track of users who do not support the extensions defined in this document.

In another scenario, the policy of a chat room may indicate that certain functions are not allowed. For example, the policy may indicate that nicknames or private messages are forbidden.

In order to provide the user with a good chat room experience, we define a new 'chatroom' SDP attribute. The 'chatroom' attribute is a media-level value attribute [RFC4566] that MAY be included in conjunction with an MSRP media stream (i.e., when an "m=" line in SDP indicates "TCP/MSRP" or "TCP/TLS/MSRP"). The 'chatroom' attribute without further modifiers (e.g., chat-tokens) indicates that the endpoint supports the procedures described in this document for transferring MSRP messages to/from a chat room. The 'chatroom' attribute can be complemented with additional modifiers that further indicate the intersection of support and local policy allowance for a number of functions specified in this document. Specifically, we provide the means to indicate support for the use of nicknames and private messaging.

The 'chatroom' attribute merely indicates the capabilities supported and allowed by the local policy. This attribute is not a negotiation subject to the SDP offer/answer model [RFC3264], but instead a declaration. Therefore, a 'chatroom' attribute included in an SDP answer does not need to be a subset of the values included in the 'chatroom' attribute of its corresponding SDP offer. Consequently, an SDP answer MAY contain a 'chatroom' attribute even if its corresponding SDP offer did not include it.

In subsequent SDP offer/answer [RFC3264] exchanges pertaining to the same session, the 'chatroom' attribute MAY be modified with respect to an earlier SDP offer/answer exchange. The new value of this attribute indicates the current support and local policy, meaning that some restrictions can apply now or might have been removed. If the 'chatroom' attribute is not included in a subsequent SDP offer/answer, but a corresponding MSRP stream is still in place, it indicates that support for the procedures indicated in this document are disabled.

The ’chatroom’ SDP attribute has the following ABNF [RFC5234] syntax:

```
attribute         =/ chatroom-attr
                         ; attribute defined in RFC 4566
chatroom-attr     = chatroom-label [":" chat-token
                    *(SP chat-token)]
chatroom-label    = "chatroom"
chat-token        = (nicknames-token / private-msg-token /
                     ext-token)
nicknames-token   = "nickname"
private-msg-token = "private-messages"
ext-token         = private-token / standard-token
private-token     = toplabel "." *(domainlabel ".") token
                         ; toplabel defined in RFC 3261
                         ; domainlabel defined in RFC 3261
                         ; token defined in RFC 3261
standard-token    = token
```

A given ’chat-token’ value MUST NOT appear more than once in a ’chatroom’ attribute.

A conference focus that includes the ’nicknames’ token in the session description is signaling that the MSRP switch supports and the chat room allows the use of the procedures specified in Section 7. A conference focus that includes the ’private-messages’ in the SDP description is signaling that the MSRP switch supports and the chat room allows the use of the procedures specified in Section 6.2.

An example of the ’chatroom’ attribute for an MSRP media stream that indicates the acceptance of nicknames and private messages:

```
a=chatroom:nickname private-messages
```

An example of a ’chatroom’ attribute for an MSRP media stream where the endpoint, e.g., an MSRP switch, does not allow nicknames or private messages.

```
a=chatroom
```

The ’chatroom’ attribute allows extensibility with the addition of new tokens. No IANA registry is provided at this time, since no extensions are expected at the time of this writing. Extensions to the ’chatroom’ attribute can be defined in IETF documents or as private-vendor extensions.

Extensions defined in an IETF document MUST follow the 'standard-token' ABNF previously defined. In this type of extension, care must be taken in the selection of the token to avoid a clash with any of the tokens previously defined.

Private extensions MUST follow the 'private-token' ABNF previously defined. The 'private-token' MUST be included in the DNS name of the vendor. Then, the token is reversed in order to avoid clashes of tokens. The following is an example of an extension named "foo.chat" by a vendor "example.com"

```
a=chatroom:nickname private-messages com.example.chat.foo
```

Note that feature names created by different organizations are not intended to have the same semantics or even interoperate.

