ECRIT
Internet-Draft
Expires: April 26, 2006

H. Schulzrinne
Columbia U.

October 23, 2005

Location-to-URL Mapping Protocol (LUMP)
draft-schulzrinne-ecrit-lump-01

Status of this Memo

Copyright Notice

Abstract

LUMP (Location-to-URL Mapping Protocol) maps geographic locations, described as PIDF-LO objects containing civic or geospatial information, to one or more URLs. It is based on a standard RPC mechanism and supports updates. This document describes the message formats, while a companion document describes the overall system architecture.

Table of Contents

1. Terminology

In this document, the key words "MUST", "MUSTNOT", "REQUIRED", "SHALL", "SHALLNOT", "SHOULD", "SHOULDNOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [1] and indicate requirement levels for compliant implementations.

2. Definitions

In addition to the terms defined in [11], this document uses the following terms to describe LUMP:

authoritative resolver: Resolver that can provide the authoritative answer to a particular set of queries, e.g., covering a set of PIDF-LO civic labels or a particular region described by a geometric shape. In some (rare) cases of territorial disputes, two resolvers may be authoritative for the same region.

child: A child is a resolver that is authoritative for a subregion of a particular server. A child can in turn be parent.

cluster: A cluster is a group of resolver (servers) that all share the same mapping information and return the same results for queries. Clusters provide redundancy and share query load. Clusters are fully-meshed, i.e., they all exchange updates with each other.

complete: A civic mapping region is considered complete if it covers a set of hierarchical labels in its entirety, i.e., there is no other resolver that covers parts of the same region. (A complete mapping may have children that cover strict subsets of this region.) For example, a region spanning the whole country is complete, but a region spanning only some of the streets in a city is not.

hint: A hint provides a mapping from a region to a server name, used to short-cut mapping operations.

first resolver: The first resolver is the resolver contacted directly by the ESRP or end system to obtain a mapping. Architecturally, all resolvers can serve as first resolvers, although local policy may disallow this.

leaf: A resolver that has no children.

mapping: A mapping is a short-hand for 'mapping from a location object to one or more URLs describing either another mapping server or the desired PSAP URLs.

parent: A resolver that covers the region of all of its children. A resolver without a parent is a root resolver.

peer: A resolver maintains associations other resolvers, called peers. Peers synchronize their region maps.

querier: The resolver, ESRP or end system requesting a mapping.
region map: A data object describing a contiguous area covered by a resolver, either as a subset of a civic address or a geometric object.
root region map: A data object describing a contiguous area covered by a resolver, with no parent map.
resolver: The server providing (part of) the mapping service. Resolvers cooperate to offer the mapping service to queriers.
root resolver: A resolver without parents is a root resolver.

3. Introduction

The location-to-URL mapping protocol (LUMP) maps a tuple consisting of a service URN and a civic or geospatial location, typically specified as a PIDF-LO object, to a set of URLs that describe the services available for that location. The initial application is the mapping of locations to the appropriate Public Safety Answering Point (PSAP) for emergency calling. LUMP uses a common RPC protocol for its operations.

LUMP has the following properties, described more fully later in this document:

- Satisfies the requirements [11] for mapping protocols.
- LUMP supportes lookup as well as address validation for civic addresses.
- LUMP allows separate hierarchies and geographic service boundaries for each type of service.
- LUMP re-uses of the most commonly used RPC protocol, SOAP, with a variety of transport and security options. (Other mechanisms, such as XML-RPC or REST-style HTTP, may also work.) The choice is motivated by the availability of numerous well-tested implementations, both open and closed source, in just about any conceivable language framework (with the possible exception of Fortran and Cobol).
- LUMP uses a robust clustering and replication architectures that distributes load as widely as possible, with every resolver as an entry point.
- LUMP fully specifies mechanisms for distributing coverage-region information.
- Mapping can be based on either civic or geospatial location information, with no performance penalty for either.
- Service regions can overlap.
- LUMP supports split responsibility for a single civic hierarchy level. (Example: A city has three public safety agencies, with three PSAPs and independent mapping databases, each covering a subset of the streets in the city.)

