

SIP
Internet Draft
Expires: September 2, 2003

H. Khartabil
Nokia
March 3, 2003

**Congestion safety and Content Indirection
draft-khartabil-sip-congestionsafe-ci-02.txt**

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 23, 2003.

Copyright Notice

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

Abstract

The Content-Indirection service has many uses. This document describes this service by combining congestion safely and Content-Indirection Mechanism into the one document and presents scenarios where the 2 could be combined. It also introduces extensions to SIP that allows end points to specify their maximum acceptable message size.

Table of Contents

1.0 Terminology.....3

2.0 Introduction.....3

3.0 Overview of Operation.....4

4.0 UA Behaviour.....5

4.1 UAC Sending Requests.....5

4.2 UAC Behaviour when receiving Response.....5

4.2.1 413 or 513 Error Responses.....5

4.3 UA Registering Desired Maximum SIP Message Size.....6

4.4 UAS Receiving A SIP Message With Large Contents.....7

4.5 UAS Receiving a Request with Indirected Content.....8

5.0 Proxy Behaviour.....8

5.1 Congestion Safe Proxy.....8

5.2 Proxy Receiving A SIP Message With Large Content.....8

5.3 Proxy Refusing To Handle Large SIP Requests.....10

5.4 Proxy Performing Content Indirection.....10

5.5 Proxy Receiving 413 Response From Downstream.....11

5.6 Proxy Receiving a Request With Indirected Content.....11

6.0 Content Storage Server (CSS) Behaviour.....12

7.0 Syntax for Extensions.....12

7.1 Max-Size Header.....12

7.2 Max-size parameter.....12

8.0 Security considerations.....13

9.0 Examples.....13

9.1 UAC Posting Contents to CSS Before Sending SIP MESSAGE.....13

9.2 Receiving A MESSAGE With Large Contents after a REGISTER, Home
Proxy Performs Content-Indirection.....15

9.3 Receiving A NOTIFY With Large Contents, UAS Refuses To Handle
Request.....17

10.0 IANA Considerations.....19

11.0 Acknowledgements.....19

12.0 References.....19

Authors' Addresses.....19

Full Copyright Statement.....20

Acknowledgement.....20

1.0 Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC 2119](#) [5].

2.0 Introduction

The content-indirection service has gained a lot of support from the SIP community lately. There have been many scenarios where content-indirection service can be useful. This document discusses this service and describes how it can be deployed in a network. This document covers, in more detail, the scenarios where senders, intermediaries and receivers of SIP messages may want to make use of the content-indirection service.

A SIP UA originating a SIP request may not want to send SIP requests into the network, but instead chooses to publish a link to a URI where it has posted such content, using the mechanism described in [3]. This restriction is maybe due to charging, network configuration, or merely SIP message size constraints (the UA limiting its SIP transport to message oriented protocol such as UDP). Good examples of such SIP messages are MESSAGE [2] and PUBLISH [6].

The Session Initiation Protocol [1] allows the use of UDP for transport of SIP messages. The use of UDP with large payloads risks network congestion problems, as UDP itself does not define congestion prevention, detection, or correction mechanisms. Large SIP messages may cause UDP datagram fragmentation, something not desired by a network of SIP entities.

The root of the problem lies with the SIP proxies. SIP proxies are able to convert the transport protocol from reliable to unreliable on hop-by-hop basis, therefore end-to-end congestion-safe path for SIP messages cannot be guaranteed.

Conventional SIP payloads carry signalling information about media, but not media itself. There is potential that when a SIP infrastructure is shared between call signalling and instant messaging [2], the IM traffic will interfere with call signalling traffic. This also applies to other types of SIP Requests that have potential to carry large contents (NOTIFY is an example).

Furthermore, mobile terminals might want to limit the size of a SIP message received regardless of the underlying transport protocol (e.g. TCP). This may be due to memory limitations or the fact that the radio link that the terminal is accessing at this instant is

very slow and the user wants to postpone collecting data until a better link is acquired.

the request with a 513 Message Too Large error response (congestion safety). UA2 home Proxy also has the option of forwarding the request to a Content Storage Server (CSS). That home

proxy learns the maximum acceptable message size from registration by UA2 (regardless of transport protocol) or through configuration.

UA2 home proxy, unaware of the limitations, can forward the large message to UA2, who consequently, can reject the message with error response "413 Request Entity Too Large". UA2 can indicate its maximum acceptable message size using extension in this document.

UA2 home proxy receiving the "413" response can either just forward that response upstream or act as a B2BUA and resend the requested with content-indirection to the CSS.

In the second scenario, UA1, using means outside the scope of this document, posts the contents of the SIP request to the CSS and sends the SIP request with a link to the contents as described in [3]. Home proxies can then fetch the contents and amend the SIP request, or it can be left up to UA2 to fetch the contents. The behaviour of the home proxies receiving the SIP request is described in more detail in the following chapters.

