Network Working Group
Internet-Draft
Expires: December 18, 2006

T. Hardie
Qualcomm, Inc.
A. Newton
SunRocket
H. Schulzrinne
Columbia U.
H. Tschofenig
Siemens
June 16, 2006

LoST: A Location-to-Service Translation Protocol
draft-ietf-ecrit-lost-00.txt

Status of this Memo

This Internet-Draft will expire on December 18, 2006.

Copyright Notice

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

This document describes a protocol for mapping a service identifier[6] and location information compatible with PIDF-LO [10] 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:

- Supports queries using civic as well as geospatial location information.
- Can be used in both recursive and iterative resolution.
- Can be used for civic address validation.
- A hierarchical deployment of mapping servers is independent of civic location labels.
- Can indicate errors in the location data to facilitate debugging and proper user feedback while simultaneously providing best-effort answers.
- Mapping can be based on either civic or geospatial location information, with no performance penalty for either.
- Service regions can overlap.
- Satisfies the requirements [5] for mapping protocols.
- Minimizes round trips by caching individual mappings and by supporting return of coverage regions ("hinting").
- Facilitates reuse of 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. [12] is a first attempt to describe such a mapping server architecture.

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

```
       Location
       Info      +----------+
       --------> |          |
       Service   |  LoST    |
       URN       |  Server  |
       --------> |          |
                 +----------+

        Query


            URI +----------+
       <------- |          |
       Optional |  LoST    |
    Info (hints)|  Server  |
       <------- |          |
                +----------+

        Response
```

Figure 1: Overview

The query message carries location information and a service identifier enconded 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 the location object. 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 region where the same mapping would apply, a reference to another server to which the client should send a query, and error messages indicating problems with interpretation of location information. 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 request a short response that contains only the mapping data, omitting region information. In some operational environments a UDP-based transport may be available and MAY be used to confirm or update data already available.

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. Server Discovery

There are likely to be a variety of ways that clients can discover appropriate LoST servers, including DHCP, SIP device configuration, or DNS records for their signaling protocol domain, e.g., the AOR domain for SIP. The appropriate server depends on, among other considerations, who operates LoST services, including the Internet Service Provider (ISP), Voice Service Provider (VSP), or the user's home domain. A DNS based approach utilizing the S-NAPTR mechanism is specified in [6].

5. Query

LoST provides the ability to use civic or geospatial location information in the query message 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 findLoSTByCivic and the findLoSTByGeo query. Geospatial location information uses GML format [9] and civic location information utilizes the format defined in [10]. Hence, the location format is not defined in this document but references already available standards.

5.2. Service Identifier Element

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

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"?>
<findLoSTByGeo
    validate="false"
    xmlns:p2="http://www.opengis.net/gml"
    xmlns="urn:ietf:params:xml:ns:lost1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

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

</findLoSTByGeo>
```

Figure 2: 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' service.

```
<?xml version="1.0"?>
<findLoSTByCivic
    validate="true"
    xmlns="urn:ietf:params:xml:ns:lost1"
    xmlns:p2="urn:ietf:params:xml:ns:pidf:geopriv10:civilLoc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <civicLocation>
        <p2:country>Germany</p2:country>
        <p2:A1>Bavaria</p2:A1>
        <p2:A3>Munich</p2:A3>
        <p2:A6>Neu Perlach</p2:A6>
        <p2:HNO>96</p2:HNO>
        <p2:PC>81675</p2:PC>
    </civicLocation>
    <service>urn:service:sos.police</service>

</findLoSTByCivic>
```

Figure 3: 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' service. The query also indicates that validation is desired.

6. Response

A response message might either be a responseGeo or a responseCivic depending on the type of query message. If the query message was a findLoSTByCivic then the response will be a responseCivic. If a findLoSTByGeo message was sent as a query then the response will be a findLoSTByGeo. The location information that is provided by the response message depends on the query and refers to the service boundary as described in Section 6.3.

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 Element

Each displayName element contains a string that is suitable for display. displayName elements are of type "text" as described in Section 6.7.

6.3. Region Element

Each region element contains either one or more civic location elements derived from the GeoPriv civic address schema or feature.xsd expression from GML.

6.4. Dialstring Element

Each dialstring element contains from one to sixteen digits. Note that a Tel URI may also contain the same target, expressed in a different format; see RFC 3966 [11].

6.5. TimeToLive Attribute

Each timeToLive attribute is a positive integer, expressing the validity period of the response in seconds.

6.6. Validated Element

Each validated element contains a string which is composed by by concatenating the elements from the request which have been recognized as valid by the server.

6.7. text Attribute

This is a text type suitable for internationalized human readable text.

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"?>
<responseGeo
    timeToLive="10000"
    xmlns="urn:ietf:params:xml:ns:lost1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p2="http://www.opengis.net/gml">

    <displayName>New York City Police Department</displayName>
    <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>
    <uri>sip:nypd@example.com</uri>
    <uri>xmpp:nypd@example.com</uri>
    <dialstring>911</dialstring>

</responseGeo>
```

