

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 8, 2007

T. Hardie
Qualcomm, Inc.
A. Newton
SunRocket
H. Schulzrinne
Columbia U.
H. Tschofenig
Siemens
September 4, 2006

LoST: A Location-to-Service Translation Protocol
draft-ietf-ecrit-lost-01.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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 March 8, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document describes an XML-based protocol for mapping service identifiers and geospatial or civic location information to service contact URIs. In particular, it can be used to determine the location-appropriate PSAP for emergency services.

Table of Contents

1.	Introduction	4
2.	Requirements Notation	6
3.	Usage	7
4.	Resolving Service URNs Using LoST	8
5.	Query	9
5.1.	Location Information Element	9
5.2.	Service Element	9
5.3.	Validate Attribute	9
5.4.	Query Message Examples	9
6.	Response	11
6.1.	Uniform Resource Identifiers (URI) Element	11
6.2.	Display Name Element	11
6.3.	Service Element	11
6.4.	ServiceBoundary Element	12
6.5.	ServiceNumber Element	12
6.6.	TimeToLive Attribute	12
6.7.	Validation Element	12
6.8.	Response Message Examples	12
7.	List Services Query and Response	15
7.1.	List Service Query	15
7.2.	List Service Response	15
8.	Status Code Definitions	17
8.1.	Informational 1xx	17
8.2.	Successful 2xx	17
8.2.1.	200 OK	17
8.2.2.	201 Service Substitution	17
8.3.	Redirection 3xx	17
8.3.1.	301 Move Permanently	17
8.3.2.	302 Moved Temporarily	18
8.3.3.	Example	18
8.4.	Client Error 4xx	18
8.4.1.	400 Bad Request	18
8.4.2.	403 Forbidden	18
8.4.3.	404 Not Found	18
8.4.4.	414 Location Error	18
8.4.5.	Example	18
8.5.	Server Error 5xx	20
8.5.1.	500 Server Internal Error	20

8.5.2.	501 Service Not Implemented	20
8.5.3.	504 Server Time-Out	21
8.5.4.	Example	21
9.	LoST Transport	22
10.	LoST Uniform Resource Locators	23
11.	Example	24
12.	Deployment Methods	26
13.	Relax NG Schema	28
14.	Internationalization Considerations	33
15.	IANA Considerations	34
15.1.	Content-type registration for 'application/lost+xml' . . .	34
15.2.	LoST Relax NG Schema Registration	35
15.3.	LoST Namespace Registration	36
15.4.	Registration Template	36
16.	Security Considerations	38
17.	Acknowledgments	39
18.	Open Issues	40
19.	References	41
19.1.	Normative References	41
19.2.	Informative References	42
Appendix A.	Non-Normative RELAX NG Schema in XML Syntax	43
	Authors' Addresses	51
	Intellectual Property and Copyright Statements	52

1. Introduction

This document describes a protocol for mapping a service identifier [6] and location information compatible with PIDF-LO [11] to one or more service contact URIs. Example contact URI schemes include sip, xmpp, and tel. While the initial focus is on providing mapping functions for emergency services, it is likely that the protocol is applicable to any service URN. For example, in the United States, the "2-1-1" and "3-1-1" services follow a similar location-to-service behavior as emergency services.

This document names this protocol usage "LoST" for Location-to-Service Translation Protocol. The features of LoST are:

- o Supports queries using civic as well as geospatial location information.
- o Support for recursive and iterative resolution.
- o Support for address validation.
- o A hierarchical deployment of mapping servers is independent of civic location labels.
- o Indication of errors in the location data to facilitate debugging and proper user feedback while simultaneously providing best-effort answers.
- o Mapping can be based on either civic or geospatial location information, with uniform protocol treatment of both.
- o Support for overlapping service regions.
- o Satisfies the requirements [5] for mapping protocols.
- o Minimizes round trips by caching individual mappings and by supporting return of coverage regions ("hinting").
- o Facilitates reuse of Transport Layer Security (TLS).

This document focuses on the description of the protocol between the mapping client (seeker or resolver) and the mapping server (resolver or other servers). The relationship between other functions, such as discovery of mapping servers, data replication and the overall mapping server architecture in general, will be described in a separate document. [20] is a first attempt to describe such a mapping server architecture.

The high-level protocol operation can be described as follows:

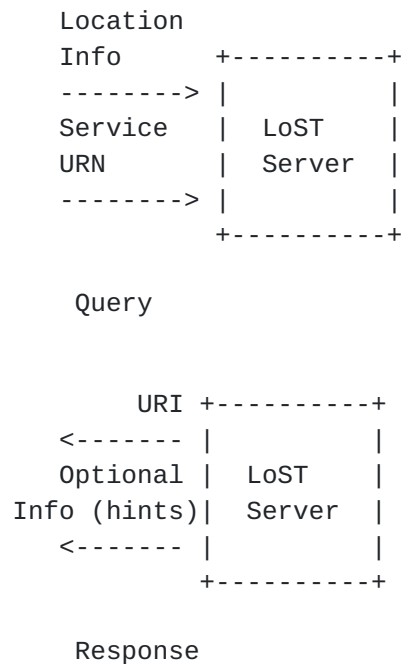


Figure 1: Overview

The query message carries location information and a service identifier encoded as a Uniform Resource Name (URN) (see [6]) from the LoST client to the LoST server. The LoST server uses its database to map the input values to a Uniform Resource Identifiers (URI) and returns it including optional information such as hints about the service boundary in a response message back to the LoST client.