9. Examples

9.1. Joining a Chat Room

Figure 3 presents a flow diagram where Alice joins a chat room by sending an INVITE request. This INVITE request contains a session description that includes the chat room extensions defined in this document.

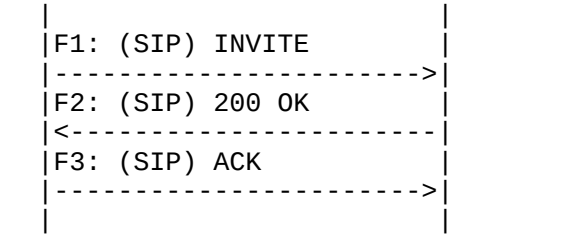

Figure 3: Flow Diagram of a User Joining a Chat Room

F1: Alice constructs an SDP description that includes an MSRP media stream. She also indicates her support for the chat room extensions defined in this document. She sends the INVITE request to the chat room server.

```
INVITE sip:chatroom22@chat.example.com SIP/2.0
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Chatroom 22 <sip:chatroom22@chat.example.com>
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:alice@client.atlanta.example.com;transport=tcp>
Content-Type: application/sdp
Content-Length: 290

v=0
o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com
s=-
c=IN IP4 client.atlanta.example.com
m=message 7654 TCP/MSRP *
a=accept-types:message/cpim text/plain text/html
a=path:msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
a=chatroom:nickname private-messages
```

F2: The chat room server accepts the session establishment. It includes the 'isfocus' and other relevant feature tags in the Contact header field of the response. The chat room server also builds an SDP answer that forces the reception of messages wrapped in Message/CPIM wrappers. It also includes the 'chatroom' attribute with the allowed extensions.

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
 ;received=192.0.2.101
From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
To: Chatroom 22 <sip:chatroom22@chat.example.com>;tag=8321234356
Call-ID: 3848276298220188511@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:chatroom22@chat.example.com;transport=tcp> \
          ;methods="INVITE,BYE,OPTIONS,ACK,CANCEL,SUBSCRIBE,NOTIFY" \
          ;automata;isfocus;message;event="conference"
Content-Type: application/sdp
Content-Length: 290
```

```
v=0
o=chat 2890844527 2890844527 IN IP4 chat.example.com
s=-
c=IN IP4 chat.example.com
m=message 12763 TCP/MSRP *
a=accept-types:message/cpim
a=accept-wrapped-types:text/plain text/html *
a=path:msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
a=chatroom:nickname private-messages
```

F3: The session established is acknowledged (details not shown).

9.2. Setting Up a Nickname

Figure 4 shows an example of Alice setting up a nickname using the chat room as provider. Her first proposal is not accepted because that proposed nickname is already in use. Then, she makes a second proposal with a new nickname. This second proposal is accepted.

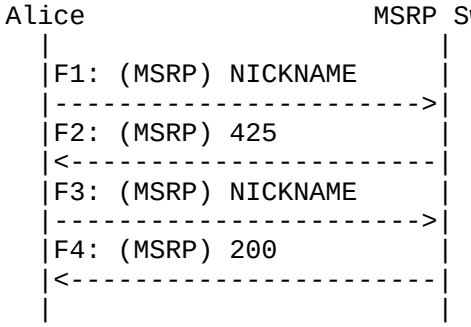

Figure 4: Flow Diagram of a User Setting up Her Nickname

F1: Alice sends an MSRP NICKNAME request that contains her proposed nicknames in the Use-Nickname header field.

```
MSRP d93kswow NICKNAME
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Use-Nickname: "Alice the great"
-------d93kswow$
```

F2: The MSRP switch analyzes the existing allocation of nicknames and detects that the nickname "Alice the great" is already provided to another participant in the chat room. The MSRP switch answers with a 425 response.

