Internet Engineering Task Force SIP WG
Internet Draft
K. Chan
G. v. Bochmann
University of Ottawa
draft-ietf-sip-fneg-01.txt
May 31, 2003
Expires: December 2003

Feature Negotiation for the Session Initiation Protocol (SIP)

STATUS OF THIS MEMO

Abstract

This document describes a set of extensions to the Session Initiation Protocol (SIP) which allow features to negotiate their operational constraints in a session. Our proposal extends the caller

preferences
draft [CPref]. However, there are some key differences between our proposal and the caller preferences draft, and they are that 1) our proposal is intended for supporting a variety of feature negotiations which tend to involve more complex operational parameters than those in caller preferences, 2) our negotiation scheme takes the well-known feature interactions problem into account; it supports feature interaction resolution [KChan1], 3) our negotiation approach is more flexible and is based on the proposal-re-proposal and proposal-counter-proposal negotiation paradigm [NegFI], and 4) the proposal, re-proposal, or counter-proposal initiator of the negotiation needs

not be the caller. In accordance to the caller preferences draft [CPref], our extension does so by defining three new request header fields, Accept-Contact, Reject-Contact, and Request-Disposition, which specify the negotiation parties' operational parameters (constraints). The extension also defines new parameters for the Contact header field that describe the preconditions, post-conditions, and trigger events of the feature.

1. Introduction

In a distributed internet telphony system,

- constraints are associated to the current SIP transaction. A feature that spans across multiple SIP transaction would require to specify multiple constraints for each SIP transaction.

Table of Contents

1 Introduction

When a Session Initiation Protocol (SIP) [1] server receives a request, there are a number of decisions it can make regarding processing of the request. These include:

- o whether to proxy or redirect the request
- o which URIs to proxy or redirect to
- o whether to fork or not
- o whether to search recursively or not
- o whether to search in parallel or sequentially

The server can base these decisions on any local policy. This policy can be statically configured, or can be based on programmatic execution or database access.

However, the administrator of the server is the not the only entity with an interest in request processing. There are at least three parties which have an interest: (1) the administrator of the server, (2) the user that sent the request, and (3) the user to whom the request is directed. The directives of the administrator are embedded in the policy of the server. The preferences of the user to whom

the
request is directed (referred to as the callee, even though the request may not be INVITE) can be expressed most easily through a script written in some type of scripting language, such as the Call Processing Language (CPL) [22]. However, no mechanism exists to incorporate the preferences of the user that sent the request (also referred to as the caller, even though the request may not be INVITE). For example, the caller might want to speak to a specific user, but want to reach them only at work, because the call is a business call. As another example, the caller might want to reach a user, but not their voicemail, since it is important that the caller
talk to the called party. In both of these examples, the caller's preference amounts to having a proxy make a particular routing choice
based on the preferences of the caller.

This extension allows the caller to have these preferences met. It does so by specifying mechanisms by which a caller can provide preferences on processing of a request. There are two types of preferences. One of them, called request handling preferences, are encapsulated in the Request-Disposition header field. They provide specific request handling directives for a server. The other, called
feature preferences, are present in the Accept-Contact and Reject-Contact header fields. They allow the caller to provide a feature set