

**Establishing jabber Messaging Sessions with the Session Initiation
Protocol
draft-sparks-simple-jabber-sessions-00**

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 27, 2003.

Copyright Notice

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

Abstract

This document explores modeling jabber message streams as media sessions, and how they can be initiated with the Session Initiation Protocol. It also explores how these sessions can be integrated into existing session-based multimedia communication applications.

Table of Contents

1.	An overview of session management using SIP	3
2.	Jabber messaging as a media session	3
3.	A example jabber chat session	4
4.	Acknowledgments	8
	References	8
	Author's Address	9
	Full Copyright Statement	10

1. An overview of session management using SIP

The Session Initiation Protocol [[1](#)] (SIP) is designed to manage media "sessions" between communicating parties. The types of media these sessions can use is unbounded and can range from simple audio to synchronized audio, video, and text. To establish such session, SIP first provides a rendezvous mechanism. Parties wishing to enter into a communication session do not need to know beforehand how to reach their intended peers' endpoints. Rather, they can send an invite to a session to the peer's address of record. SIP services will route that invitation to the appropriate endpoint(s) based on information the peer has provided and the characteristics of the invitation. In most cases, the invitation will contain an "offer" of a session (maybe several), describing the potential transports and encoding characteristics for the media streams within the session. The peer provides an "answer" accepting or declining each stream in the offer, and includes its contribution to the necessary transport information and characteristic tuning for those streams it accepts. ([\[2\]](#) discusses this offer/answer model in detail). At this point, the session is established and the parties begin communication over those streams. When one of the parties is finished with the session, they tell their peer "bye" using SIP, terminating the session.

2. Jabber messaging as a media session

A series of jabber "chat" or "groupchat" messages can be viewed as a session. This allows a communication user agent to offer a jabber messaging session as one of the possible media types in a session. For instance, Alice may want to communicate with Bob. Alice doesn't know at the moment whether Bob has his phone or his jabber client handy, but she's willing to use either. She sends an invitation to a session to the address of record she has for Bob, sip:bob@biloxi.com. Her invitation offers both voice and a jabber chat session. The SIP services at biloxi.com forward the invitation to Bob at his currently registered clients. Bob accepts the invitation at his jabber client and they begin a threaded chat conversation.

This session model allows jabber message sessions to be integrated into advanced communications applications with little to no additional protocol development. For example, during the above chat session, Bob decides Alice really needs to be talking to Carol. Bob can transfer Alice to Carol, introducing them into their own jabber messaging session. Jabber messaging sessions could be integrated into call-center and dispatch environments utilizing third-party call control and conferencing applications.

The rendezvous and characteristics negotiation features of SIP can also be used to introduce peers to jabber networks that have

Sparks

Expires April 27, 2003

[Page 3]

restricted server-to-server policies. A SIP invitation can contain an address and port of a jabber server for the peer to connect to. It can also securely provide credentials or other data to be consumed by that jabber server. Further, the invitation could provide material to be used to encrypt the bodies of the messages transported in the chat session being established. In the extreme, an endpoint could create an ephemeral stand-alone jabber server co-located with its client and use SIP to invite a peer to enter into a chat session at that server.

In other cases, the invitation would allow an endpoint's jabber client to use its usual "home" jabber server, establishing chat or groupchat sessions over an existing connection, if present. The session management capabilities of SIP can be used to add concrete start and stop times to those sessions and, as above, provide material that the endpoints can use to secure the session. For example, Alice might be in a phone conference with Bob and decides she needs a text sub-channel. She is currently connected to the public jabber cloud through atlanta.com and Bob is currently connected to the public cloud through biloxi.com. Alice and Bob don't know each other's jabber id. Alice can use SIP to invite Bob into a chat session, providing her jabber id, a thread to use, and maybe even a key to encrypt the messages with. Bob will provide his jabber id in the SIP answer. Armed with this information, each of their endpoints can associate the resulting chat session with the phone call.

Similarly, SIP can be used to invite peers to join ad-hoc or permanent jabber groupchat sessions.

3. A example jabber chat session

Figure 1 shows Romeo attempting to establish a session with Juliet. Romeo offers both voice and a jabber chat session. Juliet declines the voice session in favor of jabber chat, leading to the messaging dialog from section 4.5 of [3]. Caveat: This is a very rough first cut at what the session description (the SDP in the body of the messages) might look like, and must not be taken as a normative suggestion. As this problem is studied, a different form will almost certainly be decided on. The point of this example is to show, roughly, how the important information needed to set up such a session can be carried.