4.0 UA Behaviour

4.1 UAC Sending Requests

A UAC has a choice to either send requests with large contents directly to a proxy, or it can post the contents to a Content Storage Server (CSS) and then send the link to the posted content in the SIP request. Note that posting to the CSS does not need the precondition of large contents in a SIP message.

A UAC sending a SIP Request and wants to make sure that all proxies on the path are congestion-safe proxies MUST insert a "Proxy-Required" header with the tag "congestion-safe" as described in [3].

4.2 UAC Behaviour when receiving Response

A SIP Response with large contents that exceed the limitations is discarded. If the transport protocol is connection-oriented, the connection MAY be closed to stop further sending of the response.

4.2.1 "413" or "513" Error Responses

The max-size header MAY prompt the UAC of the request to send a newly formulated request immediately with either smaller contents or with content-indirection.

If the Retry-After header is also present, the UAC MUST either wait

for that time before sending the same request, or it immediately sends the newly formulated request. It SHOULD NOT resend exactly the same request within the retry-after period. The Retry-After value

MAY be used as an indication of how long this Max-Size constraint is valid for.

Max-size header indicates the SIP message size the UAS can handle for the transport protocol used to send the request. This constraint SHOULD NOT be applied to the same Request sent on different transport protocol.

If an Accept-header is present without the value message/external-body, the UAC MUST NOT send the new message with content-type: message/external-body.

The UAC MAY use this Max-size value to send new, unrelated Requests to the same URI within the duration indicated by the retry-after header. For example: UserA sends a SIP MESSAGE with large content to userB. UserB rejects the request with 413 Request Entity Too Large with a max-size header of value 1300 bytes and a Retry-After value of 80 seconds. UserA now needs to send an unrelated NOTIFY to userB, due to an earlier SUBSCRIBE. This NOTIFY will be sent within the 80 seconds specified. UserA MAY use the 1300 byte constraint to send the NOTIFY. This has to be done with care since the UAS receiving the NOTIFY could be different that the one that received the MESSAGE. The UAC has to be certain that UAS receiving the requests are the same.

4.3 UA Registering Desired Maximum SIP Message Size

Having learnt its capabilities and limitations (by user input or other means), a SIP UA issuing a REGISTER request MAY indicate its acceptable maximum size for a SIP message (including headers and body).

This section introduces a new SIP-URI parameter `max-size`.

A SIP UA registering its address will include this SIP-URI parameter in the contact-header supplied with the REGISTER request.

If the transport parameter is present in the SIP URI of the contact header, then this max-size applies to that particular transport protocol. If transport parameter is not present, then this max-size applies to the default transport protocol used (UDP).

Example REGISTER Request:

```
REGISTER sip:registrar.nokia.com SIP/2.0
Via: SIP/2.0/UDP host1.nokia.com;branch=z9hG4bK346434r
From: sip:hisham.khartabil@nokia.com;tag=31jrlkejrj3kl3jkl879defje3
To: sip:hisham.khartabil@nokia.com
```

Contact: <sip:hisham.khartabil@host1.nokia.com;max-size=1300>
Call-Id: vjn86732nv4fgiofd
CSeq: 1 REGISTER
Max-Forwards:70

Khartabil

[Page 6]

Example REGISTER Request where the max-size applies to TCP:

```
REGISTER sip:registrar.nokia.com SIP/2.0
Via: SIP/2.0/UDP host1.Nokia.com;branch=z9hG4bK346434r
From: sip:hisham.khartabil@nokia.com;tag=31jrlkejrj3kl3jkl879defje3
To: sip:hisham.khartabil@nokia.com
Contact: <sip:hisham.khartabil@host1.nokia.com;transport=tcp;max-size=1300>
Call-Id: vjn86732nv4fgiofd
CSeq: 1 REGISTER
Max-Forwards:70
```

4.4 UAS Receiving A SIP Message With Large Contents

A SIP UAS (terminal or application server like Presence Server) may not have indicated its willingness or capabilities in receiving large message content with the REGISTER request. This may be due to lack of knowledge by the UA, at the time of registration, of its memory or radio link constraints.

A SIP UAS receiving a SIP Request it is unable to handle due to size, regardless of the underlying transport protocol, MAY reject the request with "413 Request Entity Too Large". The UAS MAY close the connection, if the transport protocol was a reliable connection oriented one, to prevent the sender from continuing the request. If the condition is temporary, the UAS SHOULD include a Retry-After header field to indicate that this condition is temporary and after what time the sender can try to send the same request again.

The UAS MAY also indicate its acceptable message size capabilities. Here we introduce a new SIP header "Max-Size". The "413" response sent back by the UAS MAY contain this header. This header contains the maximum acceptable message size by this UAS.

The "413" response MAY also include an Accept-header with value message/external-body indicating its capability to handle content-indirection.