LUMP can be deployed bottom-deployment as well as top-down, with no need for a global coordinating body or the management of a global namespace or DNS name. The mechanism described does not require a country-level mapping server or a set of "root" servers.
Mapping services can be offered close to the access network, by the multimedia service provider (MSP), including voice service providers (VSPs), or by independent third parties.
LUMP supports a mechanism for updates and synchronization.
LUMP uses automated cluster replication with guaranteed convergence properties for maximum robustness [7].
LUMP can be extended to additional operations and data types.
Scalable both horizontally and vertically, i.e., any number of servers can support each subset of the mapping information and the number of levels is not bounded.
LUMP minimizes round trips by caching individual mappings as well as coverage regions ("hinting"). Unless otherwise desired, there is only one message exchange (roundtrip delay) between the ESRP or end system requesting a mapping and the designated resolver. This also facilitates reuse of TLS or other secure transport association across multiple queries.
LUMP supports both exact and approximate (best-guess) matching, controllable by the querier.
Mapping servers require only limited mutual trust.

The overall mapping architecture employed by LUMP is described in a companion [12] document. This document assumes that the reader has consulted that document, as this document only describes the basic request-response message mechanism.

4. Introductory Example

For this example, assume that there is a SIP-based VSPs V that offers a first resolver service to its customers. The VSP operates a cluster of such LUMP servers, advertised to their customers via DHCP. For simplicity, we only look at resolution by civic address; resolution by geo coordinates work exactly in the same fashion.

Assume that in the United States, each state operates a resolver, covering the counties or parishes in the state. In our example, there is no server covering all of the United States or larger regions. Each county in the state in turn has a list of coverage regions, typically consisting of one or more PSAPs. The state servers have their own database that is not shared with the rest of country. Assume that the caller is located at 123 Broad Avenue, Bergen County, Leonia, New Jersey.

An end user affiliated with V1 needs to place an emergency call and dials "9-1-1". The end device translates this into an "sos" URI,

which reaches the outbound proxy operated by V1, acting as an ESRP here. The ESRP issues a LUMP request to the local first resolver, RV1. RV1 has stored the coverage regions for all the states and matches the request to the New Jersey server, using the PIDF-LO location information contained in the SIP INVITE request for the lookup operation. Since it operates in recursive mode, it in turn queries the New Jersey server, say, lump:state.nj.example.gov. That server does not want to reveal more detailed information to the caller and simple returns a URL for the state-wide emergency services proxy, say sip:sos@emergency.nj.example.gov.

The ESRP routes the call to sip:sos@emergency.nj.example.gov, a SIP proxy server. In one or more resolution steps, that proxy server in turn consults a local LUMP server with the same PIDF-LO location information. Assume that the town of Leonia is served by two PSAPs, which do not share the same database. Streets south of a main road are served by one, those north by another. The state LUMP server only knows that Leonia has two such servers and issues a request to both, i.e., lump:north.leonianj.example.gov and lump: south.leonianj.example.gov. Broad Avenue is divided by this street, with 124 Broad Avenue happening to fall north of the dividing line. Both LUMP servers get the request and the northern server returns an answer, while the southern server indicates that this address is outside of its coverage region. The northern server returns the PSAP address, say, sip:police@leonianj.example.gov. The proxy simply routes the call to that location, including the location information.

This is only one of many possible deployment scenarios. As noted elsewhere, the area served by each server does not have to correspond to a particular civic address level or can span multiple levels. The referral graph can differ between civic and geospatial addresses and can utilize completely different servers, beyond the first resolver.

5. Overview of System Operation

A querier, such as an ESRP or end system, desiring to obtain a location mapping follows the steps below:

Identify a resolver: Using either DHCP [2], a service location protocol such as DNS-SD [9] or SLP [6], a using-protocol configuration protocol (e.g., [10] for SIP) or another configuration mechanism, the querier obtains one DNS name for a LUMP resolver.