```
MSRP d93kswow 425 Nickname reserved or already in use
To-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
From-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
-------d93kswow$
```

F3: Alice receives the response. She proposes a new nickname in a second NICKNAME request.

```
MSRP 09swk2d NICKNAME
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Use-Nickname: "Alice in Wonderland"
-------09swk2d$
```

F4: The MSRP switch accepts the nickname proposal and answers with a 200 response.

```
MSRP 09swk2d 200 OK
To-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
From-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
-------09swk2d$
```

9.3. Sending a Regular Message to the Chat Room

Figure 5 is a flow diagram where Alice is sending a regular message addressed to the chat room. The MSRP switch distributes the message to the rest of the participants.

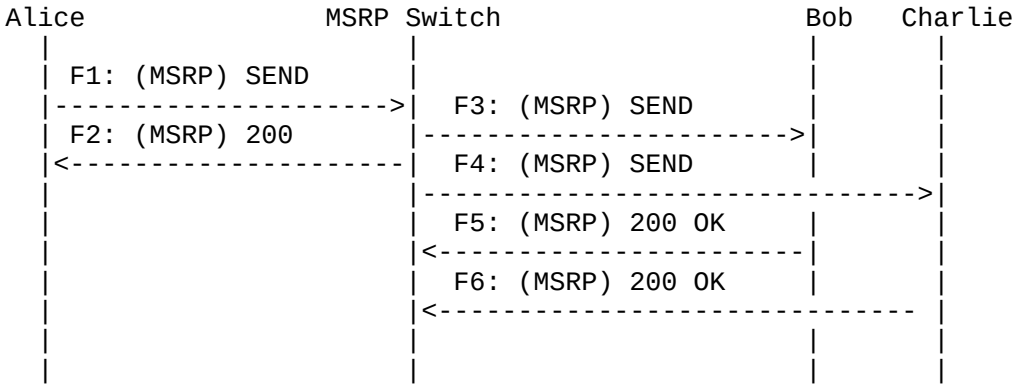

Figure 5: Sending a Regular Message to the Chat Room

F1: Alice builds a text message and wraps it in a Message/CPIM wrapper. She addresses the message to the chat room. She encloses the resulting Message/CPIM wrapper in an MSRP SEND request and sends it to the MSRP switch via the existing TCP connection.

```
MSRP 3490visdm SEND
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Message-ID: 99s9s2
Byte-Range: 1-*/*
Content-Type: message/cpim

To: <sip:chatroom22@chat.example.com;transport=tcp>
From: <sip:alice@atlanta.example.com>
DateTime: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

Hello guys, how are you today?
-------3490visdm$
```

F2: The MSRP switch acknowledges the reception of the SEND request with a 200 (OK) response.

```
MSRP 3490visdm 200 OK
To-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
From-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
Message-ID: 99s9s2
-------3490visdm$
```

F3: The MSRP switch creates a new MSRP SEND request that contains the received Message/CPIM wrapper and sends it to Bob.

```
MSRP 490ej23 SEND
To-Path: msrp://client.biloxi.example.com:4923/49dufdje2;tcp
From-Path: msrp://chat.example.com:5678/jofofo3;tcp
Message-ID: 304sse2
Byte-Range: 1-*/*
Content-Type: message/cpim

To: <sip:chatroom22@chat.example.com;transport=tcp>
From: <sip:alice@atlanta.example.com>
DateTime: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