Example response:

```
SIP/2.0 413 Request Entity Too Large
Via: SIP/2.0/UDP proxy.nokia.com;branch=z9hG4bK4kl5jr0iui0giu09df43
Via: SIP/2.0/UDP host1.somewhere.com;branch=z9hG4bK346434r
From: sip:markus.isomaki@nokia.com;tag=4jk123vxbc96
To: sip:hisham.khartabil@nokia.com;tag=fsad98754b6b64m
Call-Id: j4lk2j35879cvx7b
CSeq: 1 MESSAGE
```

Max-Size: 1300
Retry-After: 180
Max-Forwards:70
Accept: message/external-body

Khartabil

[Page 7]

4.5 UAS Receiving a Request with Indirected Content

A UAS may receive a SIP request that had content-indirection performed on it. In this case, the UAS MAY retrieve the contents from the CSS using the URI supplied (See [3] for more details), before it generates a response. It MAY retrieve the contents after it generates the response.

If a request is an INVITE, the UAS MUST retrieve the contents before generating the response.

5.0 Proxy Behaviour

5.1 Congestion Safe Proxy

A congestion safe proxy MUST be compliant with this specification as well as [3] and [4]. This proxy MUST understand the option tag `congession-safe`.

A congestion-safe proxy MUST be able to handle content-indirection if it is configured with a Content Storage Server (CSS) as described in [section 4.2](#). If not configured with CSS, the proxy behave as described in [4]

5.2 Proxy Receiving A SIP Message With Large Content

A proxy MAY be configured with a maximum allowable SIP message size that is destined to a recipient on a certain interface. This value is typically network configured with the lowest Maximum Transfer Unit (MTU) en route to the recipient (UAS) on that interface. The proxy server MUST only use the value when the recipient (UAS) is using UDP as the transport protocol. How this configuration is performed is out side the scope of this document. It is RECOMMENDED that dynamic discovery and configuration is performed. A proxy configured with this value is considered to be congestion-safe.

The proxy MAY also learn the maximum size of a particular UA by means of registration as described in [section 3.1](#) (Note in this case, the value may possibly not be the MTU). This scenario is possible when the registrar for a particular UA is co-located with the proxy (this entity is typically referred to as the home proxy).

When both values are available to the home proxy (the one that arrives in the contact-header of a registration request and the locally configured value), and the transport protocol to be used is connectionless oriented (UDP), the smaller of the two values is

used. If the registered URI uses connection-oriented protocol (such as TCP), then the configured value MUST be ignored.

When the proxy receives a request, it examines the message as follows:

Note: This assumes that the proxy server has accepted to handle a request with size larger than the configured MTU or larger than the registered max-size for that recipient. If the proxy is not willing to do so, it simply rejects the request with 0513 Message Too Large. See [section 4.3](#) for more details.

1. The proxy examines the SIP URI searching for host and transport to send the request to according to procedures in [1].
2. The proxy also searches for the max-size parameter (if a home proxy) and for the configured maximum allowable message size for the interface it will send the request on.
3. If the proxy is a home proxy and transport protocol is TCP, there are 2 cases (since configured value is not used for TCP):
 - a. The proxy is configured with a max message size value and the registered URI does not contain max-size parameter. In this case, the message size is compared to the configured value. If the message size is smaller, then processing proceeds as normal. If the message size is greater, content-indirection MAY be applied. See [section 5.4](#) for more details.
 - b. The proxy is not configured with this value and the registered URI does contain the max-size parameter. In this case, the message size is compared to the registered URI max-size parameter value. If the message size is smaller, then processing proceeds as normal. If the message size is greater, content-indirection MAY be applied. See [section 5.4](#) for more details.
4. If the proxy is a home proxy and the transport protocol is UDP, there are 4 cases:
 - a. Proxy is not configured with this value and registered URI does not contain max-size parameter. In this case, processing proceeds as described in [3].
 - b. Proxy is configured with this value and registered URI does not contain max-size parameter. In this case, the message size is compared to the configured value. If the message size is smaller, then processing proceeds as normal. If the message size is greater, content-indirection MAY be applied. See [section 5.4](#) for more details.
 - c. Proxy is not configured with this value and registered URI does contain max-size parameter. In this case, the message size is compared to the registered URI max-size parameter value. If the

message size is smaller, then processing proceeds as normal. If the message size is greater, content-indirection MAY be applied. See section 5.4 for more details.

- d. Proxy is configured with this value and registered URI does contain max-size parameter. In this case, the message size is compared to the smaller value of the two. If the message size is smaller, then processing proceeds as normal. If the message size is greater, content-indirection MAY be applied. See [section 5.4](#) for more details.
5. If the proxy is not responsible for that domain receives the request and the transport protocol is a connection-oriented one, the request is simply forwarded
 6. If the proxy is not responsible for that domain receives the request and the transport protocol is a connectionless one, there are 2 cases:
 - a. Proxy is not configured with this value. In this case, processing proceeds as described in [4].
 - b. Proxy is configured with this value. In this case, the message size is compared to the configured value. If the message size is smaller, then processing proceeds as normal. If the message size is greater, content-indirection MAY be applied. See [section 5.4](#) for more details.