Determine the resolver: The domain name obtained in the previous step is resolved using the associated SRV [3] resource record. The querier chooses the highest-priority server, and continues down the list if that server does not respond. As detailed in the SRV specification, a querier chooses randomly among multiple entries

with the same weight. The use of DNSsec is RECOMMENDED.

Send query to resolver: The querier sends a LUMP query to the resolver identified in the previous step, using an existing or newly-established secure transport association. The query contains a PIDF-LO [4] object. The resolver either determines that it is authoritative for the location contained in the query, recursively queries other servers or it provides an indication whom to ask next.

In principle, the protocols between querier and resolvers and between LUMP servers do not have to be the same. We leave this for future study.

In the next section, we describe how LUMP works "behind the scenes" to perform this resolution.

6. Resolver Discovery

LUMP services may be operated by a variety of organizations and entities, including Internet service providers, Internet access providers, voice service providers, and specialized LUMP service providers, such as public safety agencies or commercial database vendors. Each of these can either advertise their own servers or servers operated by other entities.

LUMP supports a range of resolver discovery mechanisms. Essentially, any discovery protocol may be used, including SLP [6], DNS-based [9] or UDDI. If the Internet service provider offers LUMP services, it may advertise these via DHCP. If the voice service provider offers LUMP services, it may include those in the SIP device configuration [10].

In general, it is advantageous to use a resolver that is close, in both a network topology and geographic sense, to the querier. Such proximity reduces the query latency due to reduced round-trip times and, in many cases, such servers will already have the necessary results cached, or at least pointers to appropriate authoritative resolvers and may already have established security associations with the appropriate resolver.

7. Messages

LUMP currently defines two request/response interactions: the first one requests a mapping from a geo or civic location and the second one asks the server for its coverage region.

The mapping and coverage region query indicate the desired service.

Results returned by both queries indicate their validity (expiration) time.

If a civic-location query for a mapping does not contain a precise mapping, it contains a civic location object that indicates which parts of the civic address have been used in the mapping. If there is no precise match, zero or more civic location objects are returned indicating possible alternatives that do exist within the mapping database.

Region queries are only meaningful if addressed to authoritative servers. These servers respond with a list of polygons and/or civic address descriptions indicating their coverage region.

LUMP uses web services as its protocol substrate. The schema for the LUMP messages is shown in Figure 1, while the web services definition is shown in Figure 2.

```
<schema targetNamespace="urn:ietf:params:xml:ns:lump"
     xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
     xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
     xmlns="http://www.w3.org/2000/10/XMLSchema">

  <include schemaLocation="civic.xsd"/>
  <include schemaLocation="geopriv.xsd"/>

  <complexType name="empty"/>

  <element name="mappingRequest" type="mappingRequestType"/>
  <element name="mappingResponse" type="mappingResponseType"/>
  <element name="regionRequest" type="mappingRequestType"/>
  <element name="regionResponse" type="mappingResponseType"/>

  <complexType name="mappingRequestType">
    <sequence>
      <element name="service" type="anyURI"/>
      <choice>
        <element name="recurse" type="empty"/>
        <element name="redirect" type="empty"/>
      </choice>
      <choice>
        <element name="civic" type="ca:civicAddress"/>
        <element name="geo" type="ca:civicAddress"/>
      </choice>
    </sequence>
  </complexType>

  <complexType name="mappingResponseType">
```

```
      <sequence>
        <element name="URI" type="anyURI"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="civicMatch" type="ca:civicAddress"
          minOccurs="0" maxOccurs="1"/>
        <element name="civicAlternate" type="ca:civicAddress"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="expires" type="dateTime" use="required"/>
    </complexType>

    <complexType name="regionRequestType">
      <sequence>
        <element name="service" type="anyURI"/>
      </sequence>
    </complexType>

    <complexType name="regionResponseType">
      <sequence>
        <element name="civicRegion" type="ca:civicAddress"
          minOccurs="0" maxOccurs="unbounded"/>
        <element name="geoRegion" type="ca:locInfoType"
          minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="expires" type="dateTime" use="required"/>
    </complexType>

  </schema>
```

