Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 24, 2009

A. Forte
H. Schulzrinne
Columbia University
March 23, 2009

Location-to-Service Translation Protocol (LoST) Extensions
draft-forte-ecrit-lost-extensions-02.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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 September 24, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

An important class of location-based services answer the question "What instances of this service are closest to me?" Examples include finding restaurants, gas stations, stores, automated teller machines, wireless access points (hot spots) or parking spaces. Currently, the Location-to-Service Translation (LoST) protocol only supports mapping locations to a single service based on service regions. This document describes an extension that allows queries "N nearest" and "within distance X".

Table of Contents

1. Introduction 3
2. Requirements Notation 3
3. Service Region 3
4. New Query Types: "N nearest" and "within distance X" 4
5. LoST Extensions 4
5.1. Nw Use of Shapes in Queries 4
6. Limiting the Number of Returned Service URIs 5
7. The <serviceLocation> Element in Responses 7
8. Security Considerations 9
9. IANA Considerations 9
9.1. LoST Extensions Relax NG Schema Registration 9
9.2. LoST Extensions Namespace Registration 9
10. Non-Normative RELAX NG Schema in XML Syntax 10
11. References 10
Authors' Addresses 11

1. Introduction

The Location-to-Service Translation (LoST) protocol [RFC5222] maps service identifiers (URNs) and civic or geospatial information to service URIs, based on service regions. While motivated by mapping locations to the public safety answering point (PSAP) serving that location, the protocol has been designed to generalize to other location mapping services.

However, the current LoST query model assumes that each service URI has a service region and that service regions do not overlap. This fits the emergency services model, where the service region of a PSAP is given by jurisdictional boundaries, but does not work as well for other services that do not have clearly defined boundaries. For example, any given location is likely served by a number of different restaurants, depending on how far the prospective customer is willing to walk or drive.

We extend LoST with two additional queries, giving the protocol the ability to find the N nearest instances of a particular service and all services within a given radius.

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 [RFC2119].

3. Service Region

In the LoST protocol, the <findServiceResponse> message includes a service region [RFC5222]. This is the geographical area for which a query will always receive the same response. Because of this, the LoST client will issue a new query only when it has moved out of its current service region.

In emergency services, as soon as the service region changes, the client queries the LoST server in order to discover the new PSAP. This is important since clients need to know their PSAP before an emergency occurs, so that no time is wasted in discovering the correct PSAP during the emergency.

Other location-based services are not as critical as emergency services, and points of interest can be discovered on demand, at the time they are needed and not before. Because of this, for location-based services other than emergency services, in many cases service

regions will be of little or no use.

4. New Query Types: "N nearest" and "within distance X"

We introduce two new types of querie, "N nearest" and "within distance X". The former returns the N points of interest closest to the client's physical location, the latter discovers all those points of interest residing within a given distance from the client's physical location.

5. LoST Extensions

For queries "within distance X", the LoST client needs to specify to the server the range within which instances of a particular service should be searched. In order to do this, we make use of the circular shape [PIDF-LO] in LoST queries.

For queries "N nearest", the lost client needs to let the server know N, that is, the maximum number of service URIs to be returned in a response. In order to specify this, we introduce the <limit> element.

Also, we introduce a new element in LoST responses, namely <serviceLocation>. This new element is used by the server to indicate to the client the physical location of points of interest. In doing so, the client can compute the distance and other metrics between its current location and the points of interest.

5.1. Nw Use of Shapes in Queries

In [PIDF-LO] different shapes are defined in order to represent a point and an area of uncertainty within which the user might be situated. In the present context, rather than seeing such shapes as an area of uncertainty for the physical location of the client, we see them as the area within which we want to find a service.

For example, Figure 1 shows a <findService> geodetic query using the circular shape. In particular, with the query shown in Figure 1, we are asking the LoST server to send us a list of service URNs for pizza places within 200 meters from our approximate position specified in <p2:pos>.

Aside from the circular shape, other shapes are also useful. In particular, there are situations in which it is useful to query for services in a certain direction of movement rather than in an exact physical location. For example, if a user is driving North from New

York City to Boston, it would be useful for this user to be able to query for services North of where he currently is, that is not in his current physical location nor at his final destination.