2. Requirements Notation

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 [\[3\]](#).

3. Usage

The client queries a server, indicating the desired service and location information. If the query succeeds, the server returns a result that includes one or more URIs for reaching the appropriate service for the location indicated. Depending on the query, the result may contain a service boundary where the same mapping would apply, a reference to another server to which the client should send a query, or an error messages indicating problems. The combination of these components are left to the needs and policy of the jurisdiction where the server is being operated.

The client may perform the mapping at any time. Among the common triggers for mapping are:

1. When the client starts up and/or attaches to a new network location.
2. When the client detects that its location has changed sufficiently that it is outside the bounds of the region returned in an earlier query.
3. When cached mapping information has expired.
4. When calling for a particular service. During such calls, a client may want to request a short response that contains only the mapping data, omitting service boundary information.

Cached answers are expected to be used by clients only after failing to accomplish a location-to-URI mapping at call time. Cache entries may expire according to their time-to-live value, or they may become invalid if the location of the caller's device moves outside the boundary limits of the cache entry. Boundaries for cache entries may be set in both geospatial and civic terms.

4. Resolving Service URNs Using LoST

If a LoST URL contains a host name rather than an IP address, clients need to perform a U-NAPTR [\[17\]](#) lookup to obtain a DNS A record and IP address. These records map the 'host' part of the LoST URL to one or more URLs indicating the protocol to carry the LoST request. In this document, only the HTTP and HTTPS URL schemes are defined. Note that the HTTP URL can be any valid HTTP URL, including those containing path elements.

Here is an example:

example.com.

```
IN NAPTR 100 10 "u" "LoST:https"
    "!*!.!https://lostserver.example.com/secure!" ""
```

```
IN NAPTR 200 10 "u" "LoST:http"
    "!*!.!http://lostserver.example.com!" ""
```


5. Query

LoST provides the ability to use civic or geospatial location information in the query message. In addition to location information the query also contains a service identifier. An optional parameter might furthermore request the LoST server to validate location information.

5.1. Location Information Element

LoST supports a query using geospatial and civic location information using the `<findServiceByLocation>` query. Geospatial location information uses GML format [10] and civic location information utilizes the format defined in [16]. This document does not define location formats.

5.2. Service Element

The type of service desired is specified by the `<service>` element. The (emergency) service identifiers listed in the registry established with [6] will be used in this document.

The `<service>` element is a mandatory element. In case the database at the LoST server does not provide service for the specific geographical region the LoST server has various choices with regard to the response:

- o It can send an error response.
- o It can map one service to another one, if appropriate, and return a different service identifier as described in [Section 6.3](#).
- o It can populate the URIs of one service to another service.

The operation of the LoST server is largely a policy issue. No behavior is mandated in this document. Guidelines for operating a LoST server for emergency services is provided in [21].

5.3. Validate Attribute

The 'validate' attribute implements the validation behavior described in [5].

5.4. Query Message Examples

This section shows an example of a query message providing geospatial and civic location information.


```
<?xml version="1.0" encoding="UTF-8"?>
<findServiceByLocation
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  validate="false" operation="recursive">
  <locationInfo>
    <p2:Point id="point1" srsName="epsg:4326">
      <p2:coordinates>37:46:30N 122:25:10W</p2:coordinates>
    </p2:Point>
  </locationInfo>
  <service>urn:service:sos.police</service>
</findServiceByLocation>
```

Figure 3: Query Message Example using Geospatial Location Information

The example above shows a query using geospatial location information with no validation required and asking for the 'urn:service:sos.police' service.

```
<?xml version="1.0" encoding="UTF-8"?>
<findServiceByLocation xmlns="urn:ietf:params:xml:ns:lost1"
  validate="false" operation="recursive">
  <locationInfo>
    <civicLocation>
      <country>Germany</country>
      <A1>Bavaria</A1>
      <A3>Munich</A3>
      <A6>Neu Perlach</A6>
      <HNO>96</HNO>
      <PC>81675</PC>
    </civicLocation>
  </locationInfo>
  <service>urn:service:sos.police</service>
</findServiceByLocation>
```

Figure 4: Query Message Example using Civic Location Information

The example above shows a query using a civic location in Munich asking for the 'urn:service:sos.police' service. The query indicates that validation is not desired and the query has to be executed recursively.

6. Response

A response message might either contain civic or geospatial location information depending on the type of the query. If the findServiceByLocation query message contained civic location information then the <serviceBoundary> element of the response message will also contain civic information. If the findServiceByLocation query message contained geospatial location information then the <serviceBoundary> element of the response message will contain a GML polygon. More information about the <serviceBoundary> element can be found at [Section 6.4](#).

6.1. Uniform Resource Identifiers (URI) Element

Each <uri> element contains an appropriate contact URI for the service for which mapping was requested. <uri> elements are of type xs:anyURI. In the emergency service context operators are strongly discouraged from using relative URIs, even though these are permitted by the type.

6.2. Display Name Element

Each <displayName> element contains a string that is suitable for display. <displayName> elements are of type "text" that is suitable for internationalized human-readable text.

6.3. Service Element

The <service> element is an optional element in the response message. The (emergency) service identifiers listed in the registry established with [\[6\]](#) will be used in this document. If the service that was requested by the LoST client is not available for a particular location then the server MAY return an alternate service. If it does so, it MUST indicate the actual service returned (i.e., its service URN). Alternatively, the LoST server MAY return an error response indicating that the requested service is not available.

