Internet Engineering Task Force	R. Barnes	
Internet-Draft	M. Lepinski	
Intended status: Standards Track	BBN Technologies	
Expires: December 5, 2009	June 03, 2009	

TOC

Using Imprecise Location for Emergency Context Resolution
draft-barnes-ecrit-rough-loc-03.txt

Status of this Memo

Copyright Notice

Abstract

Emergency calling works best when precise location is available for emergency call routing. However, there are situations in which a location provider is unable or unwilling to provide precise location, yet still wishes to enable subscribers to make emergency calls. This document describes the level of location accuracy that providers must provide to enable emergency call routing. In addition, we descibe how

emergency services and non-emergency services can be invoked by an endpoint that does not have access to its precise location.

Table of Contents

---

1. Introduction

TOC

Information about the location of an emergency caller is a critical input to the process of emergency call establshment. Endpoint location is used to determine which Public Safety Answering Point (PSAP) should be the destination of the call. (The entire emergency calling process is described in detail in [1] (Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, “Framework for Emergency Calling using Internet Multimedia,” July 2009.) and [2] (Rosen, B. and J. Polk, “Best Current Practice for Communications Services in support of Emergency Calling,” January 2010.).) This process is most likely to work properly when the endpoint is provided with the most accurate precise information available about its location. Using location information with maximal precision and accuracy minimizes the chance that a call will be mis-routed. And when that location is provided to the endpoint, the endpoint is able to verify that the location is correct (to the extent of the endpoint's knowledge of its own location) prior to an emergency call, and is able to perform emergency call routing functions on its own, providing redundancy for network-provided functions.

However, there may be situations in which it is not feasible for endpoints to be provided with maximally precise and accurate location. These cases may arise when computing precise location is an expensive or time-consuming operation (e.g., in the case of wireless triangulation), and location is needed quickly (as is often the case in emergency situations). Or they may arise because the policy of the location provider does not allow precise location to be provided to the endpoint (e.g. due to privacy considerations). While it is undesirable to use imprecise location for emergency call routing, the possibility that precise location may not be available to the calling device must be accomodated in order to make emergency calling possible in the largest possible set of circumstances.

This document is concerned imprecise location only in the context of routing emergency calls, i.e., for determining the correct PSAP to receive a given call (e.g., via a LoST query [3] (Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol," August 2008.)). (More generally, the provided location information will be needed to route the call to an entity that is authorized to request precise location, e.g., an Emergency Services Routing Proxy.)

Location information may also be used in the emergency calling framework to direct the dispatch of emergency responders. This usage is treated separately from call routing for purposes of this document, and this document does not place requirements on the location provided for dispatch (although it should obviously be as precise as possible). The only provision for dispatch in this document is a recommendation that the location provider supply endpoints with a URI that can be used by a PSAP or other emergency authority to obtain a different location for use in dispatch, hopefully more precise than the one used for routing.

This document describes the use of imprecise location information in the emergency call routing system. Section 3 (Determining sufficient location precision) describes how location providers can determine the precision necessary to support emergency call routing, and how they can use this information to optimize location delivery. Section 4 (Requesting emergency and non-emergency services) describes how emergency calls are placed in such an environment, and how non-emergency services can be invoked when precise location is not available to the endpoint by value.

---

2. Terminology

TOC

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 [4] (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).

We consider in this document patterns of interaction as described in [1] (Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, “Framework for Emergency Calling using Internet Multimedia,” July 2009.). The two main parties of interest are endpoints and location providers. Endpoints are hosts connected to the Internet that originate emergency calls in the emergency calling architecture, while location providers are entities that supply location information that is used for emergency calling. In addition, we will discuss how these parties interact with the LoST mapping infrastructure [7] (Schulzrinne, H., “Location-to-URL Mapping Architecture and Framework,” March 2009.), and with emergency and non-emergency location-based service providers. For convenience, we say that location information (either in LoST queries or in service boundaries) is provided "in geodetic form" if it is provided in the "geodetic-2d" location profile, and "in civic form" if it is provided in the "civic" profile.

---

3. Determining sufficient location precision

TOC

A location provider wishing to provide location information usable for emergency call routing requires a mechanism for determining when a description of location (e.g., a polygon) is precise enough to be used for emergency call routing. This mechanism might be used to decide when to terminate a positioning mechanism that converges over time, or to choose a polygon larger than the known location of the endpoint (in order to obscure the known location of the endpoint), while preserving the utility of the location for emergency call routing.
There are two base requirements for a location to be usable for emergency call routing:

1. The location SHOULD be sufficiently precise that a LoST request with the location and any service URN will return a unique URI mapping value. This may not be possible in all cases, e.g., because of overlapping service boundaries (leading to areas that do not have a unique mapping) or positioning limitations (leading to insufficient precision).