Hello guys, how are you today?
-------490ej23$
```

Since the received message is addressed to the chat room URI in the From header of the Message/CPIM header, Bob knows that this is a regular message distributed to all participants in the chat room rather than a private message addressed to him.

The rest of the message flows are analogous to the previous. They are not shown here.

9.4. Sending a Private Message to a Participant

Figure 6 is a flow diagram where Alice is sending a private message addressed to Bob's SIP AOR. The MSRP switch distributes the message only to Bob.

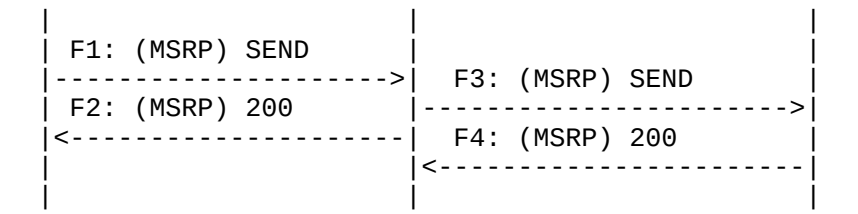

Figure 6: Sending a Private Message to Bob

F1: Alice builds a text message and wraps it in a Message/CPIM wrapper. She addresses the message to Bob's URI, which she learned from a notification in the conference event package. She encloses the resulting Message/CPIM wrapper in an MSRP SEND request and sends it to the MSRP switch via the existing TCP connection.

```
MSRP 6959ssdf SEND
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Message-ID: okj3kw
Byte-Range: 1-*/*
Content-Type: message/cpim

To: <sip:bob@example.com>
From: <sip:alice@example.com>
DateTime: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

Hello Bob.
-------6959ssdf$
```

F2: The MSRP switch acknowledges the reception of the SEND request with a 200 (OK) response.

```
MSRP 6959ssdfm 200 OK
To-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
From-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
Message-ID: okj3kw
-------6959ssdfm$
```

F3: The MSRP switch creates a new MSRP SEND request that contains the received Message/CPIM wrapper and sends it only to Bob. Bob can distinguish the sender in the From header of the Message/CPIM wrapper. He also identifies this as a private message due to the presence of his own SIP AOR in the To header field of the Message/CPIM wrapper.

```
MSRP 9v9s2 SEND
To-Path: msrp://client.biloxi.example.com:4923/49dufdje2;tcp
From-Path: msrp://chat.example.com:5678/jofofo3;tcp
Message-ID: d9fghe982
Byte-Range: 1-*/*
Content-Type: message/cpim

To: <sip:bob@example.com>
From: <sip:alice@atlanta.example.com>
DateTime: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

Hello Bob.
-------9v9s2$
```

F4: Bob acknowledges the reception of the SEND request with a 200 (OK) response.

```
MSRP 9v9s2 200 OK
To-Path: msrp://chat.example.com:5678/jofofo3;tcp
From-Path: msrp://client.biloxi.example.com:4923/49dufdje2;tcp
Message-ID: d9fghe982
-------9v9s2$
```

9.5. Chunked Private Message

The MSRP message below is a depiction of the same private message described in Section 9.4, but now the message is split in two chunks. The MSRP switch must wait for the complete set of Message/CPIM headers before distributing the messages.

```
MSRP 7443ruls SEND
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Message-ID: aft4to
Byte-Range: 1-*/174
Content-Type: message/cpim

To: <sip:bob@example.com>
From: <sip:alice@example.com>
-------7443ruls$

MSRP 7443ruls SEND
To-Path: msrp://chat.example.com:12763/kjhd37s2s20w2a;tcp
From-Path: msrp://client.atlanta.example.com:7654/jshA7weztas;tcp
Message-ID: aft4to
Byte-Range: 68-174/174
Content-Type: message/cpim

DateTime: 2009-03-02T15:02:31-03:00
Content-Type: text/plain

Hello Bob
-------7443ruls$
```

9.6. Nickname in a Conference Information Document

Figure 7 is a depiction of an XML conference information document received in a SIP NOTIFY request as a notification to the XCON Conference Event Package, RFC 6502 [RFC6502]. The conference information document follows the XCON Data Model specified in RFC 6501 [RFC6501].

The conference information document of Figure 7 presents information of two users who are participating in the conference (see each of the <user> elements). Each participant is bound to a nickname, shown in the 'nickname' attribute of the <user> element.

NOTE: The purpose of Figure 7 is to show the user-to-nickname relationship. It is believed that the example is correct, according to RFC 6501 [RFC6501]. In case of contradictions between this specification and RFC 6501 [RFC6501], the latter has precedence.