The following example illustrates the main idea. If there is a region that only understands the 'urn:service:sos' service and not 'urn:service:sos.fire', 'urn:service:sos.ambulance', and 'urn:service:sos.police'. If a LoST client asks for the 'urn:service:sos.fire' service then the LoST server could, depending on the local policy at the LoST server, return:

1. 'urn:service:sos', or
2. 'urn:service:sos.fire' with the values of 'urn:service:sos' being populated to 'urn:service:sos.fire', or

3. an error message

In case of (1) the <service> element carries the value of 'urn:service:sos'.

6.4. ServiceBoundary Element

Each <serviceBoundary> element contains either one or more civic location elements derived from the GeoPriv civic address schema or a GML-based polygon.

The <serviceBoundary> element indicates where the same query would yield to the same response, i.e., it provides information about the service boundary.

6.5. ServiceNumber Element

TBD: This element contains the (emergency) service number, which is a string of digits used to reach the (emergency) service.

6.6. TimeToLive Attribute

Each timeToLive attribute is a positive integer, expressing the validity period of the response in seconds. The LoST client MUST NOT consider the returned location current after the expiration of the validity period.

6.7. Validation Element

The <validation> element contains a string that is composed of concatenated tokens separated by a whitespace. These tokens refer to the civic location labels used in child elements of the <civicAddress> element from the request that have been recognized as valid by the server.

The following code snippet indicates that the civic address labels 'country', 'A1', 'A3', 'A6', 'PC' have been validated by the LoST server.

```
<validation>country A1 A3 A6 PC</validation>
```

6.8. Response Message Examples

This section shows an example of a query message providing geospatial and civic location information.


```
<?xml version="1.0" encoding="UTF-8"?>
<response
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml" >
  <result status="200" message="OK" xml:lang="en" timeToLive="1000">
    <displayName xml:lang="en">
      New York City Police Department
    </displayName>
    <service>urn:service:sos.police</service>
    <serviceBoundary>
      <p2:Polygon srsName="urn:ogc:def::crs:EPSG::4326">
        <p2:exterior>
          <p2:LinearRing>
            <p2:pos>37.775 -122.4194</p2:pos>
            <p2:pos>37.555 -122.4194</p2:pos>
            <p2:pos>37.555 -122.4264</p2:pos>
            <p2:pos>37.775 -122.4264</p2:pos>
            <p2:pos>37.775 -122.4194</p2:pos>
          </p2:LinearRing>
        </p2:exterior>
      </p2:Polygon>
    </serviceBoundary>
    <uri>sip:nypd@example.com</uri>
    <uri>xmpp:nypd@example.com</uri>
    <serviceNumber>911</serviceNumber>
  </result>
</response>
```

Figure 6: Response Message Example using Geospatial Location Service Boundary Hints

This example shows a response with two URIs for the previously queried service URN. Information about the service boundary is provided by a GML polygon. The <serviceNumber> element indicates the valid service number for the expressed location and service URN.


```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:ietf:params:xml:ns:lost1">
  <result status="200" timeToLive="10000">
    <displayName xml:lang="de">Munich Police Department</displayName>
    <service>urn:service:sos.police</service>
    <serviceBoundary>
      <civicLocation>
        <country>Germany</country>
        <A1>Bavaria</A1>
        <A3>Munich</A3>
        <PC>81675</PC>
      </civicLocation>
    </serviceBoundary>
    <uri>sip:munich-police@example.com</uri>
    <uri>xmpp:munich-police@example.com</uri>
    <service-number>110</service-number>
  </result>
</response>
```

Figure 7: Response Message Example providing Civic Location Service Boundary Hints

This example shows a response that returns two URIs (one for SIP and another one for XMPP), a distring that indicates the valid distring for the location provided in the query, a hint about the service boundary in the `<serviceBoundary>` element and information about the validated civic address fields. The `timeToLive` attribute indicates that the returned information can be cached for 10000 seconds and provides a `*<displayName>` element with additional, textual information about the returned information.

7. List Services Query and Response

7.1. List Service Query

This subsection describes a mechanism that offers the LoST client to query for available service identifiers supported by the LoST server. The listServices query MUST carry the <locationInfo> and the <service> element. The LoST server MUST return only immediate child elements of the service identifier specified in the <service> element of the listServices query available for the provided location information.

```
<?xml version="1.0" encoding="UTF-8"?>
<listServices
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  operation="false">

  <locationInfo>
    <p2:Point id="point1" srsName="epsg:4326">
      <p2:coordinates>37:46:30N 122:25:10W</p2:coordinates>
    </p2:Point>
  </locationInfo>
  <service>urn:service:sos</service>
</listServices>
```

Figure 8: Example for a List Service Query

This listService query aims to query the immediate child elements of the 'urn:service:sos' URN.

7.2. List Service Response

This subsection describes the response message that provides the LoST client with the list of immediate child service identifiers based on the service identifier provided by LoST client with respect to the location information provided in the listService query.

The following example shows the response to the listServices query example of Figure 8 listing the available services offered by the LoST server starting with 'urn:service:sos.ambulance' and finishing with 'urn:service:sos.suicide'.


```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:ietf:params:xml:ns:lost1">
  <serviceList status="200" message="OK" xml:lang="en">
    urn:service:sos.ambulance
    urn:service:sos.animal-control
    urn:service:sos.fire
    urn:service:sos.gas
    urn:service:sos.mountain
    urn:service:sos.marine
    urn:service:sos.physician
    urn:service:sos.poison
    urn:service:sos.police
    urn:service:sos.suicide
  </serviceList>
</response>
```

Figure 9: Example for the Response to a List Service Query

8. Status Code Definitions

Each response contains a <status> element that conveys a numeric status code and a reason phrase indicating the success or failure of the response. The appearance of other elements in the response depends on the status code. Hence, different elements are used for groups of status codes.

