Internet Draft
Document: draft-barnes-sipping-history-info-02.txt
Category: Standards Track

M. Barnes
M. Watson
Nortel Networks
Cullen Jennings
Cisco

Expires: August, 2003
February 2003

An Extension to the Session Initiation Protocol for Request History Information

Status of this Memo

Abstract

This draft defines a standard mechanism for capturing the history information associated with a SIP request. This capability enables many enhanced services by providing the information as to how and why a call arrives at a specific application or user. This draft defines a new optional SIP header, History-Info, for capturing the history information in requests. A new option tag, HistInfo, to be included in the Supported header is defined to allow UAs to indicate whether the HistInfo should be returned in responses to a request which has captured the history information.

Table of Contents

1 Request History Information Description.....3
1.1 Optionality of History-Info.....3
1.2 Securing History-Info.....3
1.3 Ensuring the Privacy of History-Info.....4
2 Request History Information Protocol Details.....4
2.1 Protocol Structure of History-Info.....4
2.2 Protocol Examples.....5

2.3 Protocol usage.....5
2.4 Security for History-Info.....7
2.5 Example Applications using History-Info.....8
3 Security Considerations.....9
References.....9
Appendix A Forking Scenarios.....10
A.1 Sequentially forking (Hist-Info in Response).....10
A.2 Sequential Forking (with Success).....11
Appendix B Voicemail.....12

Appendix C Automatic Call Distribution Example.....16
Appendix D Solution options analysis.....18
Full Copyright Statement.....20

Overview

This document provides the solution for the Request History requirements as defined in [1].

The fundamental functionality provided by the request history information is the ability to inform proxies and UAs involved in processing a request about the history or progress of that request. This functionality provides a standard mechanism for capturing the request history information to enable a wide variety of services for networks and end users, without prescribing the operation of those services.

Section 1 provides an overall description of the solution, providing references to the appropriate requirements met by each aspect of the solution. For background, further detail on some aspects of the solution with regards to optionality and the detailed protocol requirements is provided in Appendix D.

Section 2 provides the details of the additions to the SIP protocol, which are required to capture the Request History information. An example use of the request history information is included in Section 2, with additional scenarios included in the Appendix. It is anticipated that these would be moved and progressed in the Service examples draft [2] or individual informational drafts describing these specific services, since History-Info is just one of the building blocks for implementing these services. Individual drafts would be particularly useful for documenting services for which there are multiple solutions, since the use of the request history information isn't prescriptive.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [7].

In order to provide a cross reference of the solution description to the requirements defined in [1] without reiterating the entirety of the requirements in this document, the requirements are referenced as [REQNAME-req] following the text or paragraph which explicitly satisfies the requirement.

Definitions

The following terminology is used in this document:

Retarget (as defined in [1]): The process of a Proxy Server/UAC changing a URI in a request and thus changing the target of the request.

Retargeted: past of Retarget.

Retargeted-from-URI: The URI or address from which the request was retargeted.

Retargeted-to-URI: The new URI or address to which the request is in the process of being retargeted.

1 Request History Information Description

The fundamental functionality provided by the request history information is the ability to inform proxies and UAs involved in processing a request about the history or progress of that request [CAPABILITY-req]. The solution for the capture of the Request History Information defines a new header for SIP messages: History-Info [CONTENT-req].

The Request History Information can appear in any request not associated with an established dialog, which includes INVITE, REGISTER, MESSAGE and OPTIONS [REQUEST-VALIDITY-req] and any valid response to these requests.[ISSUER-req]

Request History Information is captured when a request is retargeted. In some scenarios, it might be possible for more than one instance of retargeting to occur within the same Proxy. A proxy SHOULD also generate request history information for the 'internal retargeting'. An entity (UA or proxy) retargeting in response to a redirect or REFER SHOULD include any Request History information from the redirect/REFER in the new request [GENERATION-req, FORWARDS-req].

1.1 Optionality of History-Info

The Request History Information is optional in that neither UAs nor Proxies are required to support it. The requirement for Request History information to be returned in Responses is indicated using a new Supported header: HistInfo [BACKWARDS-req]. In addition, local policy can define whether or not the information is captured by the retargeting entity for any request, or a specific Request-URI, being retargeted. In many instances, it is likely that this could restrict the applicability of services which make use of the Request History Information to be limited to retargeting within domain(s) controlled by the same local policy, or between domain(s) which negotiate policies with other domains to ensure support of the given policy, or services for which "complete" History Information isn't required to provide the service. [OPTIONALITY-req] Thus, it is highly recommended that all applications making use of the request history information clearly define the impact of the information not being available and specify the processing of such a request.

1.2 Securing History-Info

This draft defines a new header for SIP. Since, the Request History information is being inserted by an entity as it targets a Request, the resulting security requirements introduce a slightly different problem than the basic SIP header or Identity problem. For History-Info, the general requirement is to secure information that is inserted by a proxy. It is primarily the captured Request-URIs that are the security concern, since they can reflect some aspect of a user's identity and service routing. Thus, the primary objective of the security solution is to ensure that the information being captured is protected from being accessed or manipulated by non-authorized entities, with the fundamental

assumption that retargeting entities are implicitly authorized. The draft does suggest the use of a secure transport mechanism such as TLS to ensure the overall confidentiality of the History-Info[SEC-req-4]. However, the complete security solution for History-Info depends upon a general solution for protecting the captured information. This will be addressed in a separate solution draft [TBD]. Details of the use of this proposed mechanism to satisfy the security requirements are provided in section 2.4.