Figure 4: Response Message Example using Geospatial Location Service Boundary Hints

This example shows a reponse with two URIs for the previously queried service URN. Information about the service boundary is provided in the Polyon. The <dialstring> element indicates the valid dialstring for the expressed location and service URN.

```
<?xml version="1.0" encoding="UTF-8"?>
<responseCivic
    timeToLive="10000"
    xmlns="urn:ietf:params:xml:ns:lost1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p2="urn:ietf:params:xml:ns:pidf:geopriv10:civilLoc">

    <displayName>Munich Police Department</displayName>
    <region>
        <p2:country>Germany</p2:country>
        <p2:A1>Bavaria</p2:A1>
        <p2:A3>Munich</p2:A3>
    </region>
    <validated>country A1 A3 A6 PC</validated>
    <uri>sip:munich-police@example.com</uri>
    <uri>xmpp:munich-police@example.com</uri>
    <dialstring>110</dialstring>

</responseCivic>
```

Figure 5: 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 <region> element and information about the validated civic address fields. The timeToLive indicates that the returned information can be cached for 10000 seconds and provides a displayName with additional, textual information about the returned information.

7. Miscellaneous Functionality

7.1. List Service Query

This subsection describes a query that offers the LoST client to query for available service identifiers supported by the LoST server.

```
<?xml version="1.0" encoding="UTF-8"?>
<listServices
    xmlns="urn:ietf:params:xml:ns:lost1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

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

</listServices>
```

Figure 6: Example for a List Service Query

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

7.2. Response to a List Service Query

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 in the query.

```
<?xml version="1.0" encoding="UTF-8"?>
<returnServices
    timeToLive="10000"
    xmlns="urn:ietf:params:xml:ns:lost1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <service>urn:service:sos.ambulance</service>
    <service>urn:service:sos.animal-control</service>
    <service>urn:service:sos.fire</service>
    <service>urn:service:sos.gas</service>
    <service>urn:service:sos.mountain</service>
    <service>urn:service:sos.marine</service>
    <service>urn:service:sos.physician</service>
    <service>urn:service:sos.poison</service>
    <service>urn:service:sos.police</service>
    <service>urn:service:sos.suicide</service>

</returnServices>
```

Figure 7: Example for the Response to a List Service Query

This response corresponds to the query of Figure 6.

8. 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[7].

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

Figure 8: 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"?>
<findLoSTByCivic
    validate="true"
    xmlns="urn:ietf:params:xml:ns:lost1"
    xmlns:p2="urn:ietf:params:xml:ns:pidf:geopriv10:civilLoc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <civicLocation>
        <p2:country>US</p2:country>
        <p2:A1>New York</p2:A1>
        <p2:A3>New York</p2:A3>
        <p2:A6>Broadway</p2:A6>
        <p2:LOC>Suite 75</p2:LOC>
        <p2:PC>10027-0401</p2:PC>
    </civicLocation>
    <service>urn:service:sos.police</service>

</findLoSTByCivic>
```

Since the contacted LoST server has the requested information available the following response is returned. The response 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 'validated' 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 returned service boundary indicates that all of New York City would result in the same response. The dialstring element indicates that the service can be reached via the dial string 9-1-1.