```
<?xml version="1.0" encoding="UTF-8"?>
  <conference-info
   xmlns="urn:ietf:params:xml:ns:conference-info"
   xmlns:xcon="urn:ietf:params:xml:ns:xcon-conference-info"
   entity="sip:chatroom22@chat.example.com"
   state="full" version="1">
  <!--
    CONFERENCE INFO
  -->
   <conference-description>
     <subject>MSRP nickname example</subject>
    </conference-description>
  <!--
     CONFERENCE STATE
  -->
   <conference-state>
    <user-count>2</user-count>
   </conference-state>
  <!--
    USERS
  -->
   <users>
     <user entity="sip:bob@example.com"
           state="full"
           xcon:nickname="Dopey Donkey">
        <display-text>Bob Hoskins</display-text>
     </user>
  <!--
    USER
  -->
     <user entity="sip:alice@atlanta.example.com"
          state="full"
          xcon:nickname="Alice the great">
       <display-text>Alice Kay</display-text>
     </user>
   </users>

  </conference-info>
```

Figure 7: Nickname in a Conference Information Document

10. IANA Considerations

10.1. New MSRP Method

This specification defines a new MSRP method that has been added to the "Methods" subregistry of the "Message Session Relay Protocol (MSRP) Parameters" registry:

NICKNAME

See Section 7 for details.

10.2. New MSRP Header

This specification defines a new MSRP header that has been added to the "Header Fields" subregistry of the "Message Session Relay Protocol (MSRP) Parameters" registry:

Use-Nickname

See Section 7 for details.

10.3. New MSRP Status Codes

This specification defines four new MSRP status codes that have been added to the "Status Codes" subregistry of the "Message Session Relay Protocol (MSRP) parameters" registry.

The 404 status code indicates the failure to resolve the recipient’s URI in the To header field of the Message/CPIM wrapper in the SEND request, e.g., due to an unknown recipient. See Section 6.2 for details.

The 424 status code indicates a failure in allocating the requested nickname due to a malformed syntax in the Use-Nickname header field. See Section 7 for details.

The 425 status code indicates a failure in allocating the requested nickname because the requested nickname in the Use-Nickname header field is reserved or is already in use by another user. See Section 7 for details.

The 428 status code indicates that the recipient of a SEND request does not support private messages. See Section 6.2 for details.

Table 1 summarizes the IANA registration data with respect to new MSRP status codes:

Value	Description	Reference
404	Failure to resolve recipient's URI	RFC 7701
424	Malformed nickname	RFC 7701
425	Nickname reserved or already in use	RFC 7701
428	Private messages not supported	RFC 7701

Table 1: New Status Codes

10.4. New SDP Attribute

This specification defines a new media-level attribute in the "Session Description Protocol (SDP) Parameters" registry. The registration data is as follows:

Contact: Miguel Garcia <miguel.a.garcia@ericsson.com>

Phone: +34 91 339 1000

Attribute name: chatroom

Long-form attribute name: Chat Room

Type of attribute: media level only

This attribute is not subject to the charset attribute

Description: This attribute identifies support and local policy allowance for a number of chat room related functions

Specification: RFC 7701 (this document)

See Section 8 for details.

11. Security Considerations

This document proposes extensions to the Message Session Relay Protocol [RFC4975]. Therefore, the security considerations of that document apply to this document as well.

A chat room is, by its nature, a potential Denial-of-Service (DoS) accelerator as it takes a message from one entity and sends it to many. Implementers of both UAs and switches need to carefully consider the set of anti-DoS measures that are appropriate for this application, and switch implementations, in particular, ought to

include appropriate anti-DoS features. The details of what is appropriate will vary over time and will also depend on the specific needs of the implementation; thus, they cannot be specified here.

