

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 22, 2014

P. Saint-Andre
Cisco Systems, Inc.
S. Loreto
E. Gavita
N. Hossain
Ericsson
September 18, 2013

Interworking between the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP): One-to-One Text Chat
[draft-ietf-stox-chat-01](#)

Abstract

This document defines a bidirectional protocol mapping for the exchange of instant messages in the context of a one-to-one chat session between a user of the Session Initiation Protocol (SIP) and a user of the Extensible Messaging and Presence Protocol (XMPP). Specifically for SIP text chat, this document specifies a mapping to the Message Session Relay Protocol (MSRP).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on March 22, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

Internet-Draft

SIP-XMPP Interworking: Chat

September 2013

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology	3
3.	XMPP to MSRP	4
4.	MSRP to XMPP	7
5.	Composing Events	10
6.	Delivery Reports	12
7.	IANA Considerations	13
8.	Security Considerations	13
9.	References	14
9.1.	Normative References	14
9.2.	Informative References	14
Appendix A.	Acknowledgements	15
	Authors' Addresses	15

1. Introduction

Both the Session Initiation Protocol [[RFC3261](#)] and the Extensible Messaging and Presence Protocol [[RFC6120](#)] can be used for the purpose of one-to-one text chat over the Internet. To ensure interworking between these technologies, it is important to define bidirectional protocol mappings.

The architectural assumptions underlying such protocol mappings are provided in [[I-D.ietf-stox-core](#)], including mapping of addresses and error conditions. This document specifies mappings for one-to-one text chat sessions (sometimes called "session-mode" messaging); in particular, this document specifies mappings between XMPP messages of type "chat" and the Message Session Relay Protocol [[RFC4975](#)]. Mappings for single instant messages and groupchat are provided in separate documents.

The approach taken here is to directly map syntax and semantics from one protocol to another. The mapping described herein depends on the protocols defined in the following specifications:

- o XMPP chat sessions using message stanzas of type "chat" are specified in [[RFC6121](#)].
- o SIP-based chat sessions using the SIP INVITE and SEND request types are specified in [[RFC4975](#)].

In SIMPLE, a chat session is formally negotiated just as any other session type is using SIP. By contrast, a one-to-one chat "session" in XMPP is an informal construct and is not formally negotiated: a user simply sends a message of type "chat" to a contact, the contact then replies to the message, and the sum total of such messages exchanged during a defined period of time is considered to be a chat session. To overcome the disparity between these approaches, a gateway that wishes to map between SIP and XMPP for one-to-one chat sessions needs to maintain some additional state, as described below.

The discussion venue for this document is the mailing list of the STOX WG; visit <https://www.ietf.org/mailman/listinfo/stox> for subscription information and discussion archives.

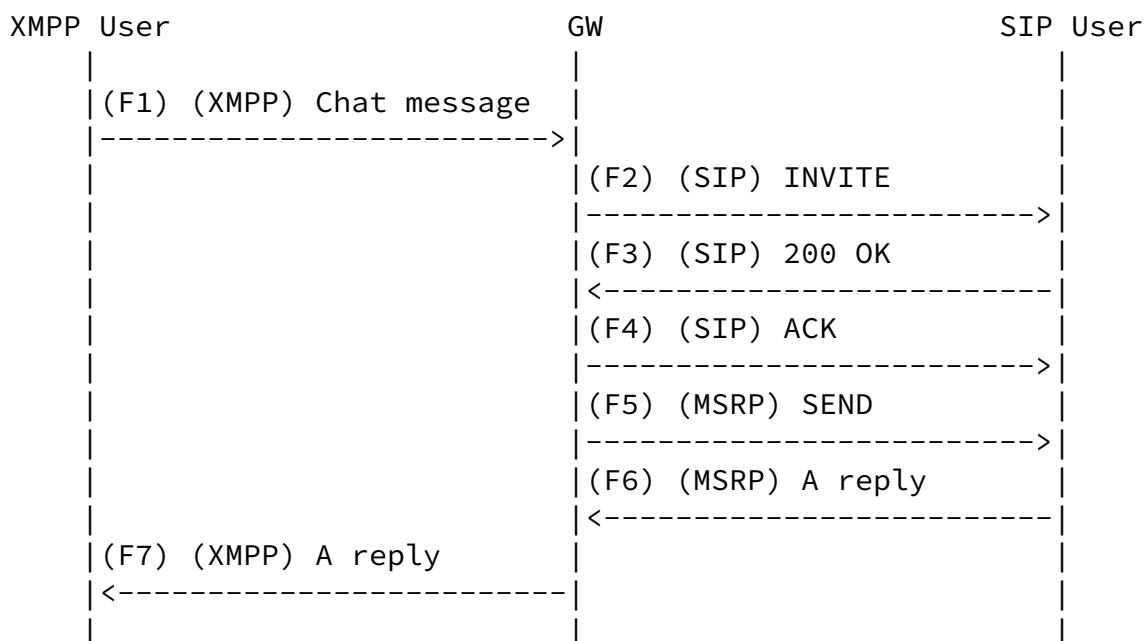
2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#).