A SIP Response with large contents that exceeds the maximum acceptable size is discarded. If the transport protocol is connection-oriented, the connection MAY be closed to stop further sending of the response.

5.3 Proxy Refusing To Handle Large SIP Requests

There are circumstances where the proxy refuses to perform content-indirection on a received SIP Request destined to a recipient where a maximum message size constraint has been imposed. These circumstances may include the user not paying for this service or simply that the proxy does not know how to perform content-indirection (see [section 4.1](#) about a congestion safe proxy). The definitions of these circumstances are outside the scope of this document.

In any case, when the proxy refuses to handle this situation or it is simply not a congestion safe proxy (does not understand the option tag `congession-safeö`), it returns a `ö513 Message Too Largeö` response. Reference [4] shows how a proxy SHOULD handle this situation.

The `ö513ö` response MAY include an Accept-header with message/external-body.

5.4 Proxy Performing Content Indirection

Khartabil

[Page 10]

A proxy that is a congestion safe proxy MUST handle content indirection.

Here we define a new functional SIP entity called Content Storage Server (CSS). A congestion safe proxy MUST be configured with the CSS address.

Once the home proxy has accepted to perform content indirection on a SIP Request, it forwards the message to the CSS. Section 16.6 of [1] defines the steps to follow. Pay special attention to step 6 where it discusses how a proxy may divert a SIP Request to a specific next hop. Below is a rewrite of that section tailored to the terminology used in this document:

A proxy MAY mandate that a request visit a CSS before being delivered to the destination. This is to accommodate content indirection. A proxy MUST ensure that CSS is a loose router. Generally, this can only be known with certainty if the CSS is within the same administrative domain. The CSS is represented by a URI (which contains the lr parameter). This URI MUST be pushed into the Route header field ahead of any existing values, if present. If the Route header field is absent, it MUST be added, containing that URI.

5.5 Proxy Receiving 413 Response From Downstream

Open issue: should we just choose to forward the response upstream?

A UA or proxy refusing to handle a large SIP Request may reject the request with 413 error response as describe in [section 3.2](#).

When a proxy receives this response, it MAY choose to handle the request itself. It does so by reissuing the request, but this time, it sends it via the CSS using the route-header, as described in [section 5.4](#). The proxy MAY also send a 181 Call Is Being Forwarded provisional response back to the sender.

There are situations where the proxy refuses to handle the request itself and alternatively just forward the 413 response to sender. See [section 5.3](#) for more details.

5.6 Proxy Receiving a Request With Indirected Content

A proxy may receive a SIP request that had content-indirection performed on it. In this case, the proxy MAY perform one of 2 things:

- a. Retrieve the contents and amend the request to contain those contents. It is RECOMMENDED for a proxy to do so if the CSS is within the same administrative domain as the proxy. It is also RECOMMENDED for a proxy to do so if the CSS is not within the

administrative domain as the proxy, but there is a trust relationship between the two domains. The proxy sends the new request to the destination, maintaining any tags in the To and From headers sent in the original request. This is to accommodate for SIP requests within dialogs being delivered to the right dialog.

- b. Forward the request without retrieving the contents. It is NOT RECOMMENDED for a proxy forwards the request without first retrieving the content and amending the request to contain it, if the CSS is within the same administrative domain as the proxy.

6.0 Content Storage Server (CSS) Behaviour

The CSS behaves as a SIP Back To Back User Agent (B2BUA). When it receives a SIP Request performs the following steps:

- a. Extracts the body of the Request and stores it in an HTTP server.
- b. Responds to the sender with a 202 Accepted response. INVITE is a special case. The CSS MUST NOT respond with 202 in this case, but instead wait for a response from the UAS.
- c. Formulates a new request that includes the URL where the content can be retrieved. This procedure is described in [3].
- d. Sends the new request to the destination, maintaining any tags in the To and From headers sent in the original request. This is to accommodate for SIP requests within dialogs being delivered to the right dialog.

OPEN ISSUE: Should the behaviour of CSS be uniform across all requests? I.e. Not generate a 202 response at all.

