Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2013

R. Barnes
M. Lepinski
BBN Technologies

July 16, 2012

Using Imprecise Location for Emergency Context Resolution
draft-ietf-ecrit-rough-loc-05.txt

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 describe additional rules for networks and endpoints to enable emergency calling by endpoints that do not have access to precise location.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 17, 2013.

Table of Contents

1. Introduction

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 [5] and [1].) This process is most likely to work properly when the endpoint is provided with the most accurate and precise information available about its location. Using location information with maximal precision and accuracy minimizes the chance that a call will be mis-routed.

When 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. Moreover, when endpoints have access to location information, they can look up PSAP contact information themselves, reducing dependence on other call-routing elements in the network, and increasing the overall resilience of the system.

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 a location provider does not allow precise location to be provided to the endpoint. 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.

To put it another way, a need for emergency calling with imprecise location can arise in two ways. Either the location of the endpoint is not known to the location provider with a high degree of precision, or the endpoint's precise location is known and the location provider chooses to provide location with lower precision. In the former case, the techniques described in this document can be used to determine whether a given positioning mechanism provides sufficient precision to support emergency calling. In the latter case, such techniques can be used to determine how much a location value can be "fuzzed" before it becomes unusable for emergency services.

This document is concerned with 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 [2]). Depending on the the structure of the local emergency service network, the location information provided to the endpoint may also be used to route the call to an entity that is authorized to request precise location, e.g., an Emergency Services Routing Proxy. The requirements and processes described in this document are the same for both cases. Detailed requirements are discussed in [6]

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 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 5 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

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].

We consider in this document patterns of interaction as described in [5]. 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], 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" LoST location profile, and "in civic form" if it is provided in the "civic" profile.

The term "precision" is not used in a quantitative sense in this document. In general, the "precision" of a location value is determined by the size of its uncertainty region. [8] Higher precision values have small uncertainty regions and lower precision values have larger uncertainty regions. The notion of "sufficient precision for emergency services is defined in Section 3

3. Location Precision Requirements

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 process 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 three basic 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 emergency service URN will return a unique URI mapping value. This may not be possible in all cases, e.g., because of overlapping service boundaries creating areas with non-unique mappings, or because of positioning limitations that prevent sufficiently precise positioning.

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 emergency 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).

4. Location Filtering

In effect, the first of these rules divide the world into regions where each point is served by the same set of emergency services (i.e., the LoST mappings are the same). We call this division of

space a "location filter" and the consituent regions of uniformity "filter regions". The second rule says that the rough location must be in the same filter region as the precise location. (The third rule is unrelated to filtering.)

A location filter is a collection of geographical regions satisfying the following criteria:

1. For any location value that is a subset of a filter region, a LoST request for any service will return a unique mapping result.

2. Any two locations within the same filter region receive the same LoST results for all services

Given a location filter, it is easy to determine when a given location value is sufficiently precise, or to create a less precise version of location that is still precise enough. Namely, a location value is precise enough when if fits within a given filter region, and any superset of a location value (e.g., a polygon containing a point) can be used as a less precise version of the location value, as long as it still fits within the same filter region.

4.1. Filter Region for a Known Location

A simple fuzzing algorithm that maintains sufficient precision for emergency services is to replace a given location value with the filter region that contains it. Given a known location, a location server can compute a filter region using a series of LoST queries.

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 different services, the intersection of those service boundaries is the region in which all of the corresponding mappings are valid. If one service boundary corresponds to the area where "urn:service:sos.fire" is served by "sip:fire@example.com" and another maps "urn:service:sos.police" to "sip:police@example.com", then the intersection is the area where both of these 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. So as was suggested above, the intersection of two service boundaries defines a set of mappings, and any two locations within that intersection are equivalent for the purpose of LoST mapping (i.e., emergency call routing).