If the participant's SIP UA does not understand the "isfocus" feature tag [RFC3840], it will not know that it is connected to a conference instance. The participant might not be notified that its MSRP client will try to send messages having potential multiple recipients to the MSRP switch. If the participant's MSRP client does not support the extensions of this specification, it is unlikely that it will try to send a message using the Message/CPIM wrapper content type [RFC3862], and the MSRP switch will reject the request with a 415 response [RFC4975]. Still, if a participant's MSRP client does create a message with a valid Message/CPIM wrapper content type [RFC3862] having the To header set to the URI of the chat room and the From header set to the URI of which the participant that is known to the chat room, the participant might be unaware that the message can be forwarded to multiple recipients. Equally, if the To header is set to a valid URI of a recipient known to the chat room, the message can be forwarded as a private message without the participant knowing.

To mitigate these problems, when the chat room detects that a UA does not support the procedures of this document (i.e., when the SIP UA is not chat room aware), the MSRP switch SHOULD send a regular MSRP message indicating that the SIP UA is actually part of a chat room and that all the messages that the user sends correctly formatted will be distributed to a number of participants. Additionally, the MSRP switch SHOULD also send a regular MSRP text message including the list of participants in the chat room so that the user becomes aware of the roster.

If a participant wants to avoid security concerns on the path between himself and the MSRP switch (e.g., eavesdropping, faked packet injection, or packet corruption), the participant's UA can force the usage of MSRP over a TLS [RFC5246] transport connection. This is negotiated in the SDP offer/answer exchange as per the regular procedures of RFC 4975 [RFC4975]. This negotiation will result in both endpoints establishing a TLS [RFC5246] transport connection that is used to exchange MSRP messages. The MSRP switch may also have local policy that forces the usage of TLS transport for all MSRP sessions, something that is also negotiated in SDP as per the regular procedures of RFC 4975 [RFC4975].

Nicknames are used to show the appearance of the participants of the chat room. A successful takeover of a nickname from a participant might lead to private messages being sent to the wrong destination. The recipient's URI will be different from the URI associated with the original owner of the nickname, but the sender might not notice

this. To avoid takeovers, the MSRP switch MUST make sure that a nickname is unique inside a chat room. Also, the security consideration for any authenticated identity mechanisms used to validate the SIP AOR will apply to this document as well. The chat room has a policy that determines the time that a nickname is still reserved for its holder, once it is no longer being used. This allows, e.g., a user that accidentally loses its connectivity, to reconnect to the chat room and keep on using the same nickname. It depends on the policy of the chat room if a nickname that has been previously used by another participant of the chat room can be reserved or not.

Section 7.1 discusses the problem of similar but different nicknames (e.g., thanks to the use of similar characters), and chat rooms MAY provide a mechanism to mitigate confusable nicknames.

Recipients of IMs should be cautious with the rendering of content, which can be malicious in nature. This includes, but is not limited to, the reception of HTML and JavaScript scripts, executable code, phishing attempts, etc. Endpoints SHOULD always request permission from the user before executing one of these actions.