```
<?xml version="1.0" encoding="UTF-8"?>
<responseCivic
    timeToLive="10000"
    xmlns="urn:ietf:params:xml:ns:lost1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p2="urn:ietf:params:xml:ns:pidf:geopriv10:civilLoc">

    <displayName>New York City Police Department</displayName>
    <region>
        <p2:country>US</p2:country>
        <p2:A1>New York</p2:A1>
        <p2:A3>New York</p2:A3>
    </region>
    <validated>country A1 A3 A6 PC</validated>
    <uri>sip:nypd@example.com</uri>
    <uri>xmpp:nypd@example.com</uri>
    <dialstring>911</dialstring>

</responseCivic>
```

9. Deployment Methods

Because services for emergency contact resolution may differ depending on local or service needs, this document only specifies the "wire format" for LoST services and explicitly leaves open the possibility for many different types of deployment.

For instance:

During discovery, a client may be directed to issue all queries to an LoST service completely authoritative for a given jursidiction.

A client may be directed to issue queries to an LoST server that acts as a reflector. In such a case, the LoST server analyzes the query to determine the best server to wich to refer the client.

Or the client may be directed to a server that performs further resolution on behalf of the client.

A LoST service may also be represented by multiple LoST servers, either grouped together or at multiple network locations. Using S-NAPTR [13], clients may be given a list of multiple servers to which queries can be sent for a single service.

For instance, the service at emergency.example.com may advertise LoST service at local1.emergency.example.com, local2.emergency.example.com, and master.emergency.example.com. Each server may given a different preference. In this case, 'local-1' and 'local-2' may be given a lower preference (more preferred) than 'master', which might be a busier server or located further away.

```
+-----------+             pref 10 +-----------+
|           |-------------------->+           |
|  client   |------               |  local-1  |
|           |---   \              |           |
+-----------+   \   \             +-----------+
                 \   \
                  \   \           +-----------+
                   \   \  pref 10 |           |
                    \   --------->|  local-2  |
                     \            |           |
                      \           +-----------+
                       \
                        \                           +-----------+
                         \                  pref 20 |           |
                          ------------------------->|  master   |
```