Filter regions can be deduced constructed from LoST mappings for a sample location by intersecting all the service boundaries for

services available at that point. Figure 1 illustrates how the filter region containing the point X is the intersection of the service boundaries for police and fire services that serve X.

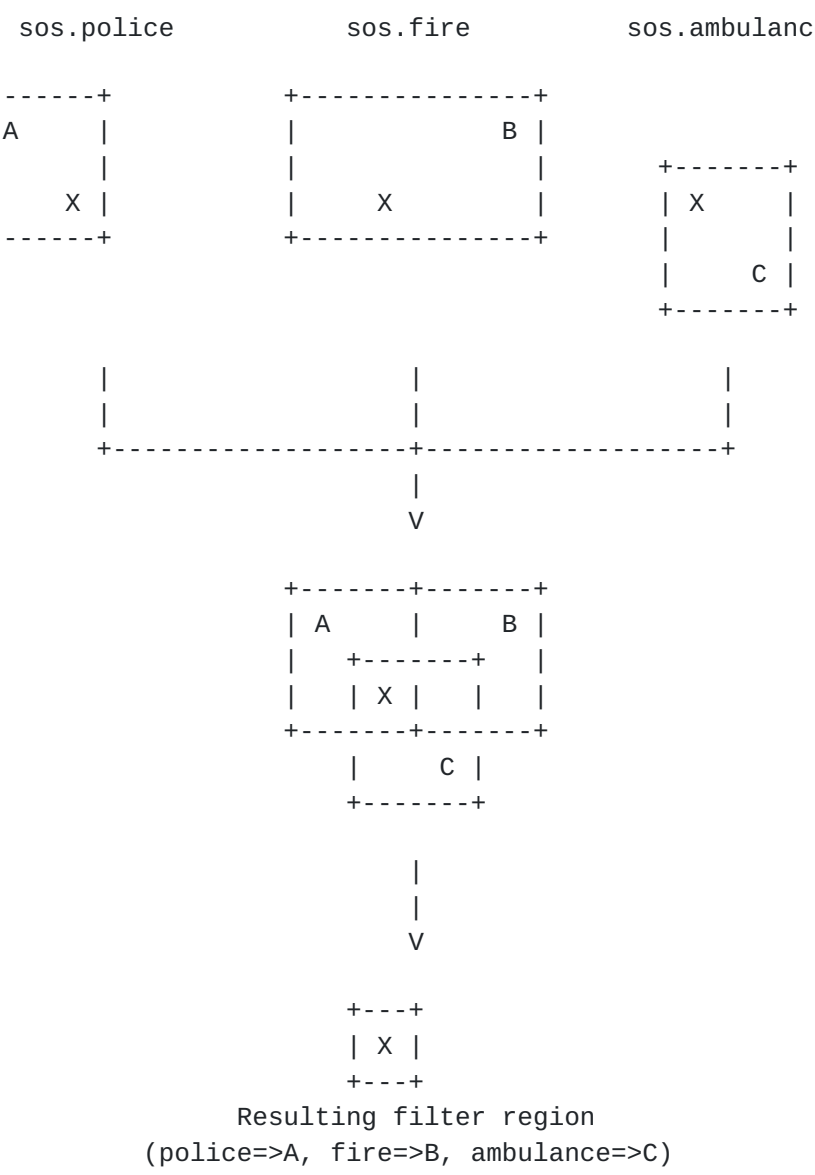

Figure 1: Generating a Filter Region from a Sample Point

In pseudocode form, algorithm for constructing a filter region from a point is as follows:

```
function filterRegion(X):
Set REGION = the world
For each service URN S in the urn:service:sos namespace
    Perform a LoST <findService> query for Y and S
    If LoST returned an error
        Continue
    Set SB = <serviceBoundary> from LoST <findServiceResponse>
    If SB is not provided, throw an error
    Else set REGION = intersection( REGION, SB )
Return REGION
```

It is important that the filter take into account all emergency services available in over the coverage area of the LIS. (That is, the services listed in the LoST serviceList elements.) The feature is necessary in order to ensure that calls to all available emergency services can be routed correctly using rough location values provided by the filter.

