Internet Engineering Task Force B. Campbell
Internet-Draft dynamicsoft
Expires: January 11, 2002 July 13, 2001

SIP Call Control - Framework
draft-ietf-sip-cc-framework-00

Status of this Memo

Copyright Notice

Abstract

This document proposes that SIP call control features be added in a modular fashion, using an open-ended framework of extensions instead of a single extension. This memo proposes a modular design philosophy for call control extensions, and lists current work-in-progress call control related drafts.

Table of Contents

1. Introduction

Most conventional telephony applications provide some level of support for modifying an in-progress call, or call control. Simple examples include call transfer and three way calling. More complex examples include conferencing and third party control.

The baseline SIP protocol[1]provides some limited support for call control, in that a call-leg participant can terminate the call leg, put it on hold, or modify the characteristics of its media stream.

However, many common call control applications require extensions to SIP in order to accomplish tasks such as referring a call to a new end point, or joining an existing call.

This memo proposes a modular approach to call control extension.

2. Changes from Previous Version

This revision has only minor changes from the previous version:

Removed open item concerning usage of the term "attended transfer."

Renamed file to reflect status as a SIP working group item.

Added references to the Call Control Model draft.[3]

Added a section listing Call Control drafts that are currently in process.

Removed discussion of original SIP call control draft.

Made minor editorial revisions to improve clarity.

3. Call Control Feature Examples

The following examples are call features for which extensions are currently under development, or may require extensions in the near future. These are examples only, and should not be considered authoritative; a formal treatment of call control features and terminology can be found in [3].

Transfer with Consultation Hold - The transferring party establishes a session with the transfer target before completing the transfer (Currently proposed in [4]).

Attended transfer - the transferring party establishes a session with the target and mixes both sessions together so that all three

parties can participate, then disconnects leaving the transferee and transfer target with an active session.

Conference Bridge - Callers join a conference on a centralized bridge.

Fully meshed conference - Callers establish sessions with all other callers on the conference. Each client mixes media streams.

Call Park - Call participant transfers a call to a call park, then retrieves it at a later time.

Call Pick - A party picks up a call that was ringing at another station.

Call Monitoring - A call center supervisor joins an in-progress call for monitoring purposes.

These examples are not exhaustive; we expect that more call control feature requirements will be proposed as SIP usage matures. Therefore it is not possible for this document to enumerate all call control extensions in advance.

4. A Modular Approach

We propose the SIP call control extensions be handled in a modular fashion. Instead of having a single unified call control extension, we should instead have a framework of extensions. Each of these extensions would focus on a bounded and coherent requirement (or extension) set.

A framework approach allows SIP entities to negotiate feature support with more granularity. For example, an implementation could assert that it supports call transfer without implying that it also supports conferencing.

5. Call Control Extension Design Philosophy

Each call control extension should address a coherent group of requirements that are most likely to be needed as a set. If implementers find themselves having to add features that would not normally be required by their application just because they are defined by the extension, it is probably to big.

The negotiated support of one call control extension MUST not imply the support of other extensions. While multiple extensions MAY share extended methods or headers, they MUST NOT do so unless the semantics are identical for all extensions.

Call Control extension designers SHOULD NOT overload existing methods and headers, unless the new function is actually a logical extension of the method or header in question.

> Overloaded headers and extension create complications for protocol implementations. For example, if an extension overloads INVITE by adding a new header, the implementation must check every INVITE for the presence of the header before taking action. If the implementation supports many extensions that each overload INVITE, the decision logic becomes complex.

Subject to the limitation on overloading methods and headers, extensions should be as simple as possible and reuse existing SIP related features whenever appropriate.

6. Extension Negotiation

Since call control actions could conceivably be initiated by any user agent, SIP entities MUST follow the guidelines concerning feature negotiation described in the draft,"Guidelines for the Authors of SIP Extensions"[2].

If a SIP entity receives a message containing a call control extension method or header that normally requires negotiation but has not been properly negotiated, it SHOULD behave as if it had no knowledge of the extension in question, regardless of whether the entity is capable of supporting it.

> It is tempting to suggest that if an entity recognized an un-negotiated extension, it should go ahead and act on it. However, it is dangerous for an entity to assume it understands the intent behind an extension without explicit negotiation. If two extensions were to use the same keyword for an extended feature with different semantics, the receiving entity would have no way to guess the intent of the sending entity.

7. Adding New Call Control Operations

Additional call control operations SHOULD be implemented as additional SIP extension methods. Each such extension method MUST progress through the standards process as per other IETF standards.

Such extensions SHOULD include motivations, requirements, specification of syntax and semantics, and detailed usage examples. Additionally, it SHOULD describe how to specifically apply the negotiation guidelines in [2].

8. Call Control Documents

Work is in progress on the following documents which fit into this framework:

"SIP Call Control - Model"[3]

"SIP Call Control - Transfer"[4]

9. Security Considerations

Each call control extension SHOULD describe mechanisms to prevent unauthorized parties to invoke the extensions. Any extension that allows entities not party to a call to invoke call control operations MUST describe said mechanisms.

10. Acknowledgments

The author thanks the following for their contribution to this work: Chris Cunningham, Steve Donovan, Alan Johnston, Robert Sparks, Kevin Summers, Dean Willis, and Rohan Mahy.

References

[1] Handley, M., Schulzrinne, H., Schooler, E. and J. Rosenberg, "SIP: session initiation protocol", RFC 2543, March 1999.

[2] Rosenberg, J. and H. Schulzrinne, "Guidelines for Authors of Extensions", draft-ietf-sip-guidelines-01.txt (work in progress), March 2000.

[3] Mahy, R. , "SIP Call Control Model", draft-mahy-sip-cc-models-00.txt (work in progress), March 2001.

[4] Sparks, R., "SIP Call Control - Transfer", draft-sip-cc-transfer-04.txt (work in progress), February 2001.

Author's Address

Ben Campbell
dynamicsoft
5100 Tennyson Parkway
Suite 1200
Plano, TX 75024

email: bcampbell@dynamicsoft.com

Full Copyright Statement

Acknowledgement

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