3. XMPP to MSRP

In XMPP, the "informal session" approach is to simply send someone a <message/> of type "chat" without starting any session negotiation ahead of time (as described in [\[RFC6121\]](#)). The XMPP "informal session" approach maps very well into a SIP MESSAGE request, as described in [\[I-D.ietf-stox-core\]](#). However, the XMPP informal session approach can also be mapped to MSRP if the XMPP-to-SIP gateway maintains additional state.

The order of events is as follows.



```

| | |
| | |
| | |
| | (F8) (SIP) BYE
| | <----->
| | (F9) (SIP) 200 OK
| | ----->
| | |

```

First the XMPP user would generate an XMPP chat message.

Example: (F1) Juliet sends an XMPP message

```

| <message from='juliet@example.com/balcony'
|      to='romeo@example.net'
|      type='chat'>
|   <thread>711609sa</thread>
|   <body>Art thou not Romeo, and a Montague?</body>
| </message>

```

The local SIP-to-XMPP gateway at the SIMPLE server would then initiate an MSRP session with Romeo on Juliet's behalf (since there is no reliable way for the SIMPLE server to determine if Romeo's user agent supports MSRP, it simply needs to guess).

Example: (F2) Gateway starts a formal session on behalf of Juliet

```

| INVITE sip:romeo@example.net SIP/2.0
| To: <sip:romeo@example.net>
| From: <sip:juliet@example.com>
| Contact: <sip:juliet@example.com>;gr=balcony
| Subject: Open chat with Juliet?
| Call-ID: 711609sa

```

```
| Content-Type: application/sdp
|
| c=IN IP4 x2s.example.com
| m=message 7654 TCP/MSRP *
| a=accept-types:text/plain
| a=lang:en
| a=lang:it
| a=path:msrp://x2s.example.com:7654/jshA7weztas;tcp
```

Here we assume that Romeo accepts the MSRP session request.

Example: (F3) Romeo accepts the request

```
| SIP/2.0 200 OK
| To: <sip:juliet@example.com>;gr=balcony
| From: <sip:romeo@example.net>
| Contact: <sip:romeo@example.net>;gr=orchard
| Call-ID: 711609sa
| Content-Type: application/sdp
|
| c=IN IP4 s2x.example.net
| m=message 12763 TCP/MSRP *
| a=accept-types:text/plain
| a=lang:it
| a=path:msrp://s2x.example.net:12763/kjhd37s2s20w2a;tcp
```

The XMPP-to-SIP gateway then acknowledges the session acceptance on behalf of Romeo.

Example: (F4) Gateway sends ACK to Romeo's UA

```
| ACK sip:juliet@example.com SIP/2.0
| To: <sip:romeo@example.net>;gr=orchard
| From: <sip:juliet@example.com>
| Contact: <sip:juliet@example.com>;gr=balcony
| Call-ID: 711609sa
```

The XMPP-to-SIP gateway then transforms the original XMPP chat message into MSRP.

