Network Working Group
Request for Comments: 3608
Category: Standards Track

D. Willis
dynamicsoft Inc.
B. Hoeneisen
Switch

October 2003

Session Initiation Protocol (SIP) Extension Header Field for Service Route Discovery During Registration

Status of this Memo

Copyright Notice

Abstract

This document defines a Session Initiation Protocol (SIP) extension header field used in conjunction with responses to REGISTER requests to provide a mechanism by which a registrar may inform a registering user agent (UA) of a service route that the UA may use to request outbound services from the registrar's domain.

Table of Contents

1. Terminology

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 BCP 14, RFC 2119 [1].

2. Background

The Third Generation Partnership Project (3GPP) established a requirement for discovering home proxies during SIP registration and published this requirement in [6]. The 3GPP network dynamically assigns a home service proxy to each address-of-record (AOR). This assignment may occur in conjunction with a REGISTER operation, or out-of-band as needed to support call services when the address-of-record has no registrations. This home service proxy may provide both inbound (UA terminated) and outbound (UA originated) services.

In the inbound case, the Request-Uniform Resource Identifier (URI) of incoming SIP requests matches the address-of-record of a user associated with the home service proxy. The home service proxy then (in most cases) forwards the request to the registered contact address for that AOR. A mechanism for traversing required proxies between the home service proxy and the registered UA is presented in [4].

Outbound (UA originated) session cases raise another issue. Specifically, "How does the UA know which service proxy to use and how to get there?"

Several mechanisms were proposed in list discussions, including:

1. Configuration data in the UA. This raises questions of UA configuration management and updating, especially if proxy assignment is very dynamic, such as in load-balancing scenarios.

2. Use of some other protocol, such as HTTP, to get configuration data from a configuration server in the home network. While functional, this solution requires additional protocol engines, firewall complexity, operations overhead, and significant additional "over the air" traffic.

3. Use of lookup tables in the home network, as may be done for inbound requests in some 3G networks. This has a relatively high overhead in terms of database operations.

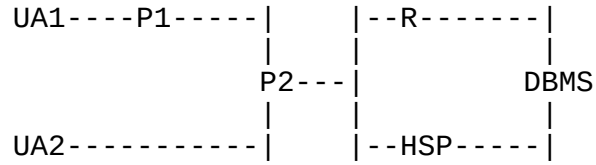
4. Returning a 302 response indicating the service proxy as a new contact, causing the upstream node processing the 302 (ostensibly the UA) to retransmit the request toward the service proxy. While this shares the database operation of the previous alternative, it does explicitly allow for caching the 302 response thereby potentially reducing the frequency and number of database operations.

5. Performing an operation equivalent to record-routing in a REGISTER transaction between the UA and the associated registrar, then storing that route in the UA and reusing it as a service route on future requests originating from the UA. While efficient, this constrains the service route for proxy operations to be congruent with the route taken by the REGISTER message.

6. Returning service route information as the value of a header field in the REGISTER response. While similar to the previous alternative, this approach grants the ability for the registrar to selectively apply knowledge about the topology of the home network in constructing the service route.

This document defines this final alternative: returning the service route information as a header field in the REGISTER response. This new header field indicates a "preloaded route" that the UA may wish to use if requesting services from the proxy network associated with the registrar generating the response.

Scenario



In this scenario, we have a "home network" containing routing proxy P2, registrar R, home service proxy HSP, and database DBMS used by both R and HSP. P2 represents the "edge" of the home network from a SIP perspective, and might be called an "edge proxy". UA1 is an external UA behind proxy P1. UA1 discovers P1 via Dynamic Host Configuration Protocol (DHCP) (this is just an example, and other mechanisms besides DHCP are possible). UA2 is another UA on the Internet, and does not use a default outbound proxy. We do not show Domain Name System (DNS) elements in this diagram, but will assume their reasonable availability in the discussion. The mission is for UA1 to discover HSP so that outbound requests from UA1 may be routed (at the discretion of UA1) through HSP, thereby receiving outbound services from HSP.