Status codes always have three digits; the list of status codes is meant to be extensible by IANA registration and follows the general pattern of the Session Initiation Protocol (SIP) [22] and HTTP [14]. The first digit indicates the type of response, with '2' signaling a successful request, '3' a redirection, '4' a request failure due to client behavior, and '5' a server failure.

If used within HTTP, LoST also utilizes the normal HTTP status codes. However, the HTTP request can succeed, while the LoST request caused an error. All LoST status codes appear in HTTP 200 (OK) responses. For example, a LoST 404, 414 or 500 status would occur in an HTTP 200 response.

Temporary unavailability of the service should be indicated by an HTTP 505 (Service Unavailable) status code.

[Editor's Note: Does this make any sense or should all or some LoST errors occur in a non-200 HTTP response?]

8.1. Informational 1xx

This document does not define informational status codes.

8.2. Successful 2xx

8.2.1. 200 OK

The query completed successfully.

8.2.2. 201 Service Substitution

The service requested is not available for the location requested, but the server is configured to provide a replacement service.

8.3. Redirection 3xx

8.3.1. 301 Move Permanently

The requested location is being mapped by a different server and all future requests for that location (and locations in the service area)

should be directed to that server.

8.3.2. 302 Moved Temporarily

The requested location is being mapped by a different server, but future requests should continue to use this server.

8.3.3. Example

This is an example of an error message with a 302 status code:

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:ietf:params:xml:ns:lost1">
  <redirect status="302"
    message="County-level routing"
    xml:lang="en"
    redirect="lost:co.lancaster.pa.us"
  </response>
```

8.4. Client Error 4xx

8.4.1. 400 Bad Request

The request could not be understood due to malformed syntax.

8.4.2. 403 Forbidden

The server understood the request, but is refusing to fulfill it. Authorization will not help, and the request SHOULD NOT be repeated.

8.4.3. 404 Not Found

The server has definitive information that there is no service mapping for the location specified.

8.4.4. 414 Location Error

The location provided does not exist or fields within the location information are contradictory.

8.4.5. Example

The first example shows an error message with a 414 status code that is attached to the response message indicating that there was a problem with the postal code:


```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:ietf:params:xml:ns:lost1">
  <result status="250" message="Default Route"
    xml:lang="en" timeToLive="10000">
    <displayName xml:lang="en">
      New York City Police Department
    </displayName>
    <service>unknown</service>
    <serviceBoundary>
      <civicLocation>
        <country>US</country>
        <A1>New York</A1>
        <A3>New York</A3>
      </civicLocation>
    </serviceBoundary>
    <uri>sip:nypd@example.com</uri>
    <uri>xmpp:nypd@example.com</uri>
    <service-number>911</service-number>
  </result>
  <failure status="414" message="Address error" xml:lang="en">
    <cause name="PC"
      message="postal code is outside of service boundary"
      xml:lang="en" />
  </failure>
</response>
```

The second example shows an error message with a 414 status code that is attached to the response message indicating that there was a problem with the provided geospatial location information:


```
<?xml version="1.0" encoding="UTF-8"?>
<response
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml" >
  <result status="250" message="Default PSAP"
    xml:lang="en" timeToLive="1000">
    <displayName xml:lang="en">
      New York City Police Department
    </displayName>
    <service>urn:service:sos.police</service>
    <serviceBoundary>
      <p2:Polygon srsName="urn:ogc:def::crs:EPSG::4326">
        <p2:exterior>
          <p2:LinearRing>
            <p2:pos>37.775 -122.4194</p2:pos>
            <p2:pos>37.555 -122.4194</p2:pos>
            <p2:pos>37.555 -122.4264</p2:pos>
            <p2:pos>37.775 -122.4264</p2:pos>
            <p2:pos>37.775 -122.4194</p2:pos>
          </p2:LinearRing>
        </p2:exterior>
      </p2:Polygon>
    </serviceBoundary>
    <uri>sip:nypd@example.com</uri>
    <uri>xmpp:nypd@example.com</uri>
    <serviceNumber>911</serviceNumber>
  </result>
  <failure status="414"
    message="Invalid Geographic Location" xml:lang="en">
    <cause name="p2:coordinates"
      message="invalid latitude" xml:lang="en" />
  </failure>
</response>
```

8.5. Server Error 5xx

8.5.1. 500 Server Internal Error

The server encountered an unexpected condition that prevented it from fulfilling the request. The client MAY retry the request after several seconds.

8.5.2. 501 Service Not Implemented

The server does not implement mapping for the service requested and cannot provide an alternate service.

8.5.3. 504 Server Time-Out

A server time-out occurs if the server contacted tries to recursively resolve the query, but cannot get an answer within the time limit set for the query.

8.5.4. Example

This is an example of an error message with a 500 status code:

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:ietf:params:xml:ns:lost1">
  <status code="500">Server failure</status>
</response>
```


9. LoST Transport

LoST needs an underlying protocol transport mechanisms to carry requests and responses. This document defines the use of LoST over HTTP and HTTP-over-TLS; other mechanisms are left to future documents. The available transport mechanisms are indicated in the LoST U-NAPTR DNS resource record. In protocols that support content type indication, LoST uses the media type `application/lost+xml`.