Example: (F5) Gateway transforms XMPP message to MSRP

```
| MSRP a786hjs2 SEND
| From-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
| To-Path: msrp://s2x.example.net:12763/kjhd37s2s20w2a;tcp
| Message-ID: 87652491
| Byte-Range: 1-25/25
| Content-Type: text/plain
|
| Art thou not Romeo, and a Montague?
| -----a786hjs2$
```

Romeo can then send a reply using his MSRP user agent.

Example: (F6) Romeo sends a reply

```
| MSRP a786hjs2 SEND
| To-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
| From-Path: msrp://s2x.example.net:12763/kjhd37s2s20w2a;tcp
| Message-ID: 87652491
| Byte-Range: 1-25/25
| Failure-Report: no
| Content-Type: text/plain
|
| Neither, fair saint, if either thee dislike.
| -----a786hjs2$
```

The SIP-to-XMPP gateway would then transform that message into appropriate XMPP syntax for routing to the intended recipient.

Example: (F7) Gateway transforms MSRP message to XMPP

```
| <message from='romeo@example.net/orchard'
|         to='juliet@example.com/balcony'
|         type='chat'>
|   <thread>711609sa</thread>
|   <body>Neither, fair saint, if either thee dislike.</body>
| </message>
```

When the MSRP user wishes to end the chat session, the user's MSRP client sends a SIP BYE.

Example: (F8) Romeo terminates the chat session

```
| BYE juliet@example.com sip: SIP/2.0
| Max-Forwards: 70
| From: <sip:romeo@example.net>;tag=087js
| To: <sip:juliet@example.com>;tag=786
| Call-ID: 711609sa
| Cseq: 1 BYE
| Content-Length: 0
```

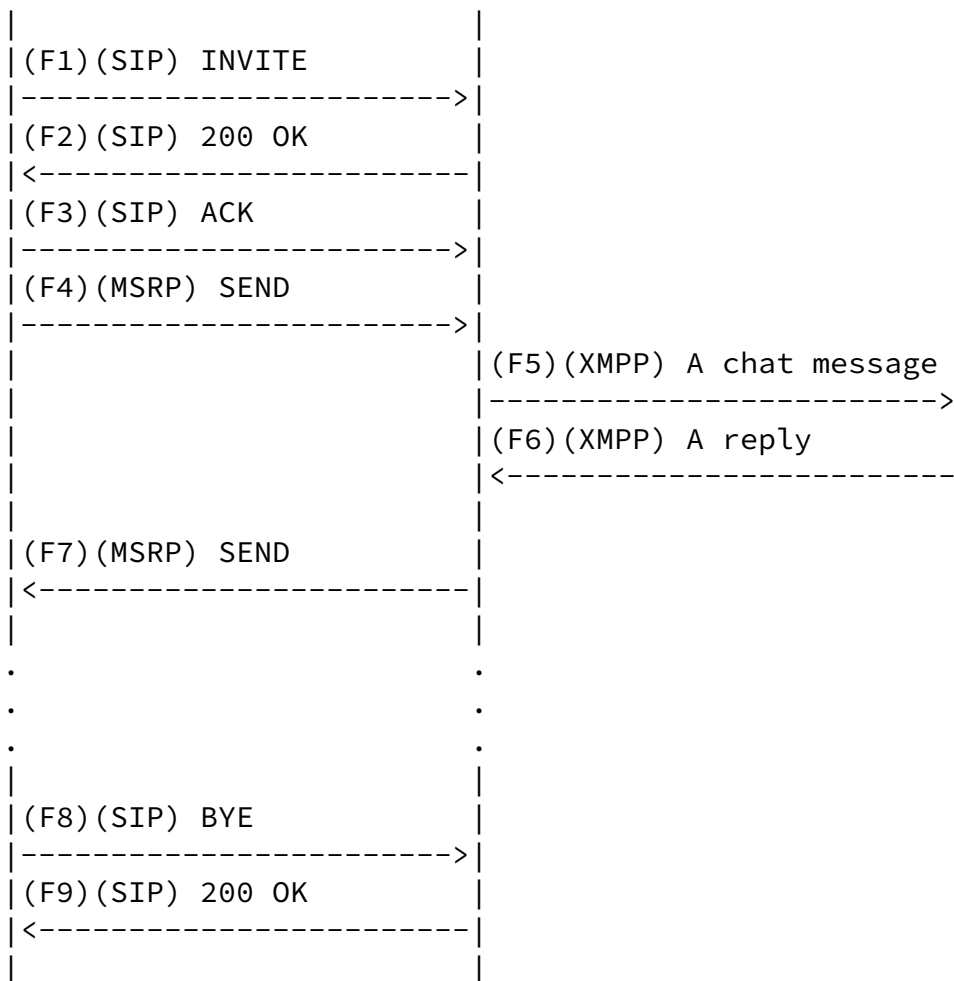
The BYE is then acknowledged by the XMPP-to-SIP gateway.