7.0 Syntax for Extensions

7.1 Max-Size Header

Max-Size = Max-Size HCOLON delta-seconds

Additions to SIP Table 3:

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG

Max-Size	413	a	-	-	-	-	-	-

7.2 Max-size parameter

Max-size-param = ômax-size=ö 1*DIGIT

Khartabil

[Page 12]

8.0 Security considerations

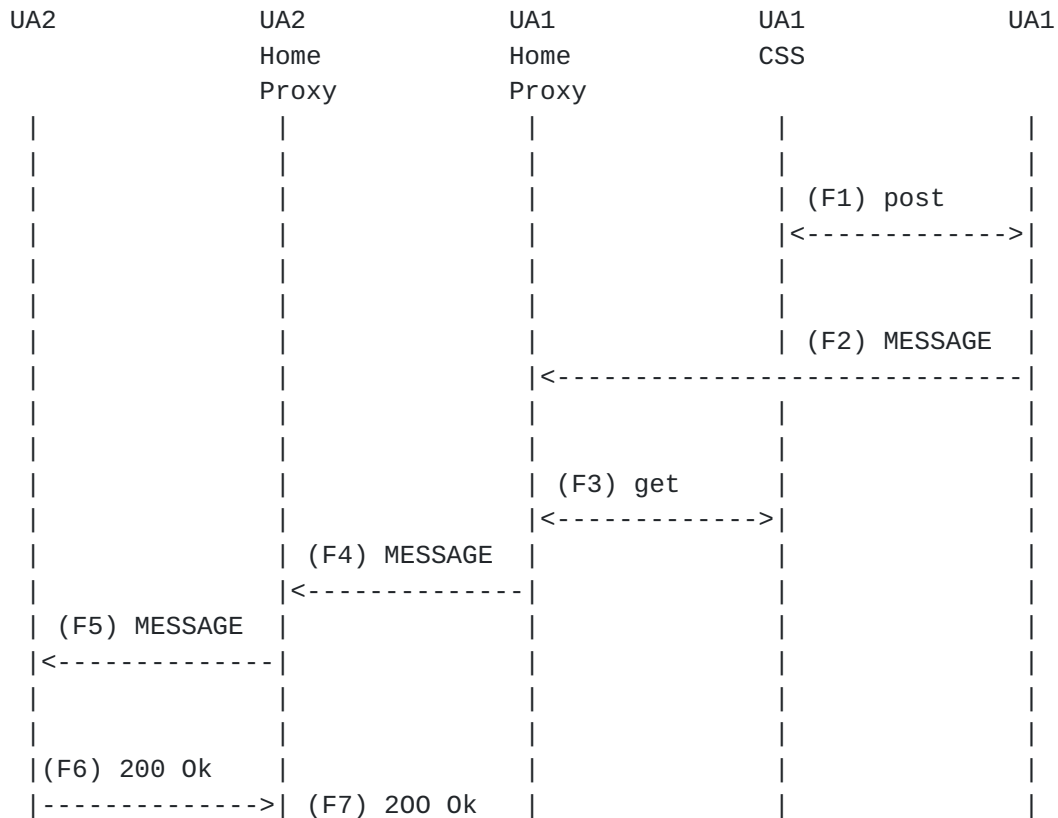
The presence of Max-size header in a SIP message has a significant effect on the ways in which the request is handled at a server. As a result, it is especially important that messages containing this extension be authenticated. The same holds true for registrations with contact parameters.

Processing of requests and looking up criteria for message size requires set operations and searches, which can require some amount of computation. This enables a DoS attack whereby a user can send requests with substantial numbers messages with large contents, in the hopes of overloading the server. To counter this, the server SHOULD authenticate the sender.

REGISTER requests can reveal sensitive information about a UA's capabilities. If this information is sensitive, it SHOULD be encrypted using SIP S/MIME capabilities.

9.0 Examples

9.1 UAC Posting Contents to CSS Before Sending SIP MESSAGE



	----->	(F8) 200 0k	
			----->



(F1) UA1 posts, using some means outside this document, the contents to the CSS

(F2) UA1 sends the SIP MESSAGE with content-type: message/external-body

```

MESSAGE sip:ua2@nokia.com SIP/2.0
Via: SIP/2.0/UDP host1.somewhere.com;branch=z9hG4bK346434r
From: sip:ua1@somewhere.com;tag=31jrlkejr3kl3jkl879defje3
To: sip:ua1@nokia.com
Contact: <sip:ua1@host1.nokia.com;max-size=1300>
Call-Id: vjn86732nv4fgiofd
CSeq: 1 MESSAGE
Max-Forwards: 70
Content-Type: message/external-body; access-type=URL;

```

```

url="http://131.228.13.2/im/9207807c03d4a5e0e6907b0dc89c9067";
Content-Length: 32

```

Content-Type: text/html

(F3) UA1 home proxy fetches the contents and amends the SIP MESSAGE with the new contents

(F4) Newly constructed MESSAGE send to the domain of the recipient

```

MESSAGE sip:ua2@nokia.com SIP/2.0
Via: SIP/2.0/TCP au1-homeproxy.nokia.com;branch=z9hG4bKewr1kjdfkjl
Via: SIP/2.0/UDP host1.somewhere.com;branch=z9hG4bK346434r
From: sip:ua1@somewhere.com;tag=31jrlkejr3kl3jkl879defje3
To: sip:ua1@nokia.com
Call-Id: vjn86732nv4fgiofd
CSeq: 1 MESSAGE
Max-Forwards: 69
Content-type: text/html
Content-length: 20000

