ECRIT R. Marshall
Internet-Draft J. Martin
Intended status: Standards Track TCS
Expires: January 24, 2015 B. Rosen
Neustar
July 23, 2014

# A LoST extension to return complete and similar location info
# draft-marshall-ecrit-similar-location-04

## Abstract

This document introduces a new way to provide returned location information in LoST responses that is either of a completed or similar form to the original input civic location, based on whether valid or invalid civic address elements are returned within the findServiceResponse message. This document defines a new extension to the findServiceResponse message within the LoST protocol [RFC5222] that enables the LoST protocol to return a completed civic address element set for a valid location response, and one or more suggested sets of similar location information for invalid LoST responses. These two types of civic addresses are referred to as either "complete location" or "similar location", and are included as compilation of ca type xml elements within the existing LoST findServiceResponse message structure.

## Status of This Memo

Copyright Notice

Table of Contents

## 1.  Introduction

The LoST protcol [RFC5222] supports the validation of civic location information as input, by providing a set of validation result status indicators.  The current usefullness of the supported xml elements, "valid", "invalid", and "unchecked", is limited, because while they each provide an indication of validity for any one location element as a part of the whole civic address, the mechanism is insufficient in providing either the complete set of civic address elements that the LoST server contains, or of providing alternate suggestions (hints) as to which civic address is intended for use.

Whether the input civic location is valid and missing information, or invalid due to missing or wrong information during input, this document provides a mechanism to return a complete set of civic address elements for those valid or invalid cases.

This enhancement to the validation feature within LoST is required by systems that rely on accurate location for processing in order to increase the likelyhood that the correct and/or complete form of a civic location becomes known in those cases where it is incomplete or just plain wrong.  One such use case is that of location based emergency calling.  The use of this protocol extension will reduce user and system input errors, and will result in a higher level of civic address matching, reducing the number of mismatch errors, where a civic address that appears to be valid gets wrongly associated with the physical location of the caller.

The structure of this document includes terminology, Section 2, followed by a discussion of the basic elements involved in location validation.  The use of these elements, by way of example, is discussed in an overview section, Section 3, with accompanying rationale, and a brief discussion of the impacts to LoST, and its current schema.

## 2.  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 RFC 2119 [RFC2119], with the important qualification that, unless otherwise stated, these terms apply to the design of the Location Configuration Protocol and the Location Dereferencing Protocol, not its implementation or application.

The following terms are defined in this document:

Location:  The term Location can be used to refer to either a civic location or a geodetic location.

Geodetic Location:  a geographic coordinate set of values that describes a point within a defined geographic datum.  For example, a WGS84 referenced lattitude, longitude coordinate pair (2D), or lattitude, longitude, and altitude (3D).  Note: geodetic location is defined here for context, but is not used elsewhere within this document.

Civic Location:  The term civic location applies to a set of one or more civic address elements that are used in conjunction with each other, and in accordance with a known ruleset to designates a

place within a defined grid or basemap, The example used within this document is a street address as defined in [RFC5139]

Civic Address: The term Civic Address is used interchangeably with the term Civic Location within this document.

Street Address: The term Street Address is used to represent a place, or location on a defined grid or map. While generally equated to both terms, Civic Location and Civic Address, it is not used within this document.

Civic Address Element: The term Civic Address Element is used within this document to apply to an individual CAtype data descriptor, for example, as is defined in [RFC4776]

Invalid: The status result of the unsuccessful attempt to match an individual input data as part of a larger set of data that has already been successfully matched and as shown by the [RFC5222] defined xml named element

Valid: The status result of the successful attempt to match an individual input data as part of a larger set of data that has already also been successfully matched and shown by the [RFC5222] defined xml named element

Invalid Location: A Civic Location that was included in a LoST request and subsequently returned with one or more civic address elements marked as invalid.

Valid Location: A Civic Location that was included in a LoST request and subsequently returned with all civic address elements marked as valid.

Complete Location: An expanded civic location that includes other civic address elements in addition to the existing validated civic address elements provided as input to a LoST server.

Similar Location: A suggested civic location that is comparatively close to the civic location which was input, but which had one or more invalid civic address elements returned by the LoST server.

Returned Location Information: A set of standard civic address elements returned in a LoST response.

## 3. Overview of Returned Location Information

This document describes an extension to LoST [RFC5222] to allow additional location information to be returned in a findServiceResponse for two different use cases.