3. Discussion of Mechanism

UAs may include a Route header field in an initial request to force that request to visit and potentially be serviced by one or more proxies. Using such a route (called a "service route" or "preloaded route") allows a UA to request services from a specific home proxy or network of proxies. The open question is, "How may a UA discover what service route to use?"

This document defines a header field called "Service-Route" which can contain a route vector that, if used as discussed above, will direct requests through a specific sequence of proxies. A registrar may use a Service-Route header field to inform a UA of a service route that, if used by the UA, will provide services from a proxy or set of proxies associated with that registrar. The Service-Route header field may be included by a registrar in the response to a REGISTER request. Consequently, a registering UA learns of a service route that may be used to request services from the system it just registered with.

The routing established by the Service-Route mechanism applies only to requests originating in the user agent. That is, it applies only to UA originated requests, and not to requests terminated by that UA.

Simply put, the registrar generates a service route for the registering UA and returns it in the response to each successful REGISTER request. This service route has the form of a Route header field that the registering UA may use to send requests through the service proxy selected by the registrar. The UA would use this route by inserting it as a preloaded Route header field in requests originated by the UA intended for routing through the service proxy.

The mechanism by which the registrar constructs the header field value is specific to the local implementation and outside the scope of this document.

4. Applicability Statement

The Service-Route mechanism is applicable when:

1. The UA registers with a registrar.

2. The registrar has knowledge of a service proxy that should be used by the UA when requesting services from the domain of the registrar. This knowledge may be a result of dynamic assignment or some other mechanism outside the scope of this document.

3. The registrar(s) has/have sufficient knowledge of the network topology, policy, and situation such that a reasonable service route can be constructed.

4. The service route constructed by the registrar is the same for all contacts associated with a single address-of-record. This mechanism does not provide for contact-specific service routes.

5. Other mechanisms for proposing a service route to the UA are not available or are inappropriate for use within the specific environment.

Other methods may also be available by which a UA may be informed of a service route. Such alternative methods are outside the scope of this document. Discussion of why one might wish to assign a service route during registration or when it might be appropriate to do so is outside the scope of this document.

5. Syntax

The syntax for the Service-Route header field is:

```
Service-Route = "Service-Route" HCOLON sr-value *( COMMA sr-value)

sr-value = name-addr *( SEMI rr-param )
```

Note that the Service-Route header field values MUST conform to the syntax of a Route element as defined in [3]. As suggested therein, such values MUST include the loose-routing indicator parameter ";lr" for full compliance with [3].

The allowable usage of header fields is described in Tables 2 and 3 of [3]. The following additions to this table are needed for Service-Route.

Addition of Service-Route to SIP Table 3:

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG	PRA
Service-Route	2xx	ar	-	-	-	-	-	o	-

6. Usage

6.1. Procedures at the UA

The UA performs a registration as usual. The REGISTER response may contain a Service-Route header field. If so, the UA MAY store the value of the Service-Route header field in an association with the address-of-record for which the REGISTER transaction had registered a contact. If the UA supports multiple addresses-of-record, it may be able to store multiple service routes, one per address-of-record. If the UA refreshes the registration, the stored value of the Service-Route is updated according to the Service-Route header field of the latest 200 class response. If there is no Service-Route header field in the response, the UA clears any service route for that address-of-record previously stored by the UA. If the re-registration request is refused or if an existing registration expires and the UA chooses not to re-register, the UA SHOULD discard any stored service route for that address-of-record.

The UA MAY choose to exercise a service route for future requests associated with a given address-of-record for which a service route is known. If so, it uses the content of the Service-Route header field as a preloaded Route header field in outgoing initial requests [3]. The UA MUST preserve the order, in case there is more than one Service-Route header field or header field value.