When using HTTP [[14](#)] and HTTP-over-TLS [[15](#)], LoST requests use the HTTP POST method. All HTTP responses are applicable. The HTTP URL is derived from the LoST URL via U-NAPTR translation, as discussed in [Section 4](#).

10. LoST Uniform Resource Locators

LoST Uniform Resource Locators (URLs) follow the format of URLs defined in [RFC 3986](#) [9], with the following ABNF:

LoST-URI = "lost:" host

'host' is defined in [Section 3.2.2 of RFC 3986](#) [9].

An example is 'lost:lostserver.example.com'

11. Example

After performing link layer attachment and end host performs stateful address autoconfiguration (in our example) using DHCP. Then, DHCP provides the end host with civic location as described in [19].

CAtype	CValue
0	US
1	New York
3	New York
6	Broadway
22	Suite 75
24	10027-0401

Figure 14: DHCP Civic Information Example

Additionally, DHCP may provide information about the LoST server that can be contacted. Alternatively, an additional step of indirection is possible, for example by having DHCP return a domain name that has to be resolved to one or more IP addresses hosting LoST servers.

Both at attachment time and call time, the client places a LoST request, including its civic location and the desired service. The request is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<findServiceByLocation xmlns="urn:ietf:params:xml:ns:lost1"
                        validate="false" operation="recursive">
  <locationInfo>
    <civicLocation>
      <country>US</country>
      <A1>New York</A1>
      <A3>New York</A3>
      <A6>Broadway</A6>
      <LOC>Suite 75</LOC>
      <PC>10027-0401</PC>
    </civicLocation>
  </locationInfo>
  <service>urn:service:sos.police</service>
</findServiceByLocation>
```

Mapping Request

Since the contacted LoST server has the requested information available the following response is returned. The <displayName> element indicates, as a human readable display string, that the 'New York City Police Department' is responsible for the given geographical area. The indicated URI allows the user to start communication using SIP or XMPP. The <validation> element indicates which parts of the civic address were matched successfully against a database and represent a known address. Other parts of the address, here, the suite number, were ignored and not validated. The <serviceBoundary> element indicates that all of New York City would result in the same response. The <serviceNumber> element indicates that the service can be reached via the emergency service number 911.

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns="urn:ietf:params:xml:ns:lost1">
  <result status="200" message="OK" xml:lang="en" timeToLive="10000">
    <displayName xml:lang="en">
      New York City Police Department
    </displayName>
    <service>unknown</service>
    <serviceBoundary>
      <civicLocation>
        <country>US</country>
        <A1>New York</A1>
        <A3>New York</A3>
      </civicLocation>
    </serviceBoundary>
    <uri>sip:nypd@example.com</uri>
    <uri>xmpp:nypd@example.com</uri>
    <service-number>911</service-number>
  </result>
</response>
```

Mapping Response

| |
+-----+

13. Relax NG Schema

This section provides the Relax NG schema used by LoST protocol in the compact form. The verbose form is included in [Appendix A](#).

```
default namespace = "urn:ietf:params:xml:ns:lost1"
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
namespace ns1 = "urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
namespace ns2 = "http://www.opengis.net/gml"

##
##      Location-to-Service Translation Protocol (LoST)
##
##      A LoST XML instance has three "root" types:
##      the findServiceByLocation query, the listServices query,
##      and the response to these queries.
##
start = findServiceByLocation | listServices | response

##
##      The queries.
##
div {
  findServiceByLocation =
    element findServiceByLocation {
      query,
      attribute validate { xsd:boolean >> a:defaultValue [ "false" ] }?
    }
  listServices = element listServices { query }
}

##
##      The response.
##
div {
  response =
    element response {

      ##
      ##      2xx responses.
      ##
      (result
      | element serviceList {
          list { xsd:anyURI* },
          status
        })?,
}
```



```
##
##          3xx, 4xx, and 4xx responses.
##
((error | redirect | failure)?),
extensionPoint
}
}

##
##          Query pattern.
##
div {
  query =
    element locationInfo { anyElement* },
    element service { xsd:anyURI },
    extensionPoint,
    [ a:defaultValue [ "recursive" ] ] attribute operation { text }?
}

##
##          A result.
##
div {

  ##
  ##          2xx response.
  ##
  result =
    element result {
      element displayName {
        xsd:string,
        attribute xml:lang { xsd:language }
      }?,
      element service { xsd:anyURI },
      element serviceBoundary {
        (civicLocation, polygon?) | (civicLocation?, polygon)
      }?,
      element uri { xsd:anyURI }+,
      element serviceNumber {
        xsd:string { pattern = "[0-9]+" }
      }?,
      element validation {
        list { xsd:QName* }
      }?,
      extensionPoint,
      attribute timeToLive { xsd:positiveInteger },
      status
    }
}
```



```
}

##
##      Non-result responses.
##
div {

    ##
    ##      5xx response.
    ##
    error = element error { status, extensionPoint }

    ##
    ##      3xx response.
    ##
    redirect =
        element redirect {
            status,
            attribute redirect { xsd:anyURI },
            extensionPoint
        }

    ##
    ##      4xx response.
    ##
    failure =
        element failure {
            status,
            element cause {
                attribute name { xsd:QName },
                attribute message { xsd:string },
                attribute xml:lang { xsd:language }
            }*,
            extensionPoint
        }
}

##
##      Status pattern.
##
div {
    status =
        attribute status { xsd:positiveInteger },
        attribute extendedStatus { xsd:positiveInteger }?,
        (attribute message { xsd:string },
         attribute xml:lang { xsd:language })?
}
```



```
##
##      Patterns for inclusion of elements from schemas in
##      other namespaces.
##
div {

    ##
    ##      A wildcard pattern for including any element
    ##      from any other namespace.
    ##
    anyElement =
        element * {
            (attribute * { text }
             | text
             | anyElement)*
        }

    ##
    ##      A point where future extensions
    ##      (elements from other namespaces)
    ##      can be added.
    ##
    extensionPoint = anyElement*

    ##
    ##      A pattern to include the GEOPRIV civil location elements.
    ##
    civicAddress =
        element ns1:* {
            (attribute * { text }
             | text
             | anyElement)*
        }

    ##
    ##      A definition of civic location from GEOPRIV.
    ##
    civicLocation = element civicLocation { civicAddress*, anyElement* }

    ##
    ##      A pattern to include GML elements.
    ##
    GML =
        element ns2:* {
            (attribute * { text }
             | text
             | anyElement)*
        }
```



```
    polygon =  
      element ns2:Polygon {  
        attribute * { text }*,  
        GML  
      }  
  }
```