Example: (F9) Gateway acknowledges termination

```
| SIP/2.0 200 OK
| From: <sip:juliet@example.com>;tag=786
| To: <sip:romeo@example.net>;tag=087js
| Call-ID: 711609sa
| CSeq: 1 BYE
| Content-Length: 0
```

[4.](#) MSRP to XMPP

When an MSRP client sends messages through a gateway to an XMPP client that does not support formal sessions, the order of events is as follows.



Example: (F1) SIP user starts the session

```

| INVITE sip:juliet@example.com SIP/2.0
| To: <sip:juliet@example.com>
| From: <sip:romeo@example.net>
| Contact: <sip:romeo@example.net>;gr=orchard
| Subject: Open chat with Romeo?
| Call-ID: 742507no
| Content-Type: application/sdp
|
| c=IN IP4 s2x.example.net
| m=message 7313 TCP/MSRP *
| a=accept-types:text/plain
| a=lang:en
| a=lang:it
| a=path:msrp://s2x.example.net:7313/ansp71weztas;tcp

```

Example: (F2) Gateway accepts session on Juliet's behalf

```
| SIP/2.0 200 OK
| To: <sip:romeo@example.net>;gr=orchard
| From: <sip:juliet@example.com>
| Contact: <sip:juliet@example.com>;gr=balcony
| Call-ID: 742507no
| Content-Type: application/sdp
|
| c=IN IP4 x2s.example.com
| m=message 8763 TCP/MSRP *
| a=accept-types:text/plain
| a=lang:it
| a=path:msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
```

Example: (F3) Romeo sends ACK

```
| ACK sip:juliet@example.com SIP/2.0
| To: <sip:juliet@example.com>;gr=balcony
| From: <sip:romeo@example.net>
| Contact: <sip:romeo@example.net>;gr=orchard
| Call-ID: 742507no
```

Example: (F4) Romeo sends a message

```
| MSRP ad49kswow SEND
| To-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
| From-Path: msrp://s2x.example.net:7313/ansp71weztas;tcp
| Message-ID: 44921zaqwsx
| Byte-Range: 1-32/32
| Failure-Report: no
| Content-Type: text/plain
|
| I take thee at thy word ...
| -----ad49kswow$
```

Example: (F5) Romeo sends a message (XMPP translation)

```
| <message from='romeo@example.net'
|         to='juliet@example.com'
|         type='chat'>
|   <thread>742507no</thread>
|   <body>I take thee at thy word ...</body>
| </message>
```

Internet-Draft

SIP-XMPP Interworking: Chat

September 2013

Example: (F6) Juliet sends a reply

```
| <message from='juliet@example.com'  
|         to='romeo@example.net'  
|         type='chat'>  
|   <thread>711609sa</thread>  
|   <body>What man art thou ...?</body>  
| </message>
```