When a LoST server is asked to validate a civic location, its goal is to take the set of civic address elements provided as the location information in the LoST request, and find a unique location in its database that matches the information in the request. Uniqueness might not require values for all possible elements in the civic address that the database might hold. Further, the input location information might not represent the form of location the users of the LoST service prefer to have. As an example, there are LoST civic address elements that could be used to define a postal location, suitable for delivery mail as well as a municipal location suitable for responding to an emergency call. While the LoST server might be able to determine the location from the postal elements provided, the emergency services would prefer that the municipal location be used for any subsequent emergency call. Since validation is often performed well in advance of an end-user placing an emergency call, if the LoST server could return the preferred form of location (or more properly, the municipal elements in addition to the postal elements), those elements could be stored in a LIS and used in a later emergency call.

Since a LoST server often contains more data than what is included within a findService request, it is expected that this additional location information, if present, SHOULD be returned within response messages that contain valid civic address elements. For valid location responses, where a LoST server contains additional location information relating to that civic address, the findServiceResponse message MAY return additional location information along with the original validated civic address elements in order to form a complete location based on local implementation policy.

In addition, this document describes the reuse of the same mechanism, but for a different purpose: to supply similar location information in the case where a LoST server response includes one or more civic address elements marked as invalid, constituting an invalid location response, offerring one or more suggested alternative address that would consist of one or more valid locations.

LoST servers that implement this extension have no way to alert clients that may not be aware of the extension's capabilities, other than supplying the extended data set. It is expected that a LoST client implementation that is not aware of this extension for complete and/or similar location SHOULD be able to still receive the

findServiceResponse data, while throwing away any extra complete or similar location data.

In a valid location response, a LoST server returns a response to a findService request that contains a set of civic address elements marked valid, the location information in the findServiceResponse message MAY be extended to include additional location information specific for that location.  As an example, the query might contain a HNO (house number), RD (road name) and A3 (city) and a few more caType elements, but might not contain A1 (state), PC (Postal Code) CAtypes.  The HNO, RD, STS, POD, and A3 civic address elements might be sufficient enough to the LoST server to uniquely locate the address specified in the request and thus be considered valid.  Yet, downstream entities might find it helpful to have the additional country, A1 (state), and PC, (Postal Code), civic address elements that are present within the LoST server, be included as part of a complete location response.  Since [RFC5222] currently does not have a way for this additional location information to be returned in the findServiceResponse, this document extends the LoST protocol so that it can include a completeLocation element within the findServiceResponse message, allowing for the representation of complete location information.

An example showing complete location information supplied:

input address: 6000 15th Ave NW Seattle

complete location: 6000 15th Ave NW Seattle, WA 98105 US

By constrast, when invalid location is received from the LoST server, with this extension, the same mechanism works as follows: if a LoST server returns a response to a findService request that contains a set of civic address elements with one or more labeled as invalid, the location information in the findServiceResponse is extended to include additional location information that it knows is specific for that location.  Differing results based on somewhat close input data as used above, where the HNO, RD, STS, A1, and A3 civic address elements are not sufficient to locate a unique address leads to an invalid location result.  This is the case, despite the fact that the LoST server typically contains additional civic address elements which could have resulted in a uniquely identifiable location if additional data had been supplied with the query.  Since [RFC5222] currently does not have a way for this additional location information to be returned in the findServiceResponse, this document extends [RFC5222] so that the LoST findServiceResponse message can include one or more similarLocation elements within the findServiceResponse message representing similar civic locations.

To show this, suppose that a slightly modified address as above is inserted within a Lost findService request:

input address: 6000 15th Ave N Seattle, WA.

Different from the previous use case, this time we make the assumption that the address is deemed "invalid" by the LoST server because there is no such thing as "15th Ave N" within the LoST server's data for the city of Seattle. However, we also happen to know for this example that there are two addresses within the address dataset that are "similar", when all parts of the address are taken as a whole. These similar addresses that could be suggested to the user are as follows:

similar address #1: 6000 15th Ave NW Seattle, WA 98107

similar address #2: 6000 15th Ave NE Seattle, WA 98105

This document proposes to include the above similar addresses as civicAddress elements in the response to locationValidation. The next section shows examples of the LoST request and response xml message fragments for the above valid and invalid scenarios, returning the complete or similar addresses, respectively:

## 4. Returned Location Information

The LoST server knows the data that is available internally, and can determine which additional civic address elements can be provided either as part of a complete location or a similar location. The inclusion of either complete location or similar location is not triggered by any message parameter, but is triggered based on whether the returned location information is valid or invalid. It is not turned on or off, but is implementation specific.