Schema for LUMP

Figure 1

```
  <?xml version="1.0" encoding="UTF-8"?>
  <definitions name="LUMP"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
    xmlns:civic="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
    xmlns:lump="urn:ietf:params:xml:ns:lump"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:tns="urn:ietf:params:xml:ns:lump:proto"
    targetNamespace="urn:ietf:params:xml:ns:lump:proto">
```

```
    <import namespace="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
      location="http://www.cs.columbia.edu/~hgs/tmp/civic.xsd"/>

    <import namespace="urn:ietf:params:xml:ns:pidf:geopriv10"
      location="http://www.cs.columbia.edu/~hgs/tmp/geopriv.xsd"/>

    <import namespace="urn:ietf:params:xml:ns:lump"
      location="http://www.cs.columbia.edu/~hgs/tmp/lump.xsd"/>

    <message name="mappingRequestMessage">
      <part name="body" element="lump:mappingRequest"/>
    </message>
    <message name="mappingReponseMessage">
      <part name="body" element="lump:mappingResponse"/>
    </message>

    <wsdl:portType name="mappingPortType">
      <wsdl:operation name="mapping" parameterOrder="body">
        <wsdl:input message="tns:mappingRequestMessage"/>
        <wsdl:output message="tns:mappingResponseMessage"/>
      </wsdl:operation>
    </wsdl:portType>

    <wsdl:binding name="lumpSoapBinding" type="tns:mappingPortType">
        <wsdlsoap:binding style="rpc"
          transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="mapping">
           <wsdlsoap:operation soapAction=""/>
           <wsdl:input>
              <wsdlsoap:body
              encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
              use="encoded"/>
           </wsdl:input>
           <wsdl:output>
              <wsdlsoap:body
              encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
              use="encoded"/>
           </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>

    <wsdl:service name="lump">
      <wsdl:port binding="tns:lumpSoapBinding" name="mappingPortType">
       <wsdlsoap:address location="http://www.example.com"/>
      </wsdl:port>
    </wsdl:service>

  </definitions>
```

Schema for LUMP

Figure 2

8. Configuring Emergency Dial Strings

For the foreseeable future, some user devices and software will emulate the user interface of a telephone, i.e., the only way to enter call address information is via a 12-button keypad. Also, emergency numbers are likely to used until essentially all communication devices feature IP connectivity and an alphanumeric keyboard. Unfortunately, more than 60 emergency numbers are in use throughout the world, with many of those numbers serving non-emergency purposes elsewhere, e.g., identifying repair or directory services. Countries also occasionally change their emergency numbers, for example, by selecting a number already in use in other countries of a region (such as 112 in Europe).

Thus, a system that allows devices to be used internationally to place emergency calls needs to allow devices to discover emergency numbers automatically. In the system proposed, these numbers are strictly of local significance and are generally not visible in call signaling messages.

For simplicity of presentation, this section assumes that emergency numbers are valid throughout a country, rather than, say, be restricted to a particular city. This appears likely to be true in countries likely to deploy IP-based emergency calling solutions. In addition, the solution proposed also works if certain countries do not use a national emergency number. There is no requirement that a country uses a single emergency number for all emergency services, such as fire, police, or rescue.

For the best user experience, systems should be able to discover two sets of numbers, namely those used in the user's home country and in the country the user is currently visiting. The user is most likely to remember the former, but a companion borrowing a device in an emergency may only know the local emergency numbers.

Determining home and local emergency numbers is a configuration problem, but unfortunately, existing configuration mechanisms are ill-suited for this purpose. For example, a DHCP server might be able to provide the local emergency number, but not the home numbers. Similarly, SIP configuration would be able to provide the numbers valid at the location of the SIP service provider, but even a SIP service provider with national footprint may serve customers that are visiting any number of other countries.