14. Internationalization Considerations

This mechanism is largely for passing protocol information from one subsystem to another; as such, most of its elements are tokens not meant for direct human consumption. If these tokens are presented to the end user, some localization may need to occur. The content of the <displayName> element may be displayed to the end user, and it is thus a complex type designed for this purpose.

15. IANA Considerations

15.1. Content-type registration for 'application/lost+xml'

This specification requests the registration of a new MIME type according to the procedures of [RFC 4288](#) [[13](#)] and guidelines in [RFC 3023](#) [[12](#)].

MIME media type name: application

MIME subtype name: lost+xml

Mandatory parameters: none

Optional parameters: charset

Indicates the character encoding of enclosed XML.

Encoding considerations:

Uses XML, which can employ 8-bit characters, depending on the character encoding used. See [RFC 3023](#) [[12](#)], Section 3.2.

Security considerations:

This content type is designed to carry LoST protocol payloads.

Interoperability considerations: None

Published specification: RFCXXXX [NOTE TO IANA/RFC-EDITOR: Please replace XXXX with the RFC number of this specification.] this document

Applications which use this media type:

Emergency and Location-based Systems

Additional information:

Magic Number: None

File Extension: .lostxml

Macintosh file type code: 'TEXT'

Personal and email address for further information: Hannes
Tschofenig, Hannes.Tschofenig@siemens.com

Intended usage: LIMITED USE

Author:

This specification is a work item of the IETF ECRIT working group,
with mailing list address <ecrit@ietf.org>.

Change controller:

The IESG <iesg@ietf.org>

[15.2.](#) LoST Relax NG Schema Registration

URI: urn:ietf:params:xml:ns:lost

Registrant Contact: IETF ECRIT Working Group, Hannes Tschofenig
(Hannes.Tschofenig@siemens.com).

Relax NG Schema: The Relax NG schema to be registered is contained
in [Section 13](#). Its first line is

default namespace = "urn:ietf:params:xml:ns:lost1"

and its last line is

}

15.3. LoST Namespace Registration

URI: urn:ietf:params:xml:ns:lost

Registrant Contact: IETF ECRIT Working Group, Hannes Tschofenig
(Hannes.Tschofenig@siemens.com).

XML:

```
BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml1-basic/xhtml1-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
    content="text/html; charset=iso-8859-1"/>
  <title>LoST Namespace</title>
</head>
<body>
  <h1>Namespace for LoST</h1>
  <h2>urn:ietf:params:xml:ns:lost</h2>
  <p>See <a href="[URL of published RFC]">RFCXXX
    [NOTE TO IANA/RFC-EDITOR:
      Please replace XXXX with the RFC number of this
      specification.]</a>.</p>
</body>
</html>
END
```

15.4. Registration Template

This registration template is in accordance with [\[8\]](#).

URL scheme name:

lost

URL scheme syntax:

See [Section 10](#)

Character encoding considerations:

See [Section 10](#)

Intended Use:

The intended usage is described in this document.

Application and protocols which use this scheme:

The usage of the LoST URL scheme is targeted for this document and hence for location-based services that make use of the mapping protocol specified in this document.

Interoperability considerations:

None

Security considerations:

See [Section 16](#)

Relevant publications:

This document provides the relevant context for this URL scheme.

Contact:

Hannes Tschofenig, Hannes.Tschofenig@siemens.com

Author/Change controller:

The IESG <iesg@ietf.org>

16. Security Considerations

There are multiple threats to the overall system of which service mapping forms a part. An attacker that can obtain service contact URIs can use those URIs to attempt to disrupt those services. An attacker that can prevent the lookup of contact URIs can impair the reachability of such services. An attacker that can eavesdrop on the communication requesting this lookup can surmise the existence of an emergency and possibly its nature, and may be able to use this to launch a physical attack on the caller.

To avoid that an attacker can modify the query or its result, the authors RECOMMEND the use of channel security, such as TLS, with LoST.

A more detailed description of threats and security requirements are provided in [\[4\]](#).

17. Acknowledgments

[Editor's Note: Names need to be added here. Forgot it...Sorry.]