## 5. Complete Location returned for Valid Location response

Based on the example input request, returned location information is provided in a findServiceResponse message when the original input address is considered valid, but is missing some additional data that the LoST server has.

```
   <!-- =====Request=================================== -->


   <findService xmlns="urn:ietf:params:xml:ns:lost1"
     validateLocation="true">
```

```
        <location id="587cd3880" profile="civic">
          <civicAddress
            xmlns="urn:ietf:params:mxl:ns:pidf:geopriv10:civicAddr">

            <A1>WA</A1>
            <A3>Seattle</A3>
            <RD>15th</RD>
            <STS>Ave</STS>
            <POD>NW</POD>
            <HNO>6000</HNO>

          </civicAddress>
        </location>

        <service>urn:service:sos</service>

      </findService>


      <!-- =====Response================================== -->


      <findServiceResponse >
        xmlns="urn:ietf:params:xml:ns:lost1"
        xmlns:rli="urn:ietf:params:xml:ns:lost-rli1">
        xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">

        <mapping
          expires="NO-CACHE"
          lastUpdated="2006-11-01T01:00:00Z"
          source="authoritative.example"
          sourceId="8799e346000098aa3e">

          <displayName xml:lang="en">Seattle 911</displayName>
          <service>urn:service:sos</service>
          <uri>sip:seattle-911@example.com</uri>
          <serviceNumber>911</serviceNumber>

        </mapping>

        <locationValidation

          <valid>ca:A3 ca:RD ca:STS ca:POD ca:HNO</valid>
          <invalid></invalid>
          <unchecked></unchecked>

          <rli:completeLocation>  <!-- completed address -->
            <ca:civicAddress>
```

```
              <ca:country>US</ca:country>
              <ca:A1>WA</ca:A1>
              <ca:A3>SEATTLE</ca:A3>
              <ca:RD>15TH</ca:RD>
              <ca:STS>AVE</ca:STS>
              <ca:POD>NW</ca:POD>
              <ca:HNO>6000</ca:HNO>
              <ca:PC>98106</ca:PC>
              <ca:PCN>SEATTLE</ca:PCN>
            </ca:civicAddress>

        </rli:completeLocation>

      </locationValidation>

        <path>
          <via source="authoritative.example"/>
        </path>

        <locationUsed id="587cd3880"/>

      </findServiceResponse>


      <!-- =============================================== -->
```

## 6. Similar Location returned for Invalid Location response

The following example shows returned location information provided in a findServiceResponse message when the original input address is considered invalid, because of the unmatchable POD data (in this example) that the LoST server needs to provide a unique mapping.

```
   <!-- =====Request==================================== -->


      <findService xmlns="urn:ietf:params:xml:ns:lost1"
        validateLocation="true">

        <location id="587cd3880" profile="civic">
          <civicAddress
            xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">

            <country>US</country>
```

```
            <A1>WA</A1>
            <A3>Seattle</A3>
            <RD>15th</RD>
            <STS>Ave</STS>
            <POD>N</POD>
            <HNO>6000</HNO>

          </civicAddress>
        </location>

        <service>urn:service:sos</service>

      </findService>

      <!-- =====Response=================================== -->


      <findServiceResponse>
        xmlns="urn:ietf:params:xml:ns:lost1"
        xmlns:rli="urn:ietf:params:xml:ns:lost-rli1">
        xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">

        <mapping
          expires="NO-CACHE"
          lastUpdated="2006-11-01T01:00:00Z"
          source="authoritative.example"
          sourceId="8799e346000098aa3e">

          <displayName xml:lang="en">Seattle 911</displayName>
          <service>urn:service:sos</service>
          <uri>sip:seattle-911@example.com</uri>
          <serviceNumber>911</serviceNumber>

        </mapping>

        <locationValidation

          <valid>ca:country ca:A1 ca:A3 ca:STS ca:RD</valid>
          <invalid>ca:POD</invalid>
          <unchecked>ca:HNO</unchecked>

          <rli:similarLocation>  <!-- similar location info -->
            <ca:civicAddress>  <!-- similar address #1 -->
              <ca:country>US</ca:country>
              <ca:A1>WA</ca:A1>
              <ca:A3>SEATTLE</ca:A3>
              <ca:RD>15TH</ca:RD>
              <ca:STS>AVE</ca:STS>
```

```
              <ca:POD>NW</ca:POD>
              <ca:HNO>6000</ca:HNO>
              <ca:PC>98106</ca:PC>
              <ca:PCN>SEATTLE</ca:PCN>
            </ca:civicAddress>

            <ca:civicAddress>  <!-- similar address #2 -->
              <ca:country>US</ca:country>
              <ca:A1>WA</ca:A1>
              <ca:A3>SEATTLE</ca:A3>
              <ca:RD>15TH</ca:RD>
              <ca:STS>AVE</ca:STS>
              <ca:POD>NE</ca:POD>
              <ca:HNO>6000</ca:HNO>
              <ca:PC>98105</ca:PC>
              <ca:PCN>SEATTLE</ca:PCN>
            </ca:civicAddress>
        </rli:similarLocation>

      </locationValidation>

        <path>
          <via source="authoritative.example"/>
        </path>

        <locationUsed id="587cd3880"/>

      </findServiceResponse>


      <!-- =============================================== -->
```