2. When the location of the endpoint is known by the provider to greater precision than is being provided, the provided location MUST return the same mappings from LoST (for all service URNs) as the known location.

3. When the location of the endpoint is known by the provider to greater precision than is being provided, the provided location MUST contain the precise location (as a geographic subset).

In this section, we describe how to use a "location filter" to determine whether a given location is usable for emergency call routing, and how to construct and maintain such a filter.

3.1. Location filtering

TOC

With each service-to-URI mapping, a LoST query provides a service boundary that represents the set of locations in which that mapping is valid. A consequence of this is that given a set of service boundaries for difference services (say, one mapping "urn:service:sos.fire" to "sip:fire@example.com" and one mapping "urn:service:sos.police" to "sip:police@example.com"), the intersection of those service boundaries is the region in which two mappings are valid ("urn:service:sos.fire" maps to "sip:fire@example.com" and "urn:service:sos.police" maps to "sip:police@example.com"). Outside that area, one or more of the mappings is invalid. Said differently, any region contained in an intersection uniquely determines mappings for the services used in the intersection, and any two locations within the same intersection are equivalent for the purpose of LoST mapping (i.e., emergency call routing).
A location filter is thus a set of regions (optionally, each region may be assigned a list of LoST mappings), as illustrated in Figure 1 (Generating a filter from service boundaries). Each region is the intersection of the service boundaries for all services available within the region, and the lists represent the mappings that are valid within that region. A filter is used to determine whether a location is useable for emergency call routing in the following way:

1. The location SHOULD be contained in exactly one of the regions in the filter. This guarantees that LoST mappings are unique.

2. When the precise location of the endpoint is known, the provided location MUST be contained in the same region(s) of the filter as the known location. This guarantees that LoST queries with the provided location return the same results as those done with the known location.

3. When the precise location of the endpoint is known, the provided location MUST contain the precise location (as a geographic subset).

When the regions are bound to lists of URN-URI mappings, the resulting filter can also be used as a cache for LoST mappings; the LoST mappings for a location are the mappings bound to the region(s) containing it.