```

[Large Body]

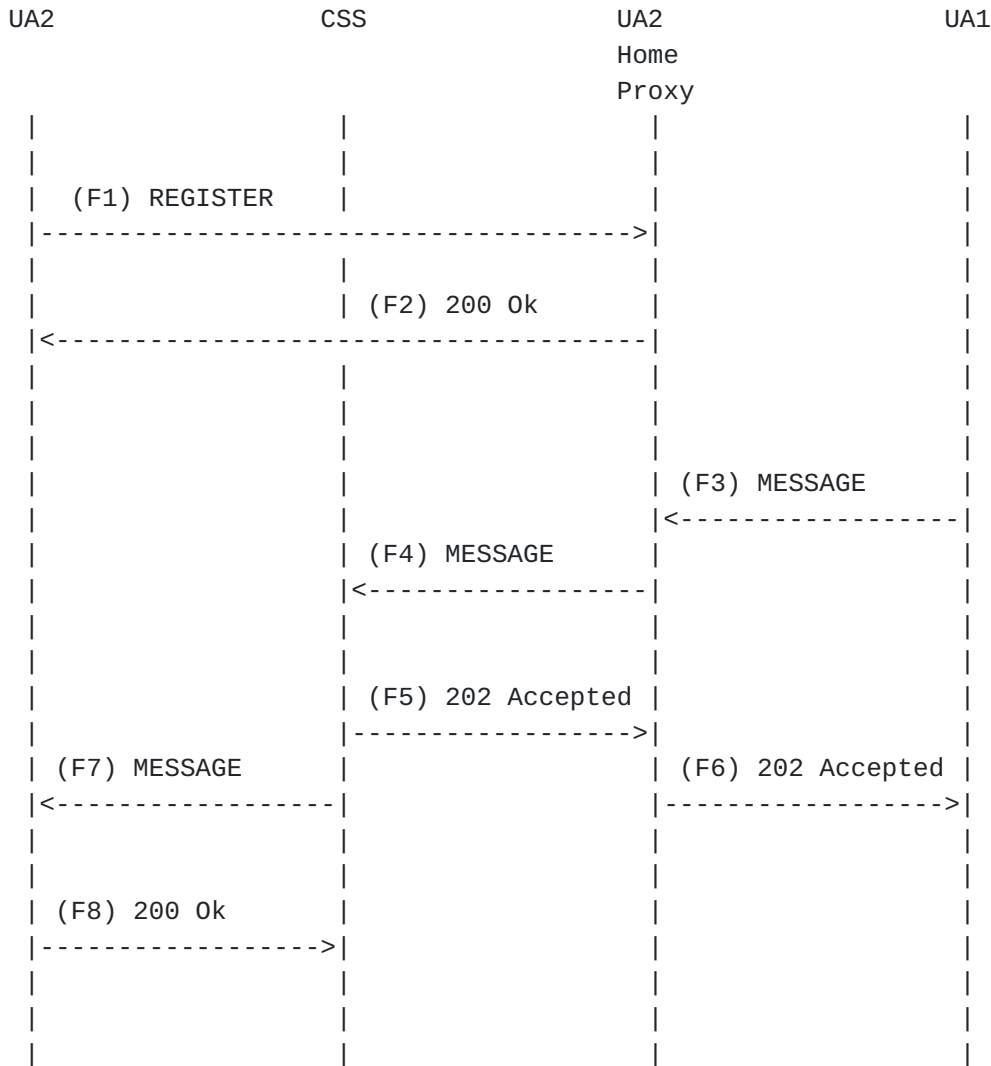
(F5) UA2 home proxy decides that the content of this message is not too large and therefore does not perform content redirection of the message

(F6) û (F8) ô200 Okö for the request.

Khartabil

[Page 14]

9.2 Receiving A MESSAGE With Large Contents after a REGISTER, Home Proxy Performs Content-Indirection



(F1) REGISTER sent from UA2 to Home Proxy (Registrar)

```

REGISTER sip:registrar.nokia.com SIP/2.0
Via: SIP/2.0/UDP host2.somewhere.com;branch=z9hG4bKue73jhhdue
From: sip:ua2@somewhere.com;tag=48er8fdjkcfnclue9843ifj
To: sip:ua2@somewhere.com
Call-Id: 3mkejdc9e9834judjd
CSeq: 1 REGISTER
Expires: 3600
Contact: <sip:ua2@host2.nokia.com;max-size=1300>
Max-Forwards: 70
  
```

(F2) 200 Ok from Home proxy to UA2

SIP/2.0 200 Ok
Via: SIP/2.0/UDP host2.somewhere.com;branch=z9hG4bKue73jhhdue
From: sip:ua2@somewhere.com;tag=48er8fdjkcfcnciue9843ifj
To: sip:ua2@somewhere.com;tag=393k3l3n3n3934