The security associated with the Request History Information is optional and depends upon local policy and the impact on specific applications of having the information compromised. Since, the Request History Information itself is also optional and it has been recommended that applications document the impact of the information not being available, it is also suggested that the impact of not supporting the security recommendations also be documented to ensure that it is sufficiently addressed by the application.

1.3 Ensuring the Privacy of History-Info

In order to satisfy the requirements of ensuring that the privacy associated with a retargeted request is maintained by the retargeting entity [PRIV-req-1] and by the receiving entity [PRIV-req-2], the retargeting entity must determine if there is any privacy associated with a request being retargeted. In some scenarios, the Privacy header would indicate whether the headers in a message should be privacy protected. However, the basic assumption is that local policy would be used to determine whether a specific request should have its privacy maintained and whether maintaining that privacy means that a specific request URI would NOT be captured or that it would be appropriately Privacy protected if it were captured. The proposal for ensuring that the privacy is protected is to recommend the use of a Privacy Service as defined by [6] for headers.

It is recognized that meeting the privacy requirements can impact the functionality of this solution by overriding the request to generate the information. As with the optionality and security requirements, applications making use of History-Info should address any impact this may have.

2 Request History Information Protocol Details

This section contains the details and usage of the proposed new SIP protocol elements. It also discusses the security aspects of the solution and provides some examples.

2.1 Protocol Structure of History-Info

History-Info is a header field as defined by [4]. It can appear in any request not associated with an established dialog, which includes INVITE, REGISTER, MESSAGE and OPTIONS and any valid response to these requests.

It carries the following information:

- Targeted-to-URI: the Request URI captured as the Request is targeted. By capturing a copy of the Request URI in the

initial request, the Retargeted-from-URI is already captured when a request is retargeted and the Retargeted-to-URI is being captured.

- Reason: An optional parameter for History-info. The reason for the retargeting is captured by including the Reason Header [3] as part of the captured Request URI.

- Index: An optional parameter for History-Info reflecting the chronological order of the information, indexed to also reflect the forking and nesting of requests.

The semantics of the captured Targeted-to-URIs are derived from the current context of the request as follows:

- Retargeted-from-URI: this is the Request URI that is being changed due to the retargeting. It is the Targeted-to-URI in the request received by the retargeting entity. If it was not explicitly captured by the original sender/forwarder of the request, it would be captured and added to the request prior to the Targeted-to-URI currently being captured. If the sender/forwarder supported History-Info, it would have been added prior to sending/forwarding the Request.

- Retargeted-to-URI: this is the Targeted-to-URI being captured in the request being retargeted.

The following summarizes the syntax of the History-Info header, based upon the standard SIP syntax [4]:

```
History-Info = ("History-Info" / "h") HCOLON

                hist-info *(COMMA hist-info)

hist-info = hi-targeted-to-uri *( SEMI HI-param )

hi-targeted-to-uri= name-addr

hi-param = hi-index / hi-extension

  hi-index = "index" EQUAL 1*DIGIT *(DOT 1*DIGIT)

hi-extension = generic-param
```

2.2 Protocol Examples

```
History-Info:<sip:UserA@ims.nortelnetworks.com?Reason=SIP;
cause=302;text=öMoved Temporarilyö>; foo=bar
```

```
History-Info: <sip:45432@vm.nortelnetworks.com?
Reason=SIP;cause=486;text="Busy Here"> ; index=1.1.2
```

2.3 Protocol usage

This section describes the processing specific to UAs and Proxies for the History-Info and the HistInfo option tag.

[Editor's note: Once the Security solution is fully fleshed out, it may be reasonable to move this section 2.3 after section 2.4 and provide the detailed security related processing prior to this section, so that security aspects can be highlighted in this section, as well.]

2.3.1 UAC Behavior

The UAC SHOULD include the HistInfo option tag in the Supported header in any request not associated with an established dialog for which the UAC would like the History-Info in the Response. In addition, the UAC should initiate the capturing of the History Information by capturing the Request-URI as the hi-targeted-to-uri and initializing the index to 1.

The processing of the History-Info received in the response is application specific and outside the scope of this draft.

2.3.2 UAS Behavior

The processing of History-Info by a UAS in a Request depends upon local policy and specific applications at the UAS which might make use of the information. If the HistInfo option tag is received in a request, the UAS should include any History-Info received in the request in the subsequent response.

2.3.3 Proxy Behavior

The use of History-Info does not alter the fundamental processing of proxies for determining request targets as defined in section 16.5 of [4]. Whether a proxy captures the History-Info depends upon several factors:

- Whether the Request contains the HistInfo option tag in the Supported header.
- Local Policy

  The following are further considerations for refinement of a local policy supporting History-Info:
  - Whether retargeting within a Proxy is captured
  - Whether the History-Info captured for a proxy/domain should go outside that domain (e.g. a Proxy knows that the information is potentially useful within that domain, however, policies (for privacy, user and network security, etc.) prohibit the exposure of that information outside that domain).