Example: (F8) Gateway transforms XMPP message to MSRP

```
| MSRP a786hjs2 SEND  
| To-Path: msrp://s2x.example.net:7313/jshA7weztas;tcp  
| From-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp  
| Message-ID: 87652491  
| Byte-Range: 1-25/25  
| Failure-Report: no  
| Content-Type: text/plain  
|  
| What man art thou ...?  
| -----a786hjs2$
```

Example: (F9) Romeo terminates the session

```
| BYE juliet@example.com sip: SIP/2.0  
| Max-Forwards: 70  
| To: <sip:juliet@example.com>;gr=balcony  
| From: <sip:romeo@example.net>  
| Contact: <sip:romeo@example.net>;gr=orchard  
| Call-ID: 742507no  
| Cseq: 1 BYE  
| Content-Length: 0
```

Example: (F10) Gateway acknowledges the termination of the session on behalf of XMPP user

```
| SIP/2.0 200 OK  
| To: <sip:juliet@example.com>;gr=balcony  
| From: <sip:romeo@example.net>
```

| Contact: <sip:romeo@example.net>;gr=orchard
| Call-ID: 742507no
| CSeq: 1 BYE

5. Composing Events

Both XMPP and MSRP enable a user agent to receive notifications when a person's conversation partner is composing an instant message

Saint-Andre, et al.

Expires March 22, 2014

[Page 10]

Internet-Draft

SIP-XMPP Interworking: Chat

September 2013

within the context of a chat session.

For XMPP, the Chat State Notifications specification [[XEP-0085](#)] defines five states: active, inactive, gone, composing, and paused. Some of these states are related to the act of message composition (composing, paused), whereas others are related to the sender's involvement with the chat session (active, inactive, gone).

For MSRP (and SIMPLE in general), the Indication of Message Composition for Instant Messaging specification [[RFC3994](#)] defines two states: idle and active. Here the idle state indicates that the sender is not actively composing a message, and the active state indicates that the sender is indeed actively composing a message (the sending user agent simply toggles between the two states, changing to active if the user is actively composing a message and changing to idle if the user is no longer actively composing a message).

Because the XEP-0085 states can represent information that is not captured in [RFC 3994](#), gateways can either (a) map only the composing-related states or (b) map all the XEP-0085 states.

The following mappings are suggested.

Table 1: Mapping of SIMPLE isComposing events to XMPP chat states

isComposing Event	Chat State
active	composing
idle	active

Table 2: Mapping of XMPP chat states to SIMPLE isComposing events

Chat State	isComposing Event
active	idle
inactive	idle
gone	[none, see note]
composing	active
paused	idle

Note: Although there is no mapping for the "gone" chat state, receipt of the "gone" state can be used as a trigger for terminating the formal chat session within MSRP.

6. Delivery Reports

Both XMPP and MSRP enable a user agent to receive notifications when a message has been received by the intended recipient.

For XMPP, the Message Receipts specification [[XEP-0184](#)] defines a method and XML namespace for requesting and returning indications that a message has been received by a client controlled by the intended recipient.

For MSRP, a native reporting feature is included, in the form of report chunks (see Sections [7.1.2](#) and [7.1.3](#) of [[RFC4975](#)]).

Examples follow.

First, the XMPP user sends a message containing a request for delivery notification.

Example: Juliet sends a message with a receipt request

```
| <message from='juliet@example.com'  
|     id='87652491'  
|     to='romeo@example.net'  
|     type='chat'>  
|   <thread>711609sa</thread>
```

```
| <body>What man art thou ...?</body>
| <request xmlns='urn:xmpp:receipts' />
| </message>
```

Example: Gateway transforms XMPP message to MSRP

```
| MSRP a786hjs2 SEND
| To-Path: msrp://s2x.example.net:7313/jshA7weztas;tcp
| From-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
| Message-ID: 87652491
| Byte-Range: 1-25/25
| Success-Report: yes
| Failure-Report: no
| Content-Type: text/plain
|
| What man art thou ...?
| -----a786hjs2$
```

Next, the recipient returns a report.