18. Open Issues

Please find open issues at: <http://www.ietf-ecrit.org:8080/lost/>

19. References

19.1. Normative References

- [1] World Wide Web Consortium, "XML Schema Part 2: Datatypes", W3C XML Schema, October 2000, <<http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>>.
- [2] World Wide Web Consortium, "XML Schema Part 1: Structures", W3C XML Schema, October 2000, <<http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>>.
- [3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [4] Taylor, T., "Security Threats and Requirements for Emergency Call Marking and Mapping", [draft-ietf-ecrit-security-threats-03](#) (work in progress), July 2006.
- [5] Schulzrinne, H. and R. Marshall, "Requirements for Emergency Context Resolution with Internet Technologies", [draft-ietf-ecrit-requirements-12](#) (work in progress), August 2006.
- [6] Schulzrinne, H., "A Uniform Resource Name (URN) for Services", [draft-ietf-ecrit-service-urn-05](#) (work in progress), August 2006.
- [7] Mealling, M., "The IETF XML Registry", [draft-mealling-iana-xmlns-registry-05](#) (work in progress), June 2003.
- [8] Petke, R. and I. King, "Registration Procedures for URL Scheme Names", [BCP 35](#), [RFC 2717](#), November 1999.
- [9] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [10] OpenGIS, "Open Geography Markup Language (GML) Implementation Specification", OGC OGC 02-023r4, January 2003.
- [11] Peterson, J., "A Presence-based GEOPRIV Location Object Format", [RFC 4119](#), December 2005.
- [12] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", [RFC 3023](#), January 2001.

- [13] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 4288](#), December 2005.
- [14] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.
- [15] Rescorla, E., "HTTP Over TLS", [RFC 2818](#), May 2000.
- [16] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for PIDF-LO", [draft-ietf-geopriv-revised-civic-lo-02](#) (work in progress), April 2006.
- [17] Daigle, L., "Domain-based Application Service Location Using URIs and the Dynamic Delegation Discovery Service (DDDS)", [draft-daigle-unaptr-00](#) (work in progress), June 2006.

19.2. Informative References

- [18] Schulzrinne, H., "The tel URI for Telephone Numbers", [RFC 3966](#), December 2004.
- [19] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information", [draft-ietf-geopriv-dhcp-civil-09](#) (work in progress), January 2006.
- [20] Schulzrinne, H., "Location-to-URL Mapping Architecture and Framework", [draft-ietf-ecrit-mapping-arch-00](#) (work in progress), August 2006.
- [21] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in support of Emergency Calling", [draft-rosen-sos-phonebcpr-01](#) (work in progress), June 2006.
- [22] 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.
- [23] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [draft-ietf-mmusic-ice-10](#) (work in progress), August 2006.
- [24] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", [RFC 3958](#), January 2005.

Appendix A. Non-Normative RELAX NG Schema in XML Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar ns="urn:ietf:params:xml:ns:lost1"
  xmlns="http://relaxng.org/ns/structure/1.0"
  xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0"
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

  <start>
    <a:documentation>
      Location-to-Service Translation Protocol (LoST)

      A LoST XML instance has three "root" types:
      the findServiceByLocation query, the listServices query,
      and the response to these queries.
    </a:documentation>
    <choice>
      <ref name="findServiceByLocation" />
      <ref name="listServices" />
      <ref name="response" />
    </choice>
  </start>

<div>
  <a:documentation>
    The queries.
  </a:documentation>

  <define name="findServiceByLocation">
    <element name="findServiceByLocation">
      <ref name="query" />
      <optional>
        <attribute name="validate">
          <data type="boolean" />
          <a:defaultValue>false</a:defaultValue>
        </attribute>
      </optional>
    </element>
  </define>

  <define name="listServices">
    <element name="listServices">
      <ref name="query" />
    </element>
  </define>

</div>
```



```
<div>
  <a:documentation>
    The response.
  </a:documentation>

  <define name="response">
    <element name="response">
      <optional>
        <choice>
          <a:documentation>
            2xx responses.
          </a:documentation>
          <ref name="result" />
          <element name="serviceList">
            <list>
              <zeroOrMore>
                <data type="anyURI" />
              </zeroOrMore>
            </list>
            <ref name="status" />
          </element>
        </choice>
      </optional>
      <optional>
        <a:documentation>
          3xx, 4xx, and 4xx responses.
        </a:documentation>
        <choice>
          <ref name="error" />
          <ref name="redirect" />
          <ref name="failure" />
        </choice>
      </optional>
      <ref name="extensionPoint" />
    </element>
  </define>

</div>

<div>
  <a:documentation>
    Query pattern.
  </a:documentation>

  <define name="query">
    <element name="locationInfo">
      <zeroOrMore>
        <ref name="anyElement"/>
      </zeroOrMore>
    </element>
  </define>

</div>
```



```
        </zeroOrMore>
    </element>
    <element name="service">
        <data type="anyURI"/>
    </element>
    <ref name="extensionPoint" />
    <optional>
        <attribute name="operation">
            <a:defaultValue>recursive</a:defaultValue>
        </attribute>
    </optional>
</define>

</div>

<div>
    <a:documentation>
        A result.
    </a:documentation>

    <define name="result">
        <a:documentation>
            2xx response.
        </a:documentation>
        <element name="result">
            <optional>
                <element name="displayName">
                    <data type="string"/>
                    <attribute name="xml:lang">
                        <data type="language" />
                    </attribute>
                </element>
            </optional>
            <element name="service">
                <data type="anyURI"/>
            </element>
            <optional>
                <element name="serviceBoundary">
                    <choice>
                        <group>
                            <ref name="civicLocation" />
                            <optional>
                                <ref name="polygon" />
                            </optional>
                        </group>
                        <group>
                            <optional>
                                <ref name="civicLocation" />
                            </optional>
                        </group>
                    </choice>
                </element>
            </optional>
        </element>
    </define>
</div>
```