  Each application making use of History-Info should address the applicability and impacts of the local policies.

Consistent with basic SIP processing of optional headers, proxies should maintain History-Info captured by other domains, received in messages which they forward, independent of whether local policy supports History-Info.

The specific processing by proxies for capturing the History-Info in Requests and Responses is described in detail in the following sections.

2.3.3.1 Capturing History-Info in Requests

If the proxy supports History-Info, the proxy SHOULD add any History-Info collected as it retargets a Request. The SIP Response Code should be included in the Reason header of the Targeted-to-URI. The History-Info SHOULD be added following any History-Info received in the request being forwarded. Additionally, if a request is received that doesn't include a captured Request URI from the previous entity, the proxy MAY add an additional entry, effectively capturing the retargeted-from-URI in the Request.

In order to maintain ordering and accurately reflect the nesting and retargeting of the request, it is recommended that an index be included along with the Targeted-to-URI being captured. The basic rule for adding the index are to read the value from the previous History-Info, if available, and capture the index.n as the index for the History-Info being captured, where n would typically be 1 for a forwarded request. Thus, the level of nesting of the index reflects the number of hops. For retargets within a proxy, the proxy MUST maintain the current level of nesting by incrementing the lowest/last digit of the index for each instance of retargeting, thus reflecting the number of retargets within the proxy. If there is no previous History-Info entry, or index in the previous entry, an index MAY be included for the current entry, with the index starting at 1. An index SHOULD NOT be added in the scenario whereby the received request had no History-Info header and the retargeted-from-URI is being captured for completeness.

Parallel forking, as with basic SIP processing, does introduce somewhat of a special case. In the case of parallel forking, the proxy SHOULD capture each of the Request-URIs to which the Request is forked in the manner previously described. However, since the forking is parallel, it's recommended that rather than attempt to send the logical order of the requests being sent, that the information for subsequent requests or responses is built upon receipt of the initial response to ensure that the series of any subsequent forking and retargeting of any of the forked requests accurately reflects the logical sequence. Again, it is recommended that the index be captured for each forked request following a similar model as that previously described, with each new Request having a unique index. The lack of Reason headers in the captured Request-URIs should be indicative of the parallel nature of forking (i.e the Request-URIs are not the result of retargets, but are rather all simultaneous Targeted-To URIs.)

2.3.3.2 Processing History-Info in Responses

A proxy that receives a Request with the HistInfo option tag in the Supported header, and depending upon a local policy supporting the capture of History-Info, SHOULD return captured History-Info in

subsequent, provisional and final, responses to the Request.

2.4 Security for History-Info

As discussed in Section 1, the security requirements are met by recommending the use of TLS (a basic SIP requirement per [4]) and through the use of the security solution defined in [TBD].

2.4.1 Security examples

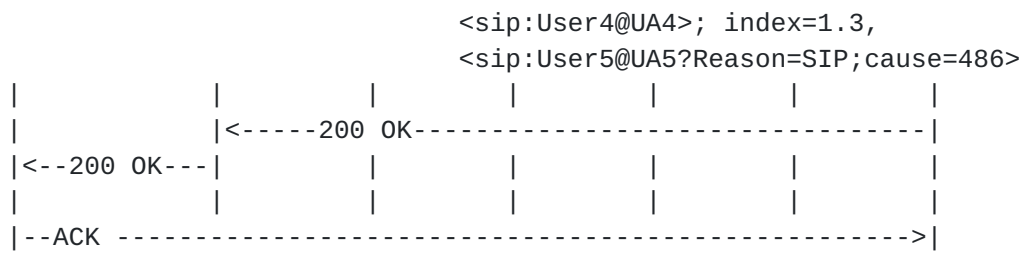
[Editor's Note: Need to add some protocol details based on the use of S/MIME for protecting History-Info once [TBD] is further along].

2.5 Example Applications using History-Info

This scenario highlights an example where the History-Info in the response is primarily of use in not retrying routes that have already been tried by another proxy. Note, that this is just an example and that there may be valid reasons why a Proxy would want to retry the routes and thus, this would like be a local proxy or even user specific policy.

UA 1 sends a call to "Bob" to proxy 1. Proxy 1 forwards the request to Proxy 2. Proxy 2 parallel forks and tries several places (UA2, UA3 and UA4) before sending a response to Proxy 1 that all the places are busy. Proxy 1, without the History-Info, would try several of the same places (UA3 and UA4)based upon registered contacts for "Bob", before completing at UA5. However, with the History-Info, Proxy 1 determines that UA3 and UA4 have already received the invite, thus the INVITE goes directly to UA5.

```
                                <sip:User4@UA4>; index=1.3
   |            |         |        |        |        |        |
  /* Upon receipt of the response, P1 determines another route for the
INVITE, but finds that it matches some routes already attempted (e.g.
UA2 and UA3, thus the INVITE is only forwarded to UA5, where the
session is successfully established  */
   |            |         |        |        |        |        |
   |            |-----------------INVITE ---------------------->|
                  History-Info: <sip:Bob@P1>, <sip:Bob@P2>; index=1,
                                <sip:User2@UA2>; index=1.1,
                                <sip:User3@UA3>; index=1.2,
```



Additional detailed scenarios are available in the appendix.