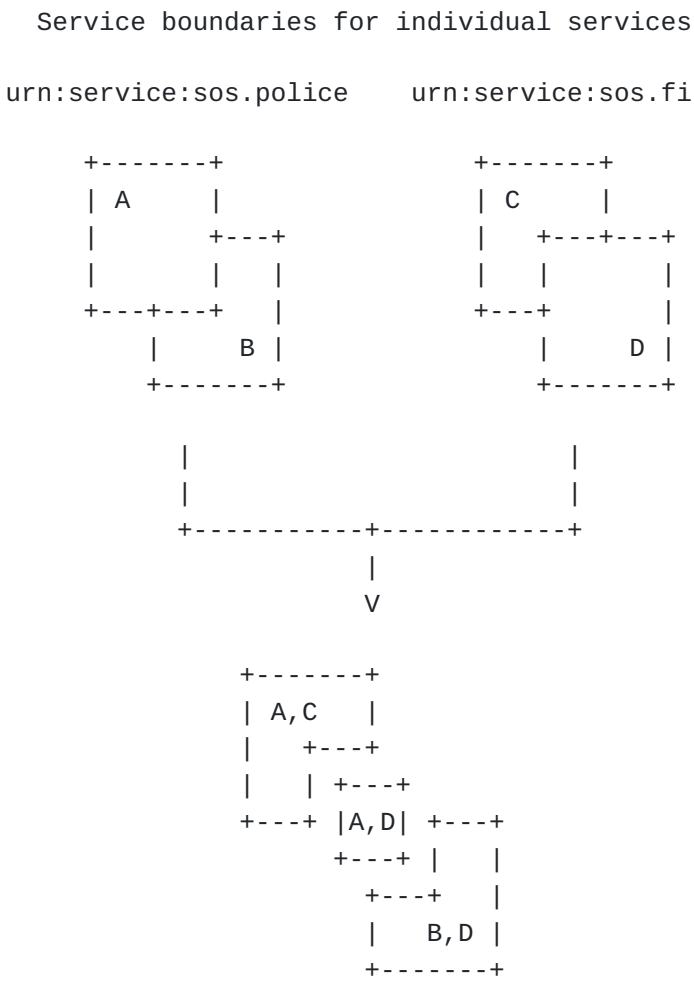

Figure 1: Generating a filter from service boundaries

When the location of the endpoint is known to more precision than the location provided to the endpoint, although any location meeting the two criteria above is equivalent to the known location for purposes of LoST, the provided location MUST contain the known location in order to avoid errors if the location is used for other purposes in the course of an emergency (e.g., if the location is provided to first responders for dispatch). This guarantee also allows the endpoint to do some course verification that the provided location is correct (in order to prevent very gross errors in routing). Thus, any location that (1) contains the known location and (2) is contained in the same filter region as the known location is allowable. Locations that also are contained in only one filter region are preferred. Adding randomness to the provided locations may have privacy benefits in some cases, as discussed in the security considerations below.

3.2. Constructing location filters

TOC

For simplicity, we assume that the entity performing filtering will only be using the filter to test locations contained within a particular geographic "coverage area". (In principle, this coverage area could be the entire world, but assuming a more limited coverage area allows for a filter to be built more quickly) Given a coverage area and the ability to act as a LoST client, a location service provider can autonomously compute a location filter using the following algorithm:

First, the server must obtain mappings and service boundaries for all services and for all points within the coverage area. For each emergency service URN, the server goes through the following process to build a service map: First, the server queries LoST for the complete coverage map for the desired service. This can be done with a LoST <findService> query of the following form:

```
<?xml version="1.0" encoding="UTF-8"?>
<findService
     xmlns="urn:ietf:params:xml:ns:lost1"
     serviceBoundary="value">
  <location>
    <!-- Coverage Area -->
  </location>
  <service>
    <!-- Service URN -->
  </service>
</findService>
```

If LoST returns a set of mappings whose service boundaries cover the coverage area (i.e., if LoST is configured to return all possible matches for the queried location), then the process termintes here. The coverage map for this service is the set of returned service boundaries.

If service boundaries in the LoST response to the above query do not cover the location provider's coverage area, then the location server must perform further queries. The location included in each query is the difference between the coverage area and the current coverage map, that is, the coverage area with all currently-known service boundaries removed. The server repeats this process (query, then remove service boundaries from the query location, then query again) until either (1) the coverage area is covered by the collected service boundaries or (2) LoST returns a <notFound> error.

After the location server has performed this procedure for each service, it will have a set of LoST mappings for each service, for every point in its coverage region where that service is offered.
The regions in the location filter are computed separately for service boundaries provided in civic form and in geodetic form. If all service boundaries are provided in one form (e.g., if all boundaries are provided in geodetic form, even if some are also provided in civic form), the server MAY perform the algorithm for that form. If both algorithms are being performed, and some mappings provide both civic and geodetic service boundaries, the server MUST input those mappings to both the civic and geodetic computations.

3.2.1. Geodetic service boundaries

TOC

The regions in the location filter are computed from these mappings by iterating over URI tuples: For each service URN, let uris(urn) be the set of PSAP URIs for that service URN (collected from the mappings). The set of URI tuples is then the cartesian product of these sets; if the set of servuce URNs is {urn1,...,urnN}, then the set of URI tuples is uris(urn1) x ... x uris(urnN). The server computes the regions in the filter by iterating through the set of URI tuples, either by constructing the set of URI tuples and directly iterating, or by using nested iteration through all the sets uris(urn).
For each URI tuple, the server MUST compute the intersection of the service boundaries for the URIs in the tuple. This becomes an entry in the location filter: The stored region is the intersection of the service boundaries, and the corresponding mapping table is the list of (URN, URI) pairs, where the URIs are the URIs from the tuple and the URNs are the services used to obtain them from LoST. (Empty filter regions, corresponding to URIs in a tuple with disjoint service boundaries, can of course be discarded.)

3.2.2. Civic service boundaries

TOC

As in the case of geodetic location, regions of a civic address filter are computed based on URI-tuples. For tuples where all mappings have the same service boundary, that service boundary MUST be used as the filter region for that type. For all other cases (i.e., tuples with different civic locations), the regions of the filter must be computed as the intersections of the locations according to an algorithm that is determined by local addressing standards.
Note that the resulting filter regions SHOULD still cover the location server's coverage area, i.e., there should be a filter region that contains every civic address within the coverage area. In particular,

the server SHOULD NOT use a specific address to represent a filter region: Such an address would not include many points in the service region (i.e., it would not meet the third rules from both lists of rules above). If the server chooses to return a civic address that does not, then it MUST set the 'method' element of the PIDF-LO it returns to value 'area-representative' registered in Section 7 (IANA Considerations).