Since dial strings are represented as URLs [5], the problem of determining local and home emergency numbers is a problem of mapping locations to a set of URLs, i.e., exactly the problem that LUMP is solving already.

The mapping operation is almost exactly the same as for determining the emergency service URL. The only difference is that if a querier knows the civic location at least to the country level, it will use a query where the PIDF-LO only includes the country code. If it only knows its geospatial location, it has to include that longitude and latitude. The querier uses the service identifiers "dialstring.sos", "dialstring.sos.fire", etc. The resolver returns the appropriate set of URLs and, if a geospatial location was used in the query, the current region map for the country.

Within the LUMP system, emergency calling regions are global information, i.e., they are distributed using the peer broadcast mechanism described earlier. Thus, every resolver has access to all region mappings. This makes it possible that a querier can ask any resolver for this information, reducing the privacy threat of revealing its location outside of an emergency call. The privacy threat is further reduced by the long-lived nature of the information, i.e., in almost all cases, the querier will have already cached the national boundary information or country information on its first visit to the country, using the normal LUMP hinting mechanism. (Given the modest storage needs, a querier could even cache all boundary maps.)

9. Security

LUMP addresses the following security issues, usually through the underlying transport security associations:

Server impersonation: Queriers, cluster members and peers can assure themselves of the identity of the remote party by using the facilities in the underlying channel security mechanism, such as TLS.

Query or query result corruption: To avoid that an attacker can modify the query or its result, LUMP RECOMMENDS the use of channel security, such as TLS.

Region corruption: To avoid that a third party or an untrustworthy member of the LUMP server population introduces a region map that it is not authorized for, any peer introducing a new region map MUST sign the object by encapsulating the data into a CMS wrapper. A recipient MUST verify, through a local policy mechanism, that the signing entity is indeed authorized to speak for that region. Determining who can speak for a particular region is inherently difficult unless there is a small set of authorizing entities that

resolvers can trust. Receiving resolvers should be particularly suspicious if an existing region map is replaced with a new one with a new resolver address.

Additional threats that need to be addressed by operational measures include denial-of-service attacks.

10. References

10.1 Normative References

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

[2] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[3] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[4] Peterson, J., "A Presence-based GEOPRIV Location Object Format", draft-ietf-geopriv-pidf-lo-03 (work in progress), September 2004.

[5] Rosen, B., "Dialstring parameter for the sip URI", draft-rosen-iptel-dialstring-02 (work in progress), July 2005.

10.2 Informative References

[6] Guttman, E., Perkins, C., Veizades, J., and M. Day, "Service Location Protocol, Version 2", RFC 2608, June 1999.

[7] Zhao, W., Schulzrinne, H., and E. Guttman, "Mesh-enhanced Service Location Protocol (mSLP)", RFC 3528, April 2003.

[8] Newton, A. and M. Sanz, "IRIS: The Internet Registry Information Service (IRIS) Core Protocol", RFC 3981, January 2005.

[9] Krochmal, M. and S. Cheshire, "DNS-Based Service Discovery", draft-cheshire-dnsext-dns-sd-03 (work in progress), July 2005.

[10] Petrie, D., "A Framework for Session Initiation Protocol User Agent Profile Delivery", draft-ietf-sipping-config-framework-07 (work in progress), July 2005.

[11] Schulzrinne, H. and R. Marshall, "Requirements for Emergency

Context Resolution with Internet Technologies", draft-schulzrinne-ecrit-requirements-01 (work in progress), July 2005.

[12] Schulzrinne, H., "Location-to-URL Mapping Architecture and Framework", draft-schulzrinne-ecrit-mapping-arch-00 (work in progress), October 2005.

Author's Address

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

Appendix A. Acknowledgments

Richard Stastny, ... provided helpful comments.

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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