3 Security Considerations

This draft provides a proposal for addressing the Security requirements identified in [1] in sections 1.2 and 2.4 of this draft by proposing the use of TLS between entities. The protection of the History-Info is dependent upon a general solution for securing headers added by proxies. This general solution is described in [TBD] [Editor's note: Assumed to be based upon the SIP Authenticated Identity body model defined in [5].]

4 IANA Considerations

(Note to RFC Editor: Please fill in all occurrences of XXXX in this section with the RFC number of this specification).

This document defines a new SIP header field name with a compact form: History-Info and h respectively, and a new option tag: HistInfo.

The following changes should be made to http:///www.iana.org/assignments/sip-parameters

The following row should be added to the header field section:

Header Name	Compact Form	Reference
History-Info	h	[RFCXXXX]

The following should be added to the Options Tags section:

Name	Description	Reference
HistInfo	When used with the Supported header, this option tag indicates support for the History Information to be captured for requests and returned in subsequent responses. This tag is not used in a Proxy-Require or Requires header field since support of	[RFCXXXX]

History-Info is optional.

References

[1] M. Barnes, M. Watson, C. Jennings, J. Peterson, "SIP Generic Request History Capability û Requirements", draft-ietf-sipping-req-history-02.txt, February, 2003.

[2] A. Johnson, "SIP Service Examples", draft-ietf-sipping-service-examples-03.txt, November, 2002.

[3] H. Schulzrinne, D. Oran, G. Camarillo, "The Reason Header Field for the Session Initiation Protocol", RFC 3326, December, 2002.

[4] J. Rosenberg et al, "SIP: Session initiation protocol," RFC 3261, June, 2002.

[5] J. Peterson, " SIP Authenticated Identity Body (AIB) Format", draft-ietf-sip-authid-body-00.txt, October, 2002.

[6] J. Peterson, "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November, 2002.

[7] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[8] J. Peterson, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-00.txt, October, 2002.

[9] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

Acknowledgements

The authors would like to acknowledge the constructive feedback provided by Robert Sparks, Rohan Mahy, Paul Kyzivat, Scott Orton, John Elwell, Francois Audet, Anthony Brown, and Jayshree Bharatia.

Authors' Addresses

Mary Barnes
Nortel Networks
2380 Performance Drive Phone: 1-972-684-5432
Richardson, TX USA Email: mbarnes@nortelnetworks.com

Cullen Jennings
Cisco Systems
170 West Tasman Dr Tel: +1 408 527 9132
MS: SJC-21/3 Email: fluffy@cisco.com

Mark Watson
Nortel Networks (UK)
Maidenhead Office Park (Bray House)
Westacott Way
Maidenhead,
Berkshire Tel: +44 (0)1628-434456
England Email: mwatson@nortelnetworks.com

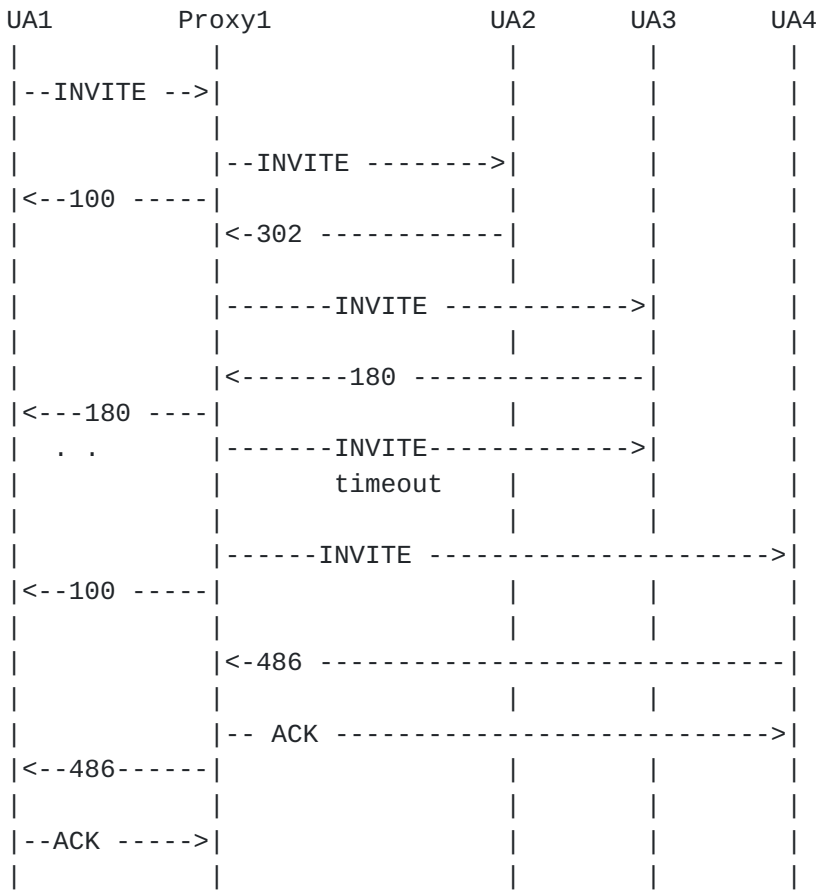
Appendix A Forking Scenarios

A.1 Sequentially forking (Hist-Info in Response)

This scenario highlights an example where the History-Info in the response is useful to an application or user that originated the request.