3.3. Maintaining location filters

TOC

As the LoST mappings that underlie the filter change, the filter will need to be updated. The entity maintaining the filter MUST obtain a new mapping for a region when an existing mapping expires. The service boundary from the new mapping is compared to the service boundary from the old mapping: If they are the same, then the filter need not be updated. If they differ, then regions in the filter that intersect either the old service boundary or the new service boundary will need to be recomputed. Note that since this operation only requires the server to determine if two service boundaries are identical, the server need only store a hash of the old boundary (to which it can compare a hash of the new boundary).

3.4. Applying location filters

TOC

After constructing a location filter, a location server can use it to optimize how it delivers location. When the location server is using a positioning algorithm that grows more accurate with time, the filter tells it how long to run the algorithm. Namely, the algorithm can be terminated when the estimated location is within one of the regions in the filter.
When the location provider knows the precise location of the caller, a location filter can also be used as a "location cache". That is, the location provider can simply look up which of the filter regions contains the caller's precise location and return that region as the caller's location (or some subset that contains the precise location). This allows an additional optimization in some cases: If the location server knows that the caller's precise location will be within the same region for a period of time, it can instruct the client not to re-query in that time. For instance, if the server is delivering location over HELD, then it can use the HTTP cache-control headers (e.g., Expires). However, the location server MUST NOT instruct the client to wait for longer than the current filter is valid; the expiry time of the location MUST be before the earliest expiry of a LoST mapping used in the filter.

4. Requesting emergency and non-emergency services

TOC

When a location provider wishes to deliver endpoints location information that is below its maximum available precision while still supporting emergency calling, it MUST provide to the endpoint both a location (by value) that is sufficient for emergency call routing (see above) and a location reference (i.e., a URI) that can subsequently be used by authorized parties to obtain more precise information about the location of the endpoint. The endpoint then can then use both the location value and the location reference to request location-based services (LBS) as described below.

4.1. Emergency calling

TOC

The procedure for placing an emergency call is indentical to that described in [1] (Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, “Framework for Emergency Calling using Internet Multimedia,” July 2009.). In particular, the endpoint requirements in Sections 8 and 9 of [2] (Rosen, B. and J. Polk, “Best Current Practice for Communications Services in support of Emergency Calling,” January 2010.) still apply to an endpoint that receives imprecise location.

In addition, an endpoint that receives location both by value and by reference from its location provider MUST include both the location value and the location reference in the SIP INVITE message that initiates an emergency call, as specified in [5] (Polk, J. and B. Rosen, “Location Conveyance for the Session Initiation Protocol,” March 2009.). When the endpoint supports LoST, it SHOULD use the location value to obtain a PSAP URI for LoST queries (as opposed to attempting to dereference the location reference). Note that the caller would also have to add the "used-for-routing" parameter to the geolocation header that points to the location value as inserted into the INVITE message. Note that this process crucially relies on the location value having sufficient precision for routing emergency calls (see Section 3 (Determining sufficient location precision) for techniques to ensure the location value is suitable for emergency call routing).

When a PSAP receives a SIP INVITE that contains both a location value and a location reference, if the value is too imprecise for use in dispatch then the PSAP SHOULD dereference the LbyR to obtain more precise information. In turn, the location provided by the location provider MUST allow access by all PSAPs whose service boundaries overlap with the region served by the location provider. This means

that either the provider must supply a reference that can be dereferenced by any party, or else the provider must establish explicit authentication and authorization relationships with all PSAPs in its service area.

4.2. Non-emergency services

TOC

Non-emergency LBSs will generally require more precise information than is required for emergency call routing. Therefore, when requesting a non-emergency LBS, the endpoint SHOULD include the location reference provided by its location provider, and MAY additionally provide the location value. If the provided location value is not sufficiently precise to deliver the requested service, then the LBS provider should then dereference the location value to request location information of sufficient precision from the location provider. If the dereference fails, then the request for service may fail as well.
Note that when the location reference provided by the location provider is access-controled, this dereference may require a pre-existing authentication and authorization agreement between the LBS provider and the location provider. In such a case, the endpoint may not know whether a given non-emergency service is authorized to obtain the endpoint's precise location using the location reference. The endpoint is always capable of requesting services without knowing whether they are authorized; in this way, the endpoint can discover authorized services by trial and error. In order to simplify this process, a location provider may supply the endpoint with references to authorized service providers, although there is currently no standard protocol for this transaction.