Khartabil

[Page 15]

Call-Id: 3mkejdc9e9834judjd
CSeq: 1 REGISTER
Contact: <sip:ua1@host2.nokia.com;max-size=1300>,expires=3600
Max-Forwards: 70

(F3) Message sent from UA1 to UA2 with very large content

MESSAGE sip:ua2@nokia.com SIP/2.0
Via: SIP/2.0/UDP host1.somewhere.com;branch=z9hG4bK346434r
From: sip:ua1@somewhere.com;tag=31jrlkejrq3kl3jkl879defje3
To: sip:ua1@nokia.com
Call-Id: vjn86732nv4fgiofd
CSeq: 1 MESSAGE
Max-Forwards: 70
Content-type: text/html
Content-length: 20000

[Large Body]

(F4) MESSAGE is directed by UA2's home proxy to CSS. Home proxy accepts to content-indirect

MESSAGE sip:ua2@nokia.com SIP/2.0
Via: SIP/2.0/TCP homeproxy.nokia.com;branch=z9hG4bKewr1kjdfkjl
Via: SIP/2.0/UDP host1.somewhere.com;branch=z9hG4bK346434r
From: sip:ua1@somewhere.com;tag=31jrlkejrq3kl3jkl879defje3
To: sip:ua1@nokia.com
Call-Id: vjn86732nv4fgiofd
CSeq: 1 MESSAGE
Route: sip:css.nokia.com
Max-Forwards: 69
Content-type: text/html
Content-length: 20000

[Large Body]

(F5) CCS returns a 202 Accepted

SIP/2.0 202 Accepted
Via: SIP/2.0/TCP homeproxy.nokia.com;branch=z9hG4bKewr1kjdfkjl
Via: SIP/2.0/UDP host1.somewhere.com;branch=z9hG4bK346434r
From: sip:ua1@somewhere.com;tag=31jrlkejrq3kl3jkl879defje3
To: sip:ua1@nokia.com;tag=5nixucvzo3241bqewmn
Contact: <sip:ua1@host1.nokia.com;max-size=1300>
Call-Id: vjn86732nv4fgiofd
CSeq: 1 MESSAGE
Max-Forwards: 70

(F6) Home proxy forwards response to UA1

SIP/2.0 202 Accepted

Via: SIP/2.0/UDP host1.somewhere.com;branch=z9hG4bK346434r

From: sip:ua1@somewhere.com;tag=31jr1kejrq3k13jkl879defje3

Khartabil

[Page 16]

To: sip:ua1@nokia.com;tag=5nixucvzo3241bqewmn
 Contact: <sip:ua1@host1.nokia.com;max-size=1300>
 Call-Id: vjn86732nv4fgiofd
 CSeq: 1 MESSAGE
 Max-Forwards: 69

(F7) CCS sends the content-indirected MESSAGE to UA2 with the URL

MESSAGE sip:ua2@nokia.com SIP/2.0
 Via: SIP/2.0/UDP css.nokia.com;branch=z9hG4bK342kj5klj43klndsm
 From: sip:ua1@somewhere.com;tag=31jrlkejr3kl3jkl879defje3
 To: sip:ua1@nokia.com
 Contact: <sip:ua1@host1.nokia.com;max-size=1300>
 Call-Id: vjn86732nv4fgiofd
 CSeq: 1 MESSAGE
 Max-Forwards: 70
 Content-Type: message/external-body; access-type=URL;

url="http://131.228.13.2/im/9207807c03d4a5e0e6907b0dc89c9067";
 Content-Length: 32

Content-Type: text/html

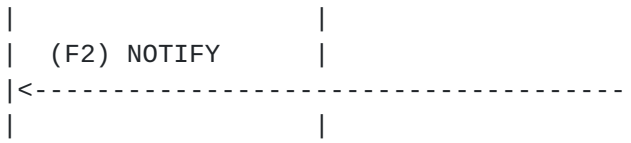
(F8) 200 Ok is returned by UA2

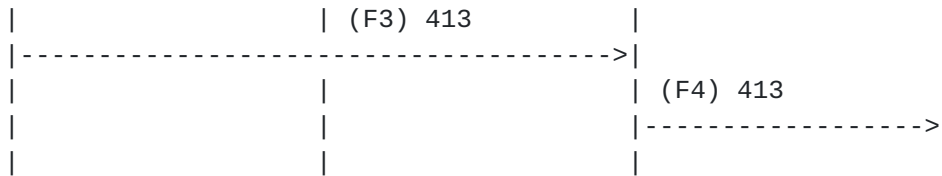
SIP/2.0 200 Ok
 Via: SIP/2.0/UDP css.nokia.com;branch=z9hG4bK342kj5klj43klndsm
 From: sip:ua1@somewhere.com;tag=31jrlkejr3kl3jkl879defje3
 To: sip:ua1@nokia.com;tag=dfjkl43kj15ldskfj
 Contact: <sip:ua1@host1.nokia.com;max-size=1300>
 Call-Id: vjn86732nv4fgiofd
 CSeq: 1 MESSAGE
 Max-Forwards: 70

9.3 Receiving A NOTIFY With Large Contents, UAS Refuses To Handle Request

This example assumes that a subscription has been established. It also assumes that UA2 has not registered a max-size limit and that the home proxy configured max-size is less than the size of the NOTIFY.