## 7. Relax NG schema

This section provides the Relax NG schema of LoST extensions in the compact form. The verbose form is included in a later section [to be supplied in a later version of this draft].

```
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
default namespace ns1 = "urn:ietf:params:xml:ns:lost-rli1"

##
##       Extension to LoST to support returned location information
##
start =
```

```
     returnedLocation

   div {
     returnedLocationResponse =
       element returnedLocationResponse {
         completeLocation, similarLocation, extensionPoint
       }
   }

   ##
   ##       completeLocation
   ##
   div {
     completeLocation =
       element location {
         attribute id { xsd:token },
         locationInformation
       }+
   }

   ##
   ##       similarLocation
   ##
   div {
     similarLocation =
       element location {
         attribute id { xsd:token },
         locationInformation
       }+
   }
   ##
   ##       Location Information
   ##
   div {
     locationInformation =
       extensionPoint+,
       attribute profile { xsd:NMTOKEN }?
   }

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

     ##
     ##         Any element not in the LoST namespace.
     ##
```

```
     notLost = element * - (ns1:* | ns1:*) { anyElement }

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

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

## 8. Security Considerations

Whether the input to the LoST server is valid or invalid, the LoST server ultimately determines what it considers to be valid. Even in the case where the input location is valid, the requester still might not actually understand where that location is. For this kind of valid location use case, this described extension would typically return more location information than the requester started with, which might reveal more about the location. While this might be very desirable in some scenarios including, for example, supporting an emergency call, it might not be as desirable for other services. Individual LoST server implementations SHOULD consider the risk of releasing more detail verses the value in doing so. Generally, it is not expected that this would be a significant problem as the requester must have enough location information to be considered valid, which in most cases is enough to uniquely locate the address. Providing more CAtypes generally doesn't actually reveal anything more. For invalid locations that are submitted, this extension would allow the LoST response to include location information which is similar to what was input, again resulting in more information provided in the response than was known during input. LoST server implementations SHOULD evaluate the particular use cases where this extension is supported, and weigh the risks around its use. Many similar database services available today via the Internet offer similar features, such as "did you mean", and address completion, so this capability is not introducing any fundamentally new threat.

## 9. IANA Considerations

### 9.1. Relax NG Schema Registration

URI: urn:ietf:params:xml:schema:lost-rli1

Registrant Contact: IETF ECRIT Working Group, Brian Rosen (br@brianrosen.net).

Relax NG Schema: The Relax NG schema to be registered is contained in Section 7. Its first line is

```
default namespace = "urn:ietf:params:xml:ns:lost-rli1
```

and its last line is

```
}
```

### 9.2. LoST Namespace Registration

URI: urn:ietf:params:xml:ns:lost-rli1

Registrant Contact: IETF ECRIT Working Group, Brian Rosen (br@brianrosen.net).

XML:

```
BEGIN
<?xml version="2.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-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 Planned Change Namespace</title>
</head>
<body>
  <h1>Namespace for LoST Returned Location Information extension</h1>
  <h2>urn:ietf:params:xml:ns:lost-rli1</h2>
<p>See <a href="http://www.rfc-editor.org/rfc/rfc????.txt">
   RFC????</a>.</p>
</body>
</html>
END
```

## 10. Acknowledgements

## 11. Changes from Previous Versions

### 11.1. Changes from draft-marshall-03 to -04

- Revised the text in Section 1 to better describe how this extension can be useful (Bradner)-

- Utilized RFC2119 language in the draft rather than removing the reference to it(Bradner)

- Added some text to explain how notification of this extension is expected for those clients that are not aware of this extension could be notified (Bradner)

- Modified security section text to include security considerations for both valid and invalid addresses used as input. (Stark)

- Acknowledged: need for extension point RNG/detailed xml and examples (Stark)

- Reworked terminology section and aligned with text based on comments (Stark)

- General editorial cleanup

## 12. References

### 12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5222] Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol", RFC 5222, August 2008.

### 12.2. Informative References

[RFC4776] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information", RFC 4776, November 2006.

[RFC5139] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", RFC 5139, February 2008.

## Authors' Addresses

Roger Marshall
TeleCommunication Systems, Inc.
2401 Elliott Avenue
2nd Floor
Seattle, WA  98121
US

Phone: +1 206 792 2424
Email: rmarshall@telecomsys.com
URI:   http://www.telecomsys.com

Jeff Martin
TeleCommunication Systems, Inc.
2401 Elliott Avenue
2nd Floor
Seattle, WA  98121
US

Phone: +1 206 792 2584
Email: jmartin@telecomsys.com
URI:   http://www.telecomsys.com

Brian Rosen
Neustar
470 Conrad Dr
Mars, PA  16046
US

Email: br@brianrosen.net