UA 1 sends a call to "Bob" via proxy 1. Proxy 1 sequentially tries several places (UA2, UA3 and UA4) unsuccessfully before sending a response to UA1.

This scenario is provided to show that by providing the History-Info to UA1, the end user or an application at UA1 could make a decision on how best to attempt finding "Bob". Without this mechanism UA1 might well attempt UA3 (and thus UA4) and then re- rd attempt UA4 on a 3 manual attempting at reaching "Bob". With this mechanism, either the end user or application could know that "Bob" is busy on his home phone and is physically not in the office. If there were an alternative address for "Bob" known to this end user or application, that hasn't been attempted, then either the application or the end user could attempt that. The intent here is to highlight an example of the flexibility of this mechanism that enables applications well beyond SIP as it is certainly well beyond the scope of this draft to prescribe detailed applications.

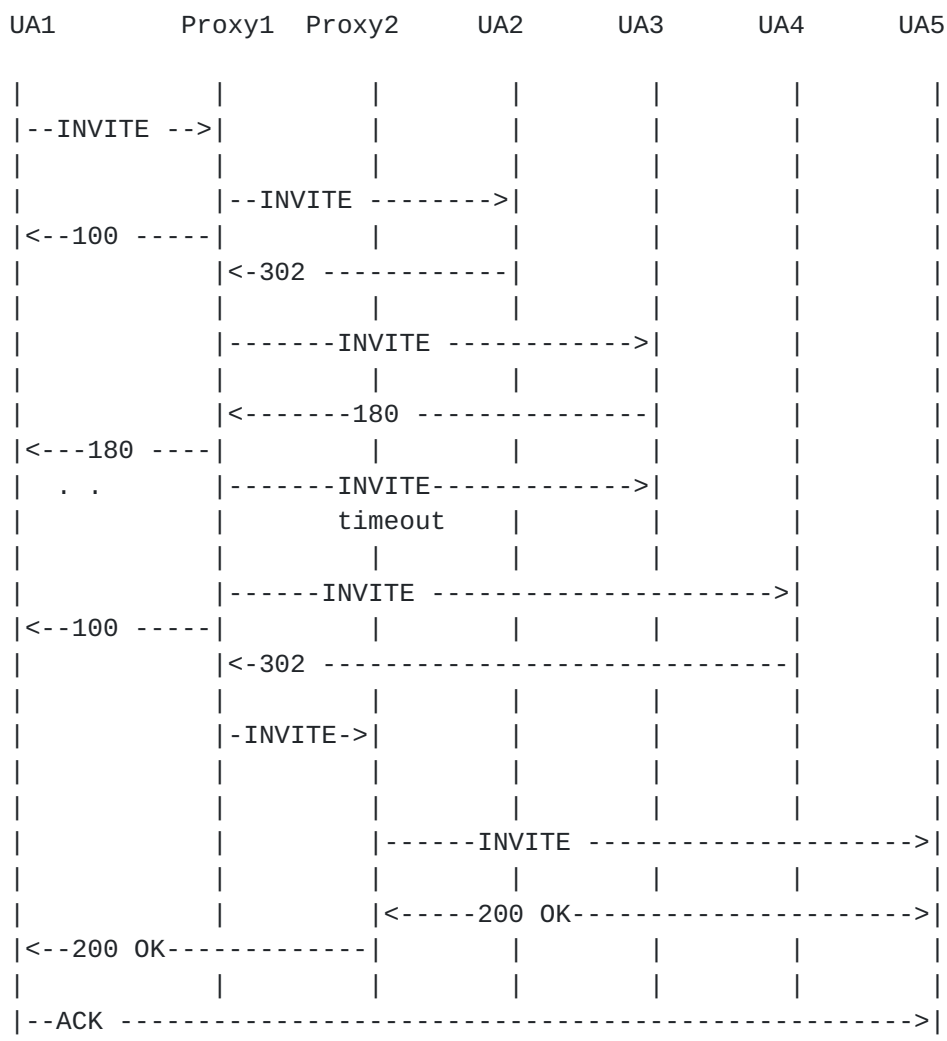


[Editor's Note: Need to detail the message flow.]

A.2 Sequential Forking (with Success)

This scenario highlights an example where the History-Info in the request is primarily of use in not retrying routes that have already been tried by another proxy. Note, that this is just an example and that there may be valid reasons why a Proxy would want to retry the routes and thus, this would like be a local proxy or even user specific policy.

UA 1 sends a call to "Bob" to proxy 1. Proxy 1 sequentially tries several places (UA2, UA3 and UA4) before retargeting the call to Proxy 2. Proxy 2, without the History-Info, would try several of the same places (UA3 and UA4)based upon registered contacts for "Bob", before completing at UA5. However, with the History-Info, Proxy 2 determines that UA3 and UA4 have already received the invite, thus the INVITE goes directly to UA5.



[Editor's Note: Need to add the details of the messages here.]