In order to do this, we use shapes such as an ellipse. The ellipse has a major and a minor dimension thus allowing for defining a "priviledged" direction by having the major dimension in the direction of movement. This concept is similar to the one of omnidirectional versus directional antennas, where a device can see in either all directions or a specific direction, respectively. In the present context the circular shape allows a device to search for services in any direction sourrounding its physical location while shapes such as the ellipse allow the device to search for services in a more specific direction. The ellipse shape is defined in [PIDF-LO].

```
<?xml version="1.0" encoding="UTF-8"?>
<findService
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  serviceBoundary="value"
  recursive="true">
  <location id="6020688f1ce1896d" profile="geodetic-2d">
    <p2:Circle srsName="urn:ogc:def:crs:EPSG::4326">
      <p2:pos>37.775 -122.422</p2:pos>
      <p2:radius uom="urn:ogc:def:uom:EPSG::9001">
         200
      </p2:radius>
    </p2:Circle>
  </location>
  <service>urn:service:local.pizza</service>
</findService>
```

Figure 1: A 'within distance X' <findService> geodetic query using the circular shape

6. Limiting the Number of Returned Service URIs

Limiting the number of results is helpful, particularly for mobile devices with limited bandwidth. For "N closest" queries, the client needs to be able to tell the server to return no more than N service URIs. In order to specify such limit, we define a new namespace "ext" for Lost extensions and introduce a new element, namely <ext: limit>, conveyed inside the <findService> element defined in [RFC5222]. Figures 2 and 3 show a <findService> geodetic query where

the client asks the server to return no more than 20 service URIs. In particular, Figure 2 shows a 'N closest' query while Figure 3 shows a query which is a combination of 'N closest' and 'within distance X'. When receiving such queries, the LoST server will return a list of no more than 20 points of interest.

If the available points of interest are more than N, then the server has to identify the N points of interest closest to the client's physical location and include those in the response.

```
<?xml version="1.0" encoding="UTF-8"?>
<findService xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  xmlns:ext="http://www.thisisnotdoneyet.net"
  serviceBoundary="value" recursive="true">
  <ext:limit>20</ext:limit>
  <location id="6020688f1ce1896d" profile="geodetic-2d">
    <p2:Point id="point1" srsName="urn:ogc:def:crs:EPSG::4326">
      <p2:pos>40.7128 -74.0092</p2:pos>
    </p2:Point>
  </location>
<service>urn:service:food.pizza</service>
</findService>
```

Figure 2: A 'N closest' <findService> geodetic query with the new <limit> element

```
<?xml version="1.0" encoding="UTF-8"?>
<findService
  xmlns="urn:ietf:params:xml:ns:lost1"
  xmlns:p2="http://www.opengis.net/gml"
  xmlns:ext="http://www.thisisnotdoneyet.net"
  serviceBoundary="value"
  recursive="true">
  <ext:limit>20</ext:limit>
  <location id="6020688f1ce1896d" profile="geodetic-2d">
    <p2:Circle srsName="urn:ogc:def:crs:EPSG::4326">
      <p2:pos>37.775 -122.422</p2:pos>
      <p2:radius uom="urn:ogc:def:uom:EPSG::9001">
         200
      </p2:radius>
    </p2:Circle>
  </location>
  <service>urn:service:local.pizza</service>
</findService>
```

Figure 3: A <findService> geodetic query with the new <limit> element. This query is a combination of 'N closest' and 'within distance X' queries.

7. The <serviceLocation> Element in Responses

It is important for the LoST client to know the location of a point of interest so that distance, route and other metrics can be computed. We introduce a new element, namely <serviceLocation>. The <serviceLocation> element contains the geodetic coordinates of a point of service and MUST be contained in a <mapping> element. In responses such as <findServiceResponse> [RFC5222], a list of service URIs, each with its own <serviceLocation> element, MUST be returned. The order of service URIs in the list is not relevant.

The <serviceLocation> element has one single attribute, 'profile', in order to specify the profile used. Only geodetic profiles SHOULD be used as the computation of the distance, route and other metrics would at some point require geocoding of the civic address in geodetic coordinates. Because of this, the position specified in <serviceLocation> SHOULD be represented by using the <Point> element. The <Point> element is described in Section 12.2 of [RFC5222] and in Section 5.2.1 of [PIDF-LO]. Figure 4 shows a <findServiceResponse> answer containing two location-to-service-URI mappings.

Since service regions are not relevant in the present context, they are not present in the response.

NOTE: The <locationUsed> element cannot be extended for this purpose as it is defined outside of the <mapping> element. In particular, in a response the <locationUsed> element is always one, while the number of service URIs is typically more than one.