While in principle, a location server could execute this algorithm to compute a fresh filter region on each query, it is much more efficient to use the offline algorithm for computing an entire location filter, described in the next section.

4.2. Constructing a Location Filter

When a location server knows ahead of time that it will be providing rough location values, it can pre-compute a location filter that contains all the filter regions for locations it's concerned with. Once the filter has been computed (as an off-line computation), the filter region for a given precise location can be found by searching for the pre-computed region that contains the precise location. When precise location is not known, a complete filter can be used to test evaluate the utility of an imprecise location by determining the degree to which it overlaps with each filter region.

For example, a simple fuzzing algorithm that maintains sufficient precision for emergency services is to replace a given location value with the filter region that contains it. This way, the server can compute the filter off-line (as described below), then provision the location of each possible device by storing a pointer to the filter region that contains the device's location.

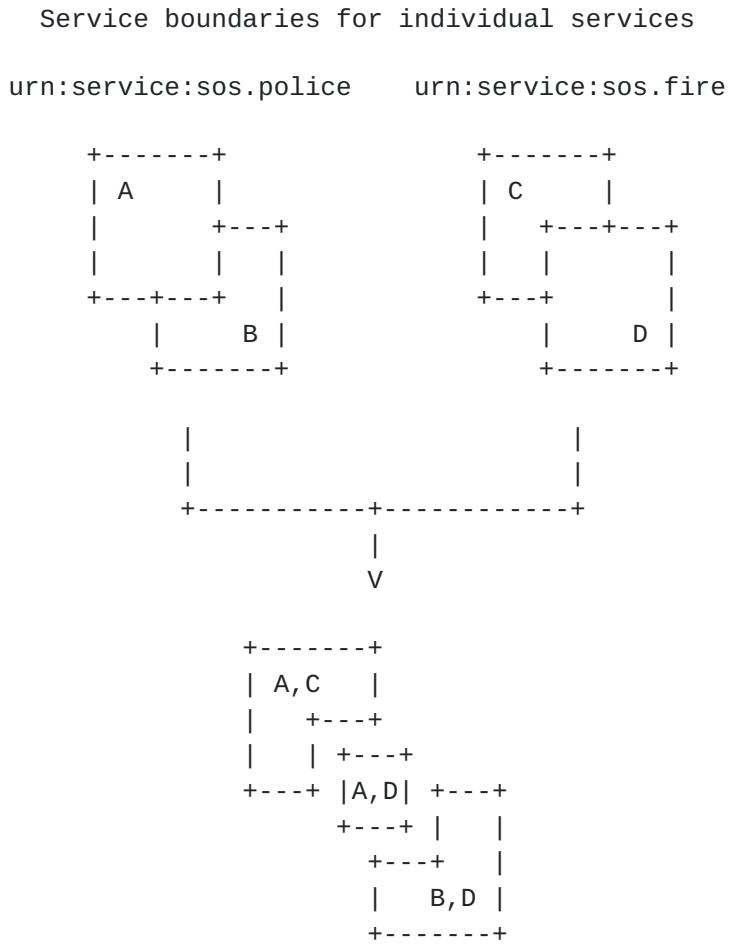

Figure 2: Generating a Filter from Service Boundaries

The filter regions in a filter can are constructed by taking intersections of service boundaries. Figure 2 shows a simple location filter: Starting with a set of four service boundaries for two different services. The filter that results from taking intersections of these boundaries has three regions:

1. A region where police calls are directed to A and fire calls are directed to C.

2. A region where police calls are directed to A and fire calls are directed to D.

3. A region where police calls are directed to B and fire calls are directed to D.

These regions satisfy the criteria for a location filter because each

one has a unique set of mappings and those mappings are valid across the entire region. The service regions for B and C do not overlap -- there is no place where police calls go to B and fire calls to C -- so there is no (B,C) region.