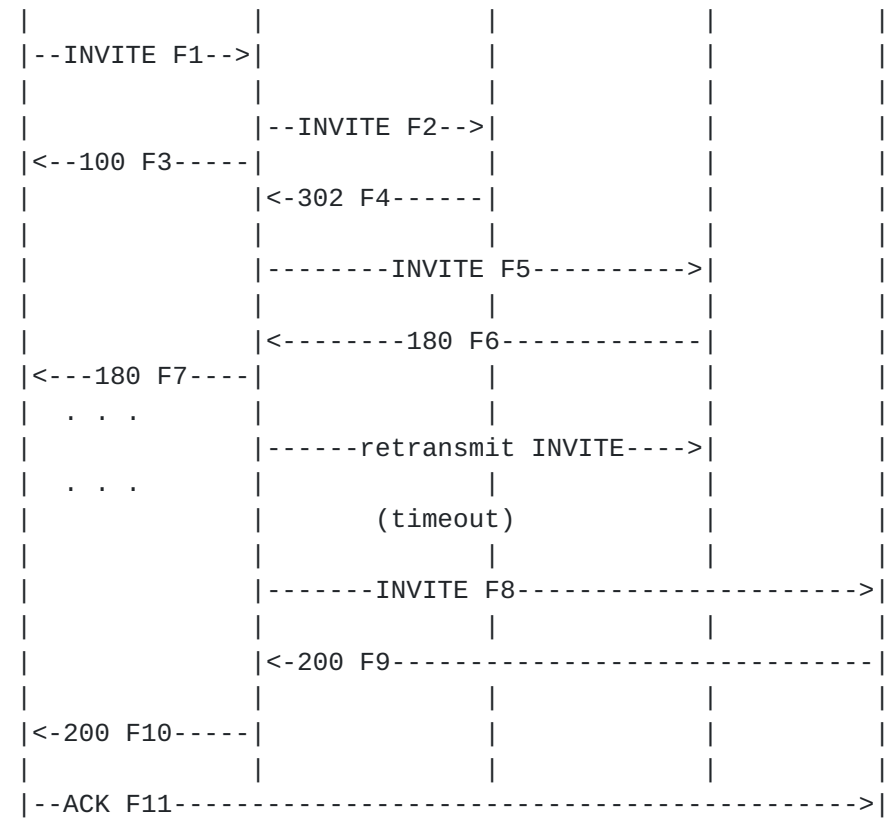
Appendix B Voicemail

This scenario highlights an example where the History-Info in the request is primarily of use by an edge service (e.g. Voicemail Server). It should be noted that this isn't intended to be a complete specification for this specific edge service as it is

quite likely that additional information is need by the edge service. History-Info is just one building block that this service makes use of.

UA 1 called UA A which had been forwarded to UA B which forwarded to a UA VM (voicemail server). Based upon the retargeted URIs and Reasons (and other information) in the INVITE, the VM server makes a policy decision about what mailbox to use, which greeting to play etc.

UA1 Proxy UA-A UA-B UA-VM



Message Details

```
INVITE F1   UA1->Proxy

INVITE sip:UserA@nortelnetworks.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:User1@here.com>
To: LittleGuy <sip:UserA@nortelnetworks.com>
Call-Id: 12345600@here.com
CSeq: 1 INVITE
Contact: BigGuy <sip:User1@here.com>
Content-Type: application/sdp
Content-Length: <appropriate value>

 v=0
    o=UserA 2890844526 2890844526 IN IP4 client.here.com
 s=Session SDP
 c=IN IP4 100.101.102.103
 t=0 0
 m=audio 49170 RTP/AVP 0
 a=rtpmap:0 PCMU/8000

 /*Client for UA1 prepares to receive data on port 49170
```

```
   from the network. */

   INVITE F2 Proxy->UA-A

   INVITE sip:UserA@ims.nortelnetworks.com SIP/2.0
   Via: SIP/2.0/UDPims.nortelnetworks.com:5060;branch=1
     Via: SIP/2.0/UDP here.com:5060
   Record-Route: <sip:UserA@nortelnetworks.com>
   From: BigGuy <sip:User1@here.com>
   To: LittleGuy <sip:UserA@nortelnetworks.com>
   Call-Id: 12345600@here.com
   CSeq: 1 INVITE
```

```
History-Info: <sip:UserA@ims.nortelnetworks.com>; index=1
Contact: BigGuy <sip:User1@here.com>
Content-Type: application/sdp
Content-Length: <appropriate value>

v=0
o=UserA 2890844526 2890844526 IN IP4 client.here.com
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

100 Trying F3 Proxy->UA1

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:User1@here.com>
To: LittleGuy <sip:UserA@nortelnetworks.com>
Call-Id: 12345600@here.com
CSeq: 1 INVITE
Content-Length: 0
```

302 Moved Temporarily F4 UserA->Proxy

```
SIP/2.0 302 Moved Temporarily
Via: SIP/2.0/UDP ims.nortelnetworks.com:5060;branch=1
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:User1@here.com>
To: LittleGuy <sip:UserA@nortelnetworks.com>;tag=3
Call-Id: 12345600@here.com
CSeq: 1 INVITE
Contact: <sip:UserB@nortelnetworks.com>
Content-Length: 0
```