(F1) NOTIFY sent from PS to UA2, via home proxy with very large content

```

NOTIFY sip:ua2@host2.nokia.com SIP/2.0
Via: SIP/2.0/TCP ps.nokia.com;branch=z9hG4bK346434r
From: sip:ps.nokia.com;tag=31jrlkejr3kl3jkl879defje3
To: sip:ua2@nokia.com;tag=eqwriuo423nf
Call-Id: vjn86732nv4fgiofd
CSeq: 2 NOTIFY
Route: sip: homeproxy.nokia.com
Max-Forwards: 70
Content-Type: application/cpim-pidf+xml
Content-length: 20000
  
```

[Large Body]

(F2) NOTIFY forwarded from home proxy to UA2 with very large content. Home proxy checked its max-size configuration and found that this message passes.

```

NOTIFY sip:ua2@host2.nokia.com SIP/2.0
Via: SIP/2.0/TCP homeproxy.nokia.com;branch=z9hG4bKre78re89jfdkj
Via: SIP/2.0/TCP ps.nokia.com;branch=z9hG4bK346434r
From: sip:ps.nokia.com;tag=31jrlkejr3kl3jkl879defje3
To: sip:ua2@nokia.com;tag=eqwriuo423nf
Call-Id: vjn86732nv4fgiofd
CSeq: 2 NOTIFY
Max-Forwards: 69
Content-Type: application/cpim-pidf+xml
Content-length: 20000
  
```

[Large Body]

(F3) UA2 refuses the request due to its large content and returns a 413 response

```

SIP/2.0 413 Message Too Large
Via: SIP/2.0/TCP homeproxy.nokia.com;branch=z9hG4bKre78re89jfdkj
Via: SIP/2.0/TCP ps.nokia.com;branch=z9hG4bK346434r
From: sip:ps.nokia.com;tag=31jrlkejr3kl3jkl879defje3
To: sip:ua2@nokia.com;tag=eqwriuo423nf
Call-Id: vjn86732nv4fgiofd
CSeq: 2 NOTIFY
  
```

Max-size: 1300
Max-Forwards: 70

Khartabil

[Page 18]

(F4) 413 passed by home proxy to PS. Home proxy did not content-indirect.

SIP/2.0 413 Message Too Large

Via: SIP/2.0/TCP ps.nokia.com;branch=z9hG4bK346434r

From: sip:ps.nokia.com;tag=31jrlkejrq3kl3jkl879defje3

To: sip:ua2@nokia.com;tag=eqwriuo423nf

Call-Id: vjn86732nv4fgiofd

CSeq: 2 NOTIFY

Max-size: 1300

Max-Forwards: 69

10.0 IANA Considerations

Once the header and parameter names have been agreed on, they will be registered with IANA.

11.0 Acknowledgements

I would like to thank Jose Costa-Requena, Mikko Lonnfors, Pekka Pessi and Nokia for their comments and support.

12.0 References

[1] J. Rosenberg, et al. "SIP: Session Initiation Protocol". RFC 3261, Internet Engineering Task Force, June 2002.

[2] B. Campbell et al. "SIP Extensions for Instant Messaging", [RFC3428](#), Internet Engineering Task Force, December 2002.

[3] S. Olson. "A Mechanism for Content Indirection in SIP Messages", [draft-ietf-sip-content-indirect-mech-01.txt](#). Internet Draft, Internet Engineering Task Force, November 2002. Work in progress.

[4] D. Willis. "SIP Extension to Assure Congestion Safety", [draft-ietf-sip-congestsafe-00.txt](#). Internet Draft, Internet Engineering Task Force, August 2002. Work in progress.

[5] S. Bradner, "Key words for use in RFCs to indicate requirement levels," [RFC 2119](#), Internet Engineering Task Force, Mar. 1997.

[3] S. Olson. Et al. "SIMPLE Presence Publication Mechanism", [draft-olson-simple-publish-01](#). Internet Draft, Internet Engineering Task Force, October 2002. Work in progress.

Authors' Addresses

Hisham Khartabil
Nokia

P.O. Box 321

Khartabil

[Page 19]

FIN-00045
NOKIA GROUP
FINLAND

Email: Hisham.Khartabil@nokia.com
Tel: + 358 7180 76161

Full Copyright Statement

Copyright (C) The Internet Society (2003). 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.