More generally, a filter region is the intersection of the service boundaries for all services available within the region. A filter can be used to determine whether a location is usable 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).

Practically speaking, a location filter is built up by computing filter regions for sample points, using the algorithm described above. In the example of Figure 2, one would need to sample three points: One in the (A,C) region, one in the (A,D) region and one in the (B,D) region. The overall algorithm thus samples random points until the computed filter regions cover the desired area. (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)

```
function filter(LS_AREA):
Set FILTER = the empty set
Set COVERAGE = the empty set
Set ERR_COUNT = 0
While COVERAGE < LS_AREA && ERR_COUNT < 100
    Choose a random uncovered point X in LS_AREA
    Compute R = filterRegion(X)
    If R != the empty set
        Add R to FILTER
        Set COVERAGE = union(COVERAGE, R)
    Else
        ERR_COUNT += 1
```

Return FILTER

If the server also stores the lists of URN-URI mappings for each region, then the 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.

If the LoST servers have been provisioned properly then this algorithm will terminate successfully. If LoST mappings do not cover a point X, then the filterRegion(X) will return the empty set, and the algorithm will give up after 100 such queries. This limit on queries introduces some risk that a small covered area will be left out of the filter and marked as uncovered; if this is a concern, then the query limit can be increased, or the algorithm can be explicitly directed to sample certain specific points.

Of course, if the location server operator has information about service boundaries through some channel other than LoST, then the LoST queries above can be replaced by queries to a local store of mapping information. The choice of random points can also be guided to ensure that all mapped areas are covered even if there are some uncovered areas. The location server can also cache service boundaries acquired during the algorithm to avoid unnecessary LoST queries.

4.3. Civic Address Considerations

This algorithm actually results in two filters -- one for geodetic service boundaries and one for civic service boundaries -- since civic and geodetic boundaries cannot be directly compared or intersected. It is RECOMMENDED that location servers always compute a geodetic filter for use with emergency services, since the notion of civic service boundaries have some inherent ambiguity.

Indeed, the notion of intersection of civic service boundaries has some dependence on the jurisdiction within which the service boundaries are defined. Civic service boundaries are comprised of a set of <civicAddress> elements, each defining a set of civic addresses that are within the boundary, namely those that match the civic elements provided.

When computing the intersection of two civic service boundaries, any <civicAddress> elements that are shared between the two service boundaries MUST be included in the resulting intersection. When two <civicAddress> elements in the service boundaries being compared are different from each other, then their intersection must be computed according to 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 the lists of rules above). If the server creates a PIDF-LO document describing a civic address that does not contain the precise location of the device, then it MUST set the 'method' element of the PIDF-LO it returns to value 'area-representative' registered in Section 8.

If the above procedures are not workable for a given scenario, then the server MAY fall back to geocoding of service boundaries, in which case the above procedures for geodetic location can be used.

Geocoding may also be necessary in cases where the location of the target is known as a civic address with limited granularity, with which service boundaries do not align. For instance, the target's location may be known to the level of a postal code, but postal code regions may span multiple PSAP service areas or filter regions. In such cases, the server MAY geocode the target's location to obtain a rough geodetic position, which can be dealt with as discussed in Section 4.6 below.

4.4. Privileged LoST Servers

In certain scenarios, it may be possible that some LoST servers are authorized to access more precise location information than networks are willing to reveal to endpoints. For example, a network operator might allow a LoST server operated by a national authority to access an endpoint's precise location, even if it does not allow the endpoint access to this information.

In these situations, the precision required for emergency services routing is different than in the base case considered above. Rather than needing location precise enough to identify a given PSAP, the location value provided to the endpoint only needs to be precise enough to route the LoST request through the mapping infrastructure to a LoST server that is authorized to access the endpoint's precise location. Once the request reaches that server, the server can request more precise location information and use that information to route the request further.