There are situations, however, in which it is helpful to include a civic address together with the geodetic coordinates of a point of service. Usually, databases already contain the civic address of points of interest and for devices with limited capabilities it is not always possible to perform decoding of geocoordinates in order to determine the civic address. Because of this, including also the civic address in a response, can be useful. In order to do this, we include the <civicAddress> element as defined in [RFC5139] in each <mapping> element. Figure 4 shows a <findServiceResponse> answer with the <civicAddress> element.

```
<?xml version="1.0" encoding="UTF-8"?>
<findServiceResponse xmlns="urn:ietf:params:xml:ns:lost1"
```

```
        xmlns:p2="http://www.opengis.net/gml">
        <mapping
          expires="2007-01-01T01:44:33Z"
          lastUpdated="2006-11-01T01:00:00Z"
          source="authoritative.example"
          sourceId="7e3f40b098c711dbb6060800200c9a66">
          <displayName xml:lang="it">
            Che bella pizza e all' anima da' pizza da Toto'
          </displayName>
          <service>urn:service:local.pizza</service>
          <uri>sip:chebella@example.com</uri>
          <uri>xmpp:chebella@example.com</uri>
          <serviceNumber>2129397040</serviceNumber>
          <serviceLocation profile="geodetic-2d">
            <p2:Point id="point1" srsName="urn:ogc:def:crs:EPSG:4326">
              <p2:pos>33.665 -112.432</p2:pos>
            </p2:Point>
          </serviceLocation>
          <civicAddress
              xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
              <country>US</country>
              <A1>New York</A1>
              <A3>New York</A3>
              <A6>Broadway</A6>
              <HNO>321</HNO>
              <PC>10027</PC>
          </civicAddress>
        </mapping>
        <mapping
          expires="2007-01-01T01:44:33Z"
          lastUpdated="2006-11-01T01:00:00Z"
          source="authoritative.example"
          sourceId="7e3f40b098c711dbb6060800200c9b356">
          <displayName xml:lang="en">
            King Mario's Pizza
          </displayName>
          <service>urn:service:local.pizza</service>
          <uri>sip:marios@example.com</uri>
          <uri>xmpp:marios@example.com</uri>
          <serviceNumber>2129397157</serviceNumber>
          <serviceLocation profile="geodetic-2d">
            <p2:Point id="point1" srsName="urn:ogc:def:crs:EPSG:4326">
              <p2:pos>33.683 -112.412</p2:pos>
            </p2:Point>
          </serviceLocation>
          <civicAddress
              xmlns="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr">
              <country>US</country>
```

```
              <A1>New York</A1>
              <A3>New York</A3>
              <A6>Amsterdam Avenue</A6>
              <HNO>123</HNO>
              <PC>10027</PC>
          </civicAddress>
        </mapping>
        <path>
          <via source="resolver.example"/>
          <via source="authoritative.example"/>
        </path>
        <locationUsed id="6020688f1ce1896d"/>
      </findServiceResponse>
```

Figure 4: A <findServiceResponse> answer

8. Security Considerations

The same security considerations as in [RFC5222] apply.

9. IANA Considerations

9.1. LoST Extensions Relax NG Schema Registration

TODO.

9.2. LoST Extensions Namespace Registration

TODO.

10. 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"
  xmlns:ls="urn:ietf:params:xml:ns:lost1"
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

<div>
  <ls:mapping>
    ...
    ...
    <a:documentation>
      Location information about the returned point of service.
    </a:documentation>

    <define name="serviceLocation">
      <element name="serviceLocation">
        <ref name="ls:locationInformation"/>
      </element>
    </define>
  </ls:mapping>
</div>

<div xmlns:ext="http://www.thisisnotdoneyet.net">
  <a:documentation>
    A limit to the number of returned results.
  </a:documentation>

  <define name="ext:limit">
    <element name="ext:limit">
      <data type="positiveInteger"/>
    </element>
  </define>
</div>

  >
```

11. 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.

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

[PIDF-LO] Winterbottom, J., Thomson, M., and H. Tschofenig, "GEOPRIV PIDF-LO Usage Clarification, Considerations and Recommendations. IETF Internet Draft (Work in Progress)", February 2008.

Authors' Addresses

Andrea G. Forte
Columbia University
Department of Computer Science
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA

Email: andreaf@cs.columbia.edu

Henning Schulzrinne
Columbia University
Department of Computer Science
1214 Amsterdam Avenue, MC 0401
New York, NY 10027
USA

Email: hgs@cs.columbia.edu