It must be noted that endpoints using a TLS client side certificate with real names in the certificates will not be anonymous to the MSRP switch to which they connect. While the name in the certificate might not be used by MSRP, the server will have a certificate with the actual name in it.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <http://www.rfc-editor.org/info/rfc3264>.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, DOI 10.17487/RFC3323, November 2002, <http://www.rfc-editor.org/info/rfc3323>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <http://www.rfc-editor.org/info/rfc3629>.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, DOI 10.17487/RFC3840, August 2004, <http://www.rfc-editor.org/info/rfc3840>.

[RFC3860] Peterson, J., "Common Profile for Instant Messaging (CPIM)", RFC 3860, DOI 10.17487/RFC3860, August 2004, <http://www.rfc-editor.org/info/rfc3860>.

[RFC3862] Klyne, G. and D. Atkins, "Common Presence and Instant Messaging (CPIM): Message Format", RFC 3862, DOI 10.17487/RFC3862, August 2004, <http://www.rfc-editor.org/info/rfc3862>.

[RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, DOI 10.17487/RFC4353, February 2006, <http://www.rfc-editor.org/info/rfc4353>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, Ed., "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, DOI 10.17487/RFC4575, August 2006, <http://www.rfc-editor.org/info/rfc4575>.

[RFC4975] Campbell, B., Ed., Mahy, R., Ed., and C. Jennings, Ed., "The Message Session Relay Protocol (MSRP)", RFC 4975, DOI 10.17487/RFC4975, September 2007, <http://www.rfc-editor.org/info/rfc4975>.

[RFC4976] Jennings, C., Mahy, R., and A. Roach, "Relay Extensions for the Message Sessions Relay Protocol (MSRP)", RFC 4976, DOI 10.17487/RFC4976, September 2007, <http://www.rfc-editor.org/info/rfc4976>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5239] Barnes, M., Boulton, C., and O. Levin, "A Framework for Centralized Conferencing", RFC 5239, DOI 10.17487/RFC5239, June 2008, <http://www.rfc-editor.org/info/rfc5239>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <http://www.rfc-editor.org/info/rfc5246>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <http://www.rfc-editor.org/info/rfc5681>.

[RFC6501] Novo, O., Camarillo, G., Morgan, D., and J. Urpalainen, "Conference Information Data Model for Centralized Conferencing (XCON)", RFC 6501, DOI 10.17487/RFC6501, March 2012, <http://www.rfc-editor.org/info/rfc6501>.

[RFC6502] Camarillo, G., Srinivasan, S., Even, R., and J. Urpalainen, "Conference Event Package Data Format Extension for Centralized Conferencing (XCON)", RFC 6502, DOI 10.17487/RFC6502, March 2012, <http://www.rfc-editor.org/info/rfc6502>.

[RFC7700] Saint-Andre, P., "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Nicknames", RFC 7700, DOI 10.17487/RFC7700, December 2015, <http://www.rfc-editor.org/info/rfc7700>.

12.2. Informative References

[RFC2810] Kalt, C., "Internet Relay Chat: Architecture", RFC 2810, DOI 10.17487/RFC2810, April 2000, <http://www.rfc-editor.org/info/rfc2810>.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, DOI 10.17487/RFC3325, November 2002, <http://www.rfc-editor.org/info/rfc3325>.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, DOI 10.17487/RFC3966, December 2004, <http://www.rfc-editor.org/info/rfc3966>.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, DOI 10.17487/RFC4474, August 2006, <http://www.rfc-editor.org/info/rfc4474>.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, March 2011, <http://www.rfc-editor.org/info/rfc6120>.

Acknowledgments

The authors want to thank Eva Leppanen, Adamu Haruna, Adam Roach, Matt Lepinski, Mary Barnes, Ben Campbell, Paul Kyzivat, Adrian Georgescu, Nancy Greene, Cullen Jennings, Flemming Andreasen, Suresh Krishnan, Christer Holmberg, Saul Ibarra, Enrico Marocco, Alexey Melnikov, Peter Saint-Andre, Stephen Farrell, and Martin Stiemerling for providing comments.

Contributors

This work would have never been possible without the fruitful discussions on the SIMPLE WG mailing list, especially with Brian Rosen (Neustar) and Paul Kyzivat (Huawei), who provided extensive review and improvements throughout the document.

Authors' Addresses

Aki Niemi

Email: aki.niemi@iki.fi

Miguel A. Garcia-Martin
Ericsson
Calle Via de los Poblados 13
Madrid, ES 28033
Spain

Email: miguel.a.garcia@ericsson.com

Geir A. Sandbakken
Cisco Systems
Philip Pedersensvei 1
1366 Lysaker
Norway

Email: geirsand@cisco.com