INVITE F5 Proxy-> UA-B

```
INVITE sip:UserB@nortelnetworks.com SIP/2.0
Via: SIP/2.0/UDP ims.nortelnetworks.com:5060;branch=2
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:User1@here.com>
To: LittleGuy <sip:UserA@nortelnetworks.com>
Call-Id: 12345600@here.com
History-Info: <sip:UserA@ims.nortelnetworks.com>; index=1,
<sip:UserB@nortelnetworks.com?Reason=SIP; cause=302; text="Moved
Temporarily">;index=2
CSeq: 1 INVITE
```

```
Contact: BigGuy <sip:User1@here.com>
Content-Type: application/sdp
Content-Length: <appropriate value>

v=0
o=User1 2890844526 2890844526 IN IP4 client.here.com
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

```
180 Ringing F6  UA-B ->Proxy

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP there.com:5060
From: BigGuy <sip:User1@here.com>
To: LittleGuy <sip:UserA@nortelnetworks.com>;tag=5
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Content-Length: 0

180 Ringing F7  Proxy-> UA1

SIP/2.0 180 Ringing
SIP/2.0/UDP here.com:5060
From: BigGuy <sip:User1@here.com>
To: LittleGuy <sip:UserA@nortelnetworks.com>
Call-Id: 12345600@here.com
CSeq: 1 INVITE
Content-Length: 0

/* User B is not available. INVITE is sent multiple
times until it times out. */

  /* The proxy forwards the INVITE to UA-VM after adding the
additional History Information entry. */


INVITE F8  Proxy-> UA-VM

INVITE sip:VM@nortelnetworks.com SIP/2.0
Via: SIP/2.0/UDP ims.nortelnetworks.com:5060;branch=3
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:User1@here.com>
   To: LittleGuy <sip:UserA@nortelnetworks.com>
Call-Id: 12345600@here.com
History-Info: <sip:UserA@ims.nortelnetworks.com>;index=1,
<sip:UserB@nortelnetworks.com?Reason=SIP; cause=302; text="Moved
Temporarily">;index=2,
<sip:VM@nortelnetworks.com?Reason=SIP;cause=480;text="Temporarily
Unavailable">;index=3
CSeq: 1 INVITE
Contact: BigGuy <sip:User1@here.com>
Content-Type: application/sdp
Content-Length: <appropriate value>

v=0
o=User1 2890844526 2890844526 IN IP4 client.here.com
s=Session SDP
```

```
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000


200 OK F9

SIP/2.0 200 OK UA-VM->Proxy

Via: SIP/2.0/UDP ims.nortelnetworks.com:5060;branch=3
```

```
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:User1@here.com>
To: LittleGuy <sip:UserA@nortelnetworks.com>;tag=3
Call-Id: 12345600@here.com
CSeq: 1 INVITE
Contact: TheVoiceMail <sip:VM@nortelnetworks.com>
Content-Type: application/sdp
Content-Length: <appropriate value>

v=0
o=UserA 2890844527 2890844527 IN IP4 vm.nortelnetworks.com
s=Session SDP
c=IN IP4 110.111.112.114
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

200 OK F10 Proxy->UA1

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ims.nortelnetworks.com:5060;branch=3
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:User1@here.com>
To: LittleGuy <sip:UserA@nortelnetworks.com>;tag=3
Call-Id: 12345600@here.com
CSeq: 1 INVITE
Contact: TheVoiceMail <sip:VM@nortelnetworks.com>
Content-Type: application/sdp
Content-Length: <appropriate value>

v=0
o=UserA 2890844527 2890844527 IN IP4 vm.nortelnetworks.com
s=Session SDP
c=IN IP4 110.111.112.114
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

ACK F11 UA1-> UA-VM

```
ACK sip:VM@nortelnetworks.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:User1@here.com>
To: LittleGuy<sip:UserA@nortelnetworks.com>;tag=3
Call-Id: 12345600@here.com
CSeq: 1 ACK
Content-Length: 0
```

```
   /* RTP streams are established between UA1 and
   UA-VM. UA-VM starts announcement for UA1 */
```

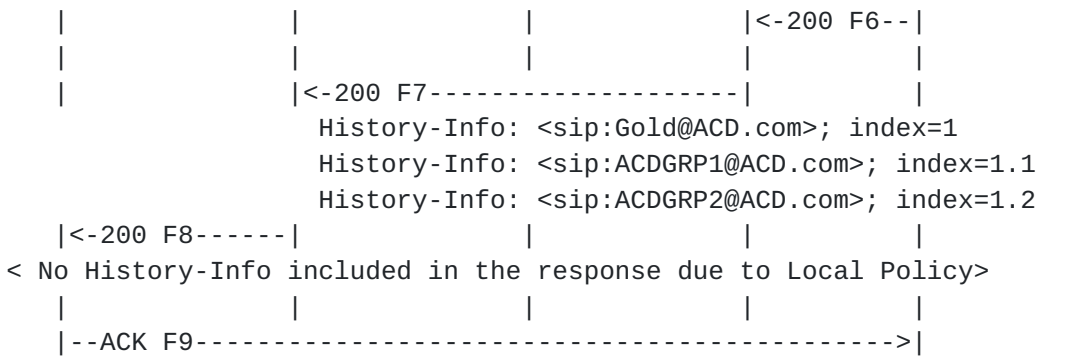
Appendix C Automatic Call Distribution Example

This scenario highlights an example of an Automatic Call Distribution service, where the agents are divided into groups based upon the type of customers they handle. In this example, the Gold customers are given higher priority than Silver customers, so a Gold call would get serviced even if all the agents servicing the

Gold group (ACDGRP1) were busy, by retargeting the request to the Silver Group. Upon receipt of the call at the agent assigned to handle the incoming call, based upon the History-Info in the message, the application at the agent can provide an indication that this is a Gold call, from how many groups it might have overflowed before reaching the agent, etc. thus can be handled appropriately by the agent.