```
        </optional>
        <ref name="polygon" />
      </group>
    </choice>
  </element>
</optional>
<oneOrMore>
  <element name="uri">
    <data type="anyURI"/>
  </element>
</oneOrMore>
<optional>
  <element name="serviceNumber">
    <data type="string">
      <param name="pattern">[0-9]+</param>
    </data>
  </element>
</optional>
<optional>
  <element name="validation">
    <list>
      <zeroOrMore>
        <data type="QName"/>
      </zeroOrMore>
    </list>
  </element>
</optional>
  <ref name="extensionPoint" />
  <attribute name="timeToLive">
    <data type="positiveInteger"/>
  </attribute>
  <ref name="status" />
</element>
</define>

</div>

<div>
  <a:documentation>
    Non-result responses.
  </a:documentation>

  <define name="error">
    <a:documentation>
      5xx response.
    </a:documentation>
    <element name="error">
      <ref name="status"/>
    </element>
  </define>
</div>
```



```
        <ref name="extensionPoint" />
      </element>
    </define>

    <define name="redirect">
      <a:documentation>
        3xx response.
      </a:documentation>
      <element name="redirect">
        <ref name="status"/>
        <attribute name="redirect">
          <data type="anyURI"/>
        </attribute>
        <ref name="extensionPoint" />
      </element>
    </define>

    <define name="failure">
      <a:documentation>
        4xx response.
      </a:documentation>
      <element name="failure">
        <ref name="status"/>
        <zeroOrMore>
          <element name="cause">
            <attribute name="name">
              <data type="QName"/>
            </attribute>
            <attribute name="message">
              <data type="string"/>
            </attribute>
            <attribute name="xml:lang">
              <data type="language"/>
            </attribute>
          </element>
        </zeroOrMore>
        <ref name="extensionPoint" />
      </element>
    </define>

  </div>

  <div>
    <a:documentation>
      Status pattern.
    </a:documentation>

    <define name="status">
```



```
<attribute name="status">
  <data type="positiveInteger"/>
</attribute>
<optional>
  <attribute name="extendedStatus">
    <data type="positiveInteger"/>
  </attribute>
</optional>
<optional>
  <group>
    <attribute name="message">
      <data type="string"/>
    </attribute>
    <attribute name="xml:lang">
      <data type="language"/>
    </attribute>
  </group>
</optional>
</define>

</div>

<div>
  <a:documentation>
    Patterns for inclusion of elements from schemas in
    other namespaces.
  </a:documentation>

  <define name="anyElement">
    <a:documentation>
      A wildcard pattern for including any
      element from any other namespace.
    </a:documentation>
    <element>
      <anyName/>
      <zeroOrMore>
        <choice>
          <attribute>
            <anyName/>
          </attribute>
          <text/>
          <ref name="anyElement"/>
        </choice>
      </zeroOrMore>
    </element>
  </define>

  <define name="extensionPoint">
```



```
<a:documentation>
  A point where future extensions
  (elements from other namespaces)
  can be added.
</a:documentation>
<zeroOrMore>
  <ref name="anyElement" />
</zeroOrMore>
</define>

<define name="civicAddress">
  <a:documentation>
    A pattern to include the GEOPRIV civil location elements.
  </a:documentation>
  <element>
    <nsName ns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"/>
    <zeroOrMore>
      <choice>
        <attribute>
          <anyName/>
        </attribute>
        <text/>
        <ref name="anyElement"/>
      </choice>
    </zeroOrMore>
  </element>
</define>

<define name="civicLocation">
  <a:documentation>
    A definition of civic location from GEOPRIV.
  </a:documentation>
  <element name="civicLocation">
    <zeroOrMore>
      <ref name="civicAddress"/>
    </zeroOrMore>
    <zeroOrMore>
      <ref name="anyElement" />
    </zeroOrMore>
  </element>
</define>

<define name="GML">
  <a:documentation>
    A pattern to include GML elements.
  </a:documentation>
  <element>
    <nsName ns="http://www.opengis.net/gml" />
```



```
<zeroOrMore>
  <choice>
    <attribute>
      <anyName/>
    </attribute>
    <text/>
    <ref name="anyElement" />
  </choice>
</zeroOrMore>
</element>
</define>

<define name="polygon">
  <element name="Polygon" ns="http://www.opengis.net/gml">
    <zeroOrMore>
      <attribute>
        <anyName/>
      </attribute>
    </zeroOrMore>
    <ref name="GML"/>
  </element>
</define>

</div>

</grammar>
```


Authors' Addresses

Ted Hardie
Qualcomm, Inc.

Email: hardie@qualcomm.com

Andrew Newton
SunRocket
8045 Leesburg Pike, Suite 300
Vienna, VA 22182
US

Phone: +1 703 636 0852
Email: andy@hxr.us

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
US

Phone: +1 212 939 7004
Email: hgs+ecrit@cs.columbia.edu
URI: <http://www.cs.columbia.edu>

Hannes Tschofenig
Siemens
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Phone: +49 89 636 40390
Email: Hannes.Tschofenig@siemens.com
URI: <http://www.tschofenig.com>

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