SIP Message 1:

Sparks

Expires April 27, 2003

[Page 4]

INVITE sip:juliet@capulet.com SIP/2.0
Via: SIP/2.0/TCP orchard.montague.net;branch=z9hG4bK2a9323
To: Juliet <sip:juliet@capulet.com>
From: Romeo <sip:romeo@montague.net>;tag=29923923
Call-ID: 2349098234@orchard.montague.net
CSeq: 13992 INVITE
Max-Forwards: 70
Contact: <sip:romeo@orchard.montague.net>
Content-Type: application/sdp
Content-Length: (appropriate value)

v=0
o=romeo 28908764872 28908764872 IN IP4 orchard.montague.net
s=-
c=IN IP4 orchard.montague.net
t=0 0
m=audio 39923 RTP/AVP 0
a=rtpmap:0 PCMU/8000
c=IN IP4 xmpp.montague.net
t=0 0
m=message 5222 xmpp/tcp chat
a=jid:romeo@mantague.net/orchard
a=thread:283461923759234

SIP Message 2 :

SIP/2.0 200 OK
Via: SIP/2.0/TCP orchard.montague.net;branch=z9hG4bK2a9323
To: Juliet <sip:juliet@capulet.com>;tag=9392312353
From: Romeo <sip:romeo@montague.net>;tag=29923923
Call-ID: 2349098234@orchard.montague.net
CSeq: 13992 INVITE
Max-Forwards: 70
Contact: <sip:juliet@balconey.capulet.com>
Content-Type: application/sdp
Content-Length: (appropriate value)

v=0
o=juliet 28908765142 28908765142 IN IP4 balcony.capulet.com
s=-
c=IN IP4 balcony.capulet.com
t=0 0
m=audio 0 RTP/AVP 0
a=rtpmap:0 PCMU/8000
c=IN IP4 xmpp.montague.net
t=0 0
m=message 5222 xmpp/tcp chat
a=jid:juliet@capulet.com/balcony

Sparks

Expires April 27, 2003

[Page 5]

SIP Message 3 :

```
ACK sip:juliet@balconey.capulet.com SIP/2.0
Via: SIP/2.0/TCP orchard.montague.net;branch=z9hG4bK239s934
To: Juliet <sip:juliet@capulet.com>;tag=9392312353
From: Romeo <sip:romeo@montague.net>;tag=29923923
Call-ID: 2349098234@orchard.montague.net
CSeq: 13992 ACK
Max-Forwards: 70
Content-Length:0
```

Figure 1: Establishing a chat session

If they aren't already connected to the public jabber network, Romeo and Juliet now establish jabber sessions with xmpp.montague.net at port 5222 and register with the service. They can then enter into the jabber chat session from section 4.5 of [\[3\]](#) also shown here in Figure 2.

```
<message
  to="romeo@montague.net/orchard"
  from="juliet@capulet.com/balcony"
  type="chat">
  <body>Art thou not Romeo, and a Montague?</body>
  <thread>283461923759234</thread>
</message>

<message
  to="juliet@capulet.com/balcony"
  from="romeo@montague.net/orchard"
  type="chat">
  <body>Neither, fair saint, if either thee dislike.</body>
  <thread>283461923759234</thread>
</message>

<message
  to="romeo@montague.net/orchard"
  from="juliet@capulet.com/balcony"
  type="chat">
  <body>How cam'st thou hither, tell me, and wherefore?</body>
  <thread>283461923759234</thread>
</message>
```

Figure 2: A jabber chat session

When they are done, Juliet terminates the session as shown in Figure 3.

SIP Message 4:

```
BYE sip:romeo@orchard.montague.net SIP/2.0
Via: SIP/2.0/TCP balconey.capulet.com;branch=z9hG4bK993231
From: Juliet <sip:juliet@capulet.com>;tag=9392312353
To: Romeo <sip:romeo@montague.net>;tag=29923923
Call-ID: 2349098234@orchard.montague.net
CSeq: 33903 BYE
Max-Forwards: 70
Content-Length:0
```

SIP Message 5:

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP balconey.capulet.com;branch=z9hG4bK993231
From: Juliet <sip:juliet@capulet.com>;tag=9392312353
To: Romeo <sip:romeo@montague.net>;tag=29923923
Call-ID: 2349098234@orchard.montague.net
CSeq: 33903 BYE
Max-Forwards: 70
Content-Length:0
```

Figure 3: Terminating a jabber chat session

4. Acknowledgments

The following have provided significant support to the creation of this document through detailed discussions and review:

Jonathan Rosenberg
Adam Roach
Ben Campbell

References

- [1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.
- [2] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with

Session Description Protocol (SDP)", [RFC 3264](#), June 2002.

- [3] Miller, J. and P. Saint-Andre, "XMPP Instant Messaging",
Internet-Draft <[draft-miller-xmpp-im-00](#)>, June 2002.

Author's Address

Robert J. Sparks
dynamicsoft
5100 Tennyson Parkway
Suite 1200
Plano, TX 75024

EMail: rsparks@dynamicsoft.com

Full Copyright Statement

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

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

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

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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