---

5. Acknowledgements

TOC

This document generalizes the concept of "rough location" that was originally discussed in the context of the location hiding problem. This concept was put forward by Henning Schulzrinne and Andy Newton, among many others, in a long-running ECRIT discussion.

---

6. Security Considerations

TOC

The use of rough location to support emergency calling enables a location provider to provide low-precision location with low assurance (e.g., of requestor identity) and high-precision location with higher

assurance. The fact that lower-precision location has lower value -- to location providers and LBS providers as a commercial asset, and to targets as private information -- this trade-off allows a location provider to avoid the cost of protecting location with high-assurance access controls when this location has low value.

However, in order to support emergency services, this expense cannot be avoided entirely. Because PSAPs require high-precision location for emergency response planning, a location provider that normally provides rough location MUST provide a location URI that a PSAP can use to obtain high-precision location. This constraint means that the provided URI MUST have either no access control at all or a policy that allows access by appropriate PSAPs (and other emergency response systems, e.g., ESRPs). That is, if such a location URI is access controlled, then the location provider MUST be able to authenticate requests from PSAPs.

One reason for a location server to provide location information below its maximum precision is to protect the privacy of the target. Some location provisioning protocols do not enable the location provider to obtain strong assurance of the identity of the location recipient; in particular, the location provider may be unable to verify that the recipient is the target of the location being provided. Therefore, there is a risk that a sufisticated attacker might be able to spoof the identifier (e.g. IP address) used by the location provider to identify the target, and obtain the target's location in this way. One way to mitigate this risk is to provide only imprecise location information to the end-point (without authentication), and to provide precise information only to trusted entities that can authenticate themselves to the location provider. Additionally, in some deployment scenarios, location providers have concerns about the comprimise of endpoint devices. Providing only imprecise location to the endpoint, prevents malware on a comprised device from obtaining the precise location of the target.

As described in Section 3.1 (Location filtering) above, the location provider choosing to provide a less precise location than a known location has a significant amount of choice in deciding which location to provide: Any location that contains the known location and is in the same filter region will do. When the provider is reducing precision for privacy purposes, there is a signficant benefit to choosing a random location meeting these criteria. If a watcher is interested in whether or not the endpoint is moving, an imprecise location may still reveal that fact if it is constant when the endpoint is at rest. If the provided location is randomized each time it is provided, then the watcher is unable to obtain even this level of information.

TOC

7. IANA Considerations

This document requests that IANA register a new PIDF-LO 'method' token in the registry defined by RFC 4119 [6] (Peterson, J., "A Presence-based GEOPRIV Location Object Format," December 2005.)

> **area-representative:** Location chosen as a representative of a region in which the target is located; may not be the target's location

---

8. References

TOC

8.1. Normative References

TOC

[1]	Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling using Internet Multimedia," draft-ietf-ecrit-framework-10 (work in progress), July 2009 (TXT).
[2]	Rosen, B. and J. Polk, "Best Current Practice for Communications Services in support of Emergency Calling," draft-ietf-ecrit-phonebcp-14 (work in progress), January 2010 (TXT).
[3]	Hardie, T., Newton, A., Schulzrinne, H., and H. Tschofenig, "LoST: A Location-to-Service Translation Protocol," RFC 5222, August 2008 (TXT).
[4]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[5]	Polk, J. and B. Rosen, "Location Conveyance for the Session Initiation Protocol," draft-ietf-sip-location-conveyance-13 (work in progress), March 2009 (TXT).
[6]	Peterson, J., "A Presence-based GEOPRIV Location Object Format," RFC 4119, December 2005 (TXT).

8.2. Informative References

TOC

[7]	Schulzrinne, H., "Location-to-URL Mapping Architecture and Framework," draft-ietf-ecrit-mapping-arch-04 (work in progress), March 2009 (TXT).

Authors' Addresses

TOC

	Richard Barnes
	BBN Technologies
	9861 Broken Land Pkwy, Suite 400
	Columbia, MD 21046
	USA
Phone:	+1 410 290 6169
Email:	rbarnes@bbn.com
	Matt Lepinski
	BBN Technologies
	10 Moulton St
	Cambridge, MA 02138
	USA
Phone:	+1 617 873 5939
Email:	mlepinski@bbn.com