Example: Recipient returns receipt

```
| MSRP hx74g336 REPORT
| To-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
| From-Path: msrp://s2x.example.net:7313/jshA7weztas;tcp
| Message-ID: 87652491
| Byte-Range: 1-106/106
| Status: 000 200 OK
| -----hx74g336$
```

Example: Transformed message receipt

```
| <message from='romeo@example.net'
|         id='87652491'
|         to='juliet@example.com'>
|   <received xmlns='urn:xmpp:receipts' id='87652491' />
| </message>
```

[7.](#) IANA Considerations

This document requests no actions of IANA.

[8.](#) Security Considerations

Detailed security considerations for instant messaging protocols are given in [[RFC2779](#)], for SIP-based instant messaging in [[RFC3428](#)] (see also [[RFC3261](#)]), and for XMPP-based instant messaging in [[RFC6121](#)] (see also [[RFC6120](#)]).

This document specifies methods for exchanging instant messages through a gateway that translates between SIP and XMPP. Such a gateway MUST be compliant with the minimum security requirements of the instant messaging protocols for which it translates (i.e., SIP and XMPP). The addition of gateways to the security model of instant messaging specified in [[RFC2779](#)] introduces some new risks. In particular, end-to-end security properties (especially confidentiality and integrity) between instant messaging user agents that interface through a SIMPLE-XMPP gateway can be provided only if common formats are supported. Specification of those common formats is out of scope for this document, although it is recommended to use [[RFC3862](#)] for instant messages.

[9.](#) References

Saint-Andre, et al. Expires March 22, 2014 [Page 13]

Internet-Draft SIP-XMPP Interworking: Chat September 2013

[9.1.](#) Normative References

[I-D.ietf-stox-core]

Saint-Andre, P., Hourii, A., and J. Hildebrand,
"Interworking between the Session Initiation Protocol
(SIP) and the Extensible Messaging and Presence Protocol
(XMPP): Core", [draft-ietf-stox-core-04](#) (work in progress),
July 2013.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.
- [RFC3862] Klyne, G. and D. Atkins, "Common Presence and Instant Messaging (CPIM): Message Format", [RFC 3862](#), August 2004.
- [RFC3994] Schulzrinne, H., "Indication of Message Composition for Instant Messaging", [RFC 3994](#), January 2005.
- [RFC4975] Campbell, B., Mahy, R., and C. Jennings, "The Message Session Relay Protocol (MSRP)", [RFC 4975](#), September 2007.
- [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", [RFC 6120](#), March 2011.
- [RFC6121] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", [RFC 6121](#), March 2011.
- [XEP-0085] Saint-Andre, P. and D. Smith, "Chat State Notifications", XSF XEP 0085, September 2009.
- [XEP-0184] Saint-Andre, P. and J. Hildebrand, "Message Delivery Receipts", XSF XEP 0184, March 2011.

[9.2.](#) Informative References

- [RFC2779] Day, M., Aggarwal, S., and J. Vincent, "Instant Messaging / Presence Protocol Requirements", [RFC 2779](#), February 2000.
- [RFC3428] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C.,

[Appendix A](#). Acknowledgements

Some text in this document was borrowed from [[I-D.ietf-stox-core](#)].

Thanks to Adrian Georgescu, Saul Ibarra, and Tory Patnoe for their feedback.

Authors' Addresses

Peter Saint-Andre
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Phone: +1-303-308-3282
Email: psaintan@cisco.com

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Salvatore.Loreto@ericsson.com

Eddy Gavita
Ericsson
Decarie Boulevard
Town of Mount Royal, Quebec
Canada

Email: eddy.gavita@ericsson.com

Nazin Hossain
Ericsson
Decarie Boulevard
Town of Mount Royal, Quebec
Canada

Email: Nazin.Hossain@ericsson.com