Loose routes may interact with routing policy in interesting ways. The specifics of how the service route set integrates with any locally required default route and local policy are implementation dependent. For example, some devices will use locally-configured explicit loose routing to reach a next-hop proxy, and others will use a default outbound-proxy routing rule. However, for the result to function, the combination MUST provide valid routing in the local environment. In general, the service route set is appended to any locally configured route needed to egress the access proxy chain. Systems designers must match the service routing policy of their nodes with the basic SIP routing policy in order to get a workable system.

6.2. Procedures at the Proxy

The Service-Route header field is generally treated like any other unknown header field by intermediate proxies. They simply forward it on towards the destination. Note that, as usual, intermediate proxies that need to be traversed by future requests within a dialog may record-route. Proxies should not assume that they will be traversed by future requests in a dialog simply because they appear in the Service-Route header field.

There is a question of whether proxies processing a REGISTER response may add themselves to the route set in the Service-Route header field. While this would enable dynamic construction of service routes, it has two significant problems. The first is one of transparency, as seen by the registrar: Intermediate proxies could add themselves without the knowledge or consent of the registrar. The second problem is interaction with end-to-end security. If the registrar uses S/MIME techniques to protect the REGISTER response, such additions would be visible to the UA as "man in the middle" alterations in the response. Consequently, intermediate proxies SHOULD NOT alter the value of Service-Route in REGISTER responses, and if they do, the UA MUST NOT be required to accept the alteration.

Additional considerations apply if a proxy is "dual homed", meaning connected to two (or more) different networks such that requests are received on one interface and proxied out through another network interface. Proxies implementing multi-homing precisely as documented in [3] record-route a request with the sending interface. When processing the reply, they replace the Record-Route header field value that represents the interface onto which they proxied the request with a new value that represents the interface onto which they will proxy the response. Consequently, the route vector seen at the User Agent Server (UAS) is not the exact inverse of the route vector seen at the User Agent Client (UAC). While in itself harmless, this complicates matters for nodes that use the recorded route vector (or recorded Path vector as per [4]) in the determination of a service route for future use.

Instead of following the procedure in [3], proxies used with Service-Route that are inserting Record-Route or Path header field values SHOULD record not one but two route values when processing the request. The first value recorded indicates the receiving interface, and the second indicates the sending interface. When processing the response, no modification of the recorded route is required. This optimization provides for fully invertible routes that can be effectively used in construction of service routes.

6.3. Procedures at the Registrar

When a registrar receives a successful REGISTER request, it MAY choose to return one or more Service-Route header field(s) in the 200 class response. The determination(s) of whether to include these header fields(s) into the 200 class response and what value(s) to insert are a matter of local policy and outside the scope of this document.

Having inserted a Service-Route header field or fields, the registrar returns the 200 class response to the UA in accordance with standard procedures.

A REGISTER operation performing a Fetching Bindings (i.e., no Contact header field is present in the request) SHOULD return the same value of Service-Route as returned in the corresponding previous REGISTER response for the address-of-record in question. In some cases, the Service-Route may be dynamically calculated by the registrar rather than stored, and the decision as to whether this route should be recalculated in the event of a Fetching Bindings operation is left to the implementation.

Note: A Fetching Bindings operation could be used by the UA to recover a lost value of Service-Route. Alternatively, a UA in this situation could just re-REGISTER.

Certain network topologies MAY require a specific proxy (e.g., firewall proxy) to be traversed before the home service proxy. Thus, a registrar with specific knowledge of the network topology MAY return more than one Service-Route header field or element in the 200 class response; the order is specified as top-down, meaning the topmost Service-Route entry will be visited first. Such constructions are implementation specific and outside the scope of this document.

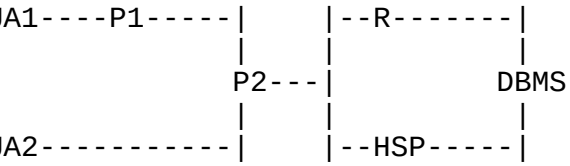
In general, the Service-Route header field contains references to elements strictly within the administrative domain of the registrar and home service proxy. For example, consider a case where a user leaves the "home" network and roams into a "visited" network. The registrar cannot be assumed to have knowledge of the topology of the visited network, so the Service-Route it returns contains elements only within the home network.