For scenarios whereby calls might overflow from the Silver to the Gold, clearly the alternate group identification, internal routing or actual agent that handles the call SHOULD not be sent to UA1, thus for this scenario, one would expect that the Proxy would not support the sending of the History-Info in the response, even if requested by the calling UA.

As with the other examples, this is not prescriptive of how one would do this type of service but an example of a subset of processing that might be associated with such a service. In addition, this example is not addressing any aspects of Agent availability, which might also be done via a SIP interface.



Message Details

[To be completed]

Appendix D Solution options analysis

This section is included to capture some background analysis forming the basis for the solution proposed in this document. This section can be deleted from a subsequent version once the content of this document is sufficiently developed and well understood.

D.1 Optionality Requirements.

In many cases, it is anticipated that whether the history is added to the Request would be a local policy decision enforced by the specific application, thus no specific protocol element is needed. However, due to the capability being "optional" from the SIP protocol perspective, the impact to an application of not having the "Request History" must be described. For example, in a scenario where there is sequential forking and retargeting, some of the destinations previously tried could be retried. The impact of not having the "Request History" information for this sample application is that routing is inefficient. However, another scenario involving a voicemail application, the impact of not having the "Request History" information would be the service could not operate without having the information as to why the call was retargeted and the initial target for the call. Thus, the expectation would be that the policy in a system that intended to support this voicemail application would have to require the entities within its domain which are capable of retargeting to capture "Request History" information.

Thus, there are several aspects to the optionality requirement:

- Optionality with regards to whether the History Information is to be included in responses to the original Request.
- Optionality with regards to whether a particular retargeting entity records the History Information.
- Due to the Privacy requirement, the information MUST not be captured for Request URIs that have indicated a requirement for privacy.

The optionality mechanisms also depends upon whether the need for the "Request History" is based upon an end user based service (e.g. a GUI that provides the list of tried entities for an unsuccessful call setup, thus ensuring that the caller doesn't re-attempt an entity in that list or attendant services) or a network based service whose use of the "Request History" would likely be transparent to the UA (e.g. the Voicemail example).

The Supported header is the chosen mechanism for a UAC to indicate

that the information should be included in subsequent responses. Whether a server processing the request supports the mechanism would be based upon local policy for that domain.

D.2 Content-req

The Content-req specifies the following:
Retargeted-to-URI
Retargeted-from-URI
Reason

Chronological ordering

The following summarizes the solution considerations for each of these content requirements:

D.2.1 Is the Retargeted-to-URI required when it can be derived at the next hop, which would capture this as the Retargeted-from-URI for subsequent retargeting?

In a series of Request History Information, the Retargeted-to-URI becomes the Retargeted-from-URI for the next occurrence of retargeting, thus it would be possible in a scenario where the Request History functionality is supported by each of the retargeting entities to derive a complete set of Retargeted-to and Retargeted-from URIs from the sequence of History Information rather than including both Retargeted-to and Retargeting-from URI in each occurrence of History Information.

However, for the scenario where a particular proxy retargets, but local policy does not support the Request History Information, this approach could result in a potential loss of information. In addition, the support of the BACKWARDS-Req does require that the retargeted-to URI also be captured to ensure completeness of information (to the extent possible based on policies, privacy, etc.) in Responses.

Another option put forth was capture only one URI, but to actually capture the initial targeted-to-URI which then becomes a retargeted-from-URI when the request is retargeted, with the retargeted-to-URI captured as the next targeted-to-URI. This would require that a UA wanting to make use of History-Info in responses would actually capture the first targeted-to-URI. In addition, a proxy that supports History-Info would need to capture two targeted-to URIs IF one was not included in the initial Request. This processing could appear to be in conflict with the privacy and optionality requirements, however, since a Request-URI is only retargeted if it indicates a resource that a proxy is responsible for, this isn't an issue. What this does mean is that a request that is only forwarded by a proxy would NOT capture the retargeted-from URI, but could capture the retargeted-to URI.

D.2.2 Reason

The Reason header field [3] seems like a possible solution for carrying the Reason associated with the Retargeting. It is proposed to include this header as an optional escaped part of the targeted-to-URI.

D.2.3 Chronological ordering

The following were considered as options for satisfying this requirement:

- No explicit count/index. The Chronological ordering requirement should not require a specific protocol element if the History-Info entries are recommended to be added in the order they are generated and collected.

If a count/index is included, the following were considered as alternatives for maintaining the logical order of the parallel forking:

- o Indexing using a dot delimiter to indicate hops and forking (e.g. 1.1.1, 1.1.2 would indicate 2 hops with 2 retargeted nd URIs at the 2 hop.)
- o ABNF reflecting the nesting/hops (whether this is even feasible was not determined).
- o Allowing the same value for the count/index (i.e. not worrying about duplicates as the value indicates only relative order).
- o Including a count and an additional branch parameter for the forking (e.g. n=1, br=1.1, n=1, br=1.2).

Full Copyright Statement