```
                                                       |           |
                                                       +-----------+
```

10. XML Schema

This section provides the XML schema used by LoST.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:ietf:params:xml:ns:lost1"
xmlns:lost="urn:ietf:params:xml:ns:lost1"
xmlns:civilLoc="urn:ietf:params:xml:ns:pidf:geopriv10:civilLoc"
xmlns:gml="http://www.opengis.net/gml"
xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
elementFormDefault="qualified" attributeFormDefault="unqualified">

     <annotation>
       <documentation>
         A schema for a Location to Service Translation Protocol
       </documentation>
     </annotation>

     <!--             -->
     <!-- Query types -->
     <!--             -->

     <!--  Abstract Query            -->

    <complexType name="queryType"/>
     <element name="query" type="lost:queryType" abstract="true"/>


     <!--  findLoSTByCivic             -->

     <element name="findLoSTByCivic" type="lost:findLoSTByCivicType"
             substitutionGroup="lost:query"/>

          <complexType name="findLoSTByCivicType">
          <complexContent>
               <extension base="lost:queryType">
                    <sequence>
                         <element name="civilAddress"
                                  type="civilLoc:civilAddress"
                                  minOccurs="0" maxOccurs="1"/>
                         <element name="service" type="anyURI"
                                  minOccurs="1" maxOccurs="1"/>
                    </sequence>
                    <attribute name="validate"
                               type="boolean" default="false"/>
               </extension>
```

```
             </complexContent>
        </complexType>


        <!-- findLoSTByGeo        -->

        <element name="findLoSTByGeo" type="lost:findLoSTByGeoType"
                 substitutionGroup="lost:query"/>

        <complexType name="findLoSTByGeoType">
             <complexContent>
                  <extension base="lost:queryType">
                       <sequence>
                            <element ref="gml:location"
                                     minOccurs="0" maxOccurs="1"/>
                            <element name="service" type="anyURI"
                                     minOccurs="1" maxOccurs="1"/>
                       </sequence>
                       <attribute name="validate"
                                  type="boolean" default="false"/>
                  </extension>
             </complexContent>
        </complexType>

        <!-- listServices             -->

        <element name="listServices" type="lost:listServicesType"
                 substitutionGroup="lost:query"/>

       <complexType name="listServicesType">
             <complexContent>
                  <extension base="lost:queryType">
                          <sequence>
                            <element name="service" type="anyURI"
                                     minOccurs="1" maxOccurs="1"/>
                   </sequence>
                  </extension>
             </complexContent>
        </complexType>


        <!--              -->
        <!-- Responses    -->
        <!--              -->


        <!--  Abstract Response            -->
```

```
        <element name="result" type="lost:resultType" abstract="true"/>

        <complexType name="resultType">
                  <attribute name="timeToLive" type="positiveInteger"
                             use="required" />
        </complexType>


       <!--  emergencyContact Response           -->

        <element name="responseCivic" type="lost:responseCivicType"
                 substitutionGroup="lost:result"/>

        <complexType name="responseCivicType">
             <complexContent>
                  <extension base="lost:resultType">
                       <sequence>
                            <element name="displayName"
                                     type="normalizedString"
                                     minOccurs="0" maxOccurs="1"/>
                            <element name="civilAddress"
                                     type="civilLoc:civilAddress"
                                     minOccurs="0" maxOccurs="1" />
                            <element name="uri" type="anyURI"
                                     minOccurs="0"
                                     maxOccurs="unbounded" />
                            <element name="dialstring"
                                     type="normalizedString"
                                     minOccurs="0" maxOccurs="1" />
                       </sequence>
                  </extension>
             </complexContent>
        </complexType>


        <element name="responseGeo" type="lost:responseGeoType"
                 substitutionGroup="lost:result"/>

        <complexType name="responseGeoType">
             <complexContent>
                  <extension base="lost:resultType">
                       <sequence>
                            <element name="displayName"
                                     type="normalizedString"
                                     minOccurs="0" maxOccurs="1"/>
                            <element ref="gml:Polygon"
                                     minOccurs="0" maxOccurs="1"/>
                            <element name="uri" type="anyURI"
```

```
                                     minOccurs="0"
                                     maxOccurs="unbounded"/>
                           <element name="dialstring"
                                    type="normalizedString"
                                    minOccurs="0" maxOccurs="1" />
                       </sequence>
                  </extension>
             </complexContent>
        </complexType>


         <element name="returnServices" type="lost:returnServicesType"
                  substitutionGroup="lost:result"/>

        <complexType name="returnServicesType">
             <complexContent>
                  <extension base="lost:resultType">
                       <sequence>
                        <element name="service" type="anyURI"
                                 minOccurs="1" maxOccurs="unbounded"/>
                       </sequence>
                  </extension>
             </complexContent>
        </complexType>



        <!--                 -->
        <!-- Error responses -->
        <!--                 -->

        <element name="genericCode" type="lost:codeType"
                 abstract="true"/>

        <element name="invalidCivicData" type="lost:codeType"
                 substitutionGroup="lost:genericCode"/>

        <element name="invalidGeoData" type="lost:codeType"
                 substitutionGroup="lost:genericCode"/>

        <element name="invalidService" type="lost:codeType"
                 substitutionGroup="lost:genericCode"/>

        <complexType name="codeType">
             <sequence minOccurs="0" maxOccurs="unbounded">
                  <element name="explanation">
                       <complexType>
                            <simpleContent>
```

```
                                 <extension base="string">
                                      <attribute name="language"
                                                 type="language"
                                                 use="required"/>
                                 </extension>
                            </simpleContent>
                       </complexType>
                  </element>
             </sequence>
        </complexType>
   </schema>
```

11. 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.

12. IANA Considerations

TBD, such as namespace registrations.

13. 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, LoST RECOMMENDS the use of channel security, such as TLS.

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

[Editor's Note: A future version of this document will describe the countermeasures based on the security requirements outlined in[4].]

14. Open Issues

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

15. References

15.1. 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", RFC 2119, BCP 14, March 1997.

[4] Taylor, T., "Security Threats and Requirements for Emergency Call Marking and Mapping", draft-ietf-ecrit-security-threats-01 (work in progress), April 2006.

[5] Schulzrinne, H. and R. Marshall, "Requirements for Emergency Context Resolution with Internet Technologies", draft-ietf-ecrit-requirements-10 (work in progress), June 2006.

[6] Schulzrinne, H., "A Uniform Resource Name (URN) for Services", draft-ietf-ecrit-service-urn-03 (work in progress), May 2006.

[7] 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.

[8] Mealling, M., "The IETF XML Registry", draft-mealling-iana-xmlns-registry-03 (work in progress), November 2001.

[9] OpenGIS, "Open Geography Markup Language (GML) Implementation Specification", OGC OGC 02-023r4, January 2003.

[10] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, December 2005.

15.2. Informative References

[11] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.

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

progress), October 2005.

[13] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.

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

Intellectual Property Statement

Disclaimer of Validity

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.

Copyright Statement

Acknowledgment

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