Note that the inserted Service-Route element(s) MUST conform to the syntax of a Route element as defined in [3]. As suggested therein, such route elements MUST include the loose-routing indicator parameter ";lr" for full compliance with [3].

6.4. Examples of Usage

We present an example in the context of the scenario presented in the Background section earlier in this document. The network diagram is replicated below:

Scenario



6.4.1. Example of Mechanism in REGISTER Transaction

This example shows the message sequence for user agent UA1 registering to HOME.EXAMPLE.COM using registrar R. R returns a Service-Route indicating that UA1 may use home service proxy HSP.HOME.EXAMPLE.COM to receive outbound services from HOME.EXAMPLE.COM.

Please note that some header fields (e.g., Content-Length) and session descriptions are omitted to provide a shorter and hopefully more readable presentation.

Message sequence for REGISTER returning Service-Route:

F1 Register UA1 -> P1

```
REGISTER sip:HOME.EXAMPLE.COM SIP/2.0
Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKcR1ntRAp
To: Lawyer <sip:UA1@HOME.EXAMPLE.COM>
From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=981211
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
 . . .
```

F2 Register P1 -> P2

```
 REGISTER sip:HOME.EXAMPLE.COM SIP/2.0
 Via: SIP/2.0/UDP P1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKlJuB1mcr
 Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKcR1ntRAp
 To: Lawyer <sip:UA1@HOME.EXAMPLE.COM>
 From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=981211
 Call-ID: 843817637684230@998sdasdh09
 CSeq: 1826 REGISTER
 Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
  . . .
```

F3 Register P2 -> R

```
 REGISTER sip:HOME.EXAMPLE.COM SIP/2.0
 Via: SIP/2.0/UDP P2.HOME.EXAMPLE.COM:5060;branch=z9hG4bKvE0R2l07o2b6T
 Via: SIP/2.0/UDP P1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKlJuB1mcr
 Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKcR1ntRAp
 To: Lawyer <sip:UA1@HOME.EXAMPLE.COM>
 From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=981211
 Call-ID: 843817637684230@998sdasdh09
 CSeq: 1826 REGISTER
 Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
  . . .
```

F4 R executes Register

```
 R Stores:
 For <sip:UA1@HOME.EXAMPLE.COM>
 Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
```

F5 R calculates Service Route

```
 In this example, R is statically configured to reference HSP as a
 service route, and R also knows that P2 is used as the provider
 edge proxy, so:

 Service-Route: <sip:P2.HOME.EXAMPLE.COM;lr>,
                <sip:HSP.HOME.EXAMPLE.COM;lr>
```

F6 Register Response r -> P2

```
 SIP/2.0 200 OK
 Via: SIP/2.0/UDP P2.HOME.EXAMPLE.COM:5060;branch=z9hG4bKvE0R2l07o2b6T
 Via: SIP/2.0/UDP P1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKlJuB1mcr
 Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKcR1ntRAp
 To: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=87654
 From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=981211
 Call-ID: 843817637684230@998sdasdh09
 CSeq: 1826 REGISTER
 Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
 Service-Route: <sip:P2.HOME.EXAMPLE.COM;lr>,
                <sip:HSP.HOME.EXAMPLE.COM;lr>
  . . .
```

F7 Register Response P2 -> P1

```
 SIP/2.0 200 OK
 Via: SIP/2.0/UDP P1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKlJuB1mcr
 Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKcR1ntRAp
 To: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=87654
 From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=981211
 Call-ID: 843817637684230@998sdasdh09
 CSeq: 1826 REGISTER
 Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
 Service-Route: <sip:P2.HOME.EXAMPLE.COM;lr>,
                <sip:HSP.HOME.EXAMPLE.COM;lr>
  . . .
```

F8 Register Response P1 -> UA1

```
 SIP/2.0 200 OK
 Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKcR1ntRAp
 To: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=87654
 From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=981211
 Call-ID: 843817637684230@998sdasdh09
 CSeq: 1826 REGISTER
 Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
 Service-Route: <sip:P2.HOME.EXAMPLE.COM;lr>,
                <sip:HSP.HOME.EXAMPLE.COM;lr>
  . . .
```

F9 UA1 stores service route for UA1@HOME.EXAMPLE.COM

6.4.2. Example of Mechanism in INVITE Transaction

This example shows the message sequence for an INVITE transaction originating from UA1 eventually arriving at UA2 using outbound services from HOME.EXAMPLE.COM. UA1 has previously registered with HOME.EXAMPLE.COM and been informed of a service route through HSP.HOME.EXAMPLE.COM. The service being provided by HOME.EXAMPLE.COM is a "logging" service, which provides a record of the call for UA1's use (perhaps the user of UA1 is an attorney who bills for calls to customers).

Note that in this example UA1 and UA2 are assumed to be registered with the same network (HOME.EXAMPLE.COM). This does not generally need to be the case to use the herein described service route mechanism.

Message sequence for INVITE using Service-Route:

F1 Invite UA1 -> P1

```
INVITE sip:UA2@HOME.EXAMPLE.COM SIP/2.0
Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKnashds7
To: Customer <sip:UA2@HOME.EXAMPLE.COM>
From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=456248
Call-ID: 38615183343@s1i1l2j6u
CSeq: 18 INVITE
Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
Route: <sip:P2.HOME.EXAMPLE.COM;lr>,
       <sip:HSP.HOME.EXAMPLE.COM;lr>
 . . .
```

Note: P1 is selected using the "outbound proxy" rule in UA1.

F2 Invite P1 -> P2

```
INVITE sip:UA2@HOME.EXAMPLE.COM SIP/2.0
Via: SIP/2.0/UDP P1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bK34ghi7ab04
Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKnashds7
To: Customer <sip:UA2@HOME.EXAMPLE.COM>
From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=456248
Call-ID: 38615183343@s1i1l2j6u
CSeq: 18 INVITE
Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
Record-Route: <sip:P1.VISITED.EXAMPLE.ORG;lr>
Route: <sip:P2.HOME.EXAMPLE.COM;lr>,
       <sip:HSP.HOME.EXAMPLE.COM;lr>
 . . .
```

Note: P1 has added itself to the Record Route.

F3 Invite P2 -> HSP

```
INVITE sip:UA2@HOME.EXAMPLE.COM SIP/2.0
Via: SIP/2.0/UDP P2.HOME.EXAMPLE.COM:5060;branch=z9hG4bKiokioukju908
Via: SIP/2.0/UDP P1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bK34ghi7ab04
Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKnashds7
To: Customer <sip:UA2@HOME.EXAMPLE.COM>
From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=456248
Call-ID: 38615183343@s1i1l2j6u
CSeq: 18 INVITE
Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
Record-Route: <sip:P2.HOME.EXAMPLE.COM;lr>
Record-Route: <sip:P1.VISITED.EXAMPLE.ORG;lr>
Route: <sip:HSP.HOME.EXAMPLE.COM;lr>
 . . .
```

Note: HSP is selected using a DNS lookup for HSP within HOME.EXAMPLE.COM.
P2 has added itself to the Record-Route.
P2 has removed itself from the Route.

F4 HSP executes service

HSP identifies the service to be executed from UA1’s stored profile. The specifics of this are outside the scope of this document. For this example HSP writes a record to "Lawyer’s log book", then looks up the AOR "sip:UA2@HOME.EXAMPLE.COM" and discovers that the current contact for UA2 is at host UAADDR2.HOME.EXAMPLE.COM. This will be the Request-URI of the next-hop INVITE.

F5 Invite HSP -> P2

```
INVITE sip:UA2@UAADDR2.HOME.EXAMPLE.COM SIP/2.0
Via: SIP/2.0/USP HSP.HOME.EXAMPLE.COM:5060;branch=z9hG4bKHSP10120323
Via: SIP/2.0/UDP P2.HOME.EXAMPLE.COM:5060;branch=z9hG4bKiokioukju908
Via: SIP/2.0/UDP P1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bK34ghi7ab04
Via: SIP/2.0/UDP UADDR1.VISITED.EXAMPLE.ORG:5060;branch=z9hG4bKnashds7
To: Customer <sip:UA2@HOME.EXAMPLE.COM>
From: Lawyer <sip:UA1@HOME.EXAMPLE.COM>;tag=456248
Call-ID: 38615183343@s1i1l2j6u
CSeq: 18 INVITE
Contact: <sip:UA1@UADDR1.VISITED.EXAMPLE.ORG>
Record-Route: <sip:HSP.HOME.EXAMPLE.COM;lr>
Record-Route: <sip:P2.HOME.EXAMPLE.COM;lr>
Record-Route: <sip:P1.VISITED.EXAMPLE.ORG;lr>
```

. . .

Note: P2 selected by outbound proxy rule on HSP.
HSP has removed itself from the Route.

INVITE propagates toward UA2 as usual.

7. Security Considerations

It is possible for proxies between the UA and the registrar during the REGISTER transaction to modify the value of Service-Route returned by the registrar, or to insert a Service-Route even when one was not returned by the registrar. The consequence of such an attack is that future requests made by the UA using the service route might be diverted to or through a node other than would normally be visited. It is also possible for proxies on the INVITE path to execute many different attacks. It is therefore desirable to apply transitive mutual authentication using sips: or other available mechanisms in order to prevent such attacks.

The "sips:" URI as defined in [3] defines a mechanism by which a UA may request transport-level message integrity and mutual authentication. Since there is no requirement for proxies to modify messages, S/MIME signed bodies may be used to provide end-to-end protection for the returned value.

Systems using Service-Route SHOULD provide hop-by-hop message integrity and mutual authentication. UAs SHOULD request this support by using a "sips:" URI. Registrars returning a Service-Route MUST implement end-to-end protection using S/MIME and SHOULD use S/MIME to protect all such responses. UAs receiving Service-Route SHOULD authenticate attached S/MIME bodies if present.

8. IANA Considerations

This document defines the SIP extension header field "Service-Route" which has been included in the registry of SIP header fields defined in [3]. The change process for SIP, [5] mandates that general SIP extension header fields be defined by a standards-track RFC. This document provides the required definition.

The following is the registration for the Service-Route header field:

RFC Number: RFC 3608

Header Field Name: Service-Route

Compact Form: none

9. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Postel, J. and J. Reynolds, "Instructions to RFC Authors", RFC 2223, October 1997.

[3] 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.

[4] Willis, D. and B. Hoeneisen, "Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts", RFC 3327, December 2002.

[5] Mankin, A., Bradner, S., Mahy, R., Willis, D., Ott, J. and B. Rosen, "Change Process for the Session Initiation Protocol (SIP)", BCP 67, RFC 3427, December 2002.

10. Informative References

[6] Garcia-Martin, M., "3rd-Generation Partnership Project (3GPP) Release 5 requirements on the Session Initiation Protocol (SIP)", Work in Progress, October 2002.

11. Intellectual Property Statement

12. Authors' Addresses

Dean Willis
dynamicsoft Inc.
3100 Independence Parkway
#311-164
Plano, TX 75075
US

Phone: +1 972 473 5455
EMail: dean.willis@softarmor.com

Bernie Hoeneisen
Switch
Limmatquai 138
CH-8001 Zuerich
Switzerland

Phone: +41 1 268 1515
EMail: hoeneisen@switch.ch, b.hoeneisen@ieee.org
URI: http://www.switch.ch/

13. Full Copyright Statement

Acknowledgement

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