Indeed, such privileged LoST server could even be operated by the network itself. In this case, if an endpoint complies with requirement ED-52 in [1], it will send its LoST request directly to the network-provided LoST server, which can look up the client's

location and return mappings. However, it is possible that clients will use an alternative LoST resolver, so it is still beneficial to provide a rough location that can route the request to a nearby privileged LoST server.

In cases where a network allows one or more privileged LoST servers to access precise location information, the network MAY designate a location that is precise enough to reach one of these LoST servers as precise enough for emergency call routing.

This document does not specify how these privileged LoST servers could obtain more precise location information from network operators. One possible solution is to extend LoST to carry a location reference in addition to a location by value.

4.5. Maintaining Location Filters

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.

4.6. Applying Location Filters

After constructing a location filter, a location server can use it to optimize how it delivers location. How this is done depends on whether the location server is trying to reduce the precision of a known precise location, or trying to determine whether an imprecise position is good enough for emergency services.

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 caching strategy 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.

When the location server starts with imprecise location, there are different ways to apply the filter, depending on the positioning technique being used. For example with a positioning algorithm that grows more accurate with time, the filter can tell the server how long to run the algorithm -- the algorithm can be terminated when the estimated location (that is, an uncertainty region containing the device's location) is within one of the regions in the filter.

A location filter can also be used to test whether a database of rough locations for IP addresses (as is commonly used for web localization today) contains precise enough values for use with emergency services. To make this determination, each value in the database woud be tested to see if it falls mostly or entirely within a given filter region. Note, however, that this test does not address concerns about the accuracy of location information, i.e., the probability that the caller is actually contained within the specified uncertainty region.

Note that the requirements for containment in a filter region differ between these two use cases. When precise location is known, the rough location that is returned MUST be contained within a single filter region; otherwise, there will be an increased risk of mis-routing. When the location server starts with imprecise location, it may choose location values that are not entirely within one filter region. The distribution of the imprecise location value among filter regions corresponds to the risk that LoST routing will provide incorrect information, so the choice of location value should balance the risk of incorrect routing against the additional time needed to obtain more precise location (which can translate to a delay in call setup). In this case, it may not even be possible for a location server to return a location value that is entirely within a single filter region.

5. Additional Requirements for Networks and Endpoints

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 a location (by value) that is sufficient for emergency call routing (as defined above) and MUST provide 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 emergency services and other location-based services (LBS).

This arrangement allows the client to provide rough location (by value) to any entity, and to provide precise location (by reference) to authorized parties. An assumption of this model, of course, is that emergency authorities are authorized by the location provider to receive precise location. Location providers may also authorize other entities to receive precise location information (e.g., commercial services that have agreed to pay for location). Authorization policy for location URIs is set by the referenced location server; a mechanism for clients to request information about this policy is described in [9].

ED-84: When an endpoint has access to location both by value and by reference, it MUST include both forms of geolocation in the SIP INVITE message initiating an emergency call, each in a separate Geolocation header. The endpoint SHOULD include a Geolocation-Routing header with the value "yes".

AN-30: Networks providing imprecise location MUST also provide location by reference.

AN-31: Networks providing imprecise location SHOULD provide DHCP-based LoST discovery, advertising a LoST server that is authorized to dereference location URIs issued by the network's LIS.

The overall procedure for placing an emergency call is identical to that described in [5]. In particular, the endpoint requirements in Sections 8 and 9 of [1] still apply to an endpoint that receives imprecise location, with the above modification (use of the location URI in LoST).

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 [10]. Note that this process crucially relies on the location value having sufficient precision for routing emergency calls (see Section 3 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, and 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 SHOULD 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. It is RECOMMENDED that location providers establish similar relationships with other PSAPs in adjoining jursidictions -- even if their service regions do not overlap with the location provider's -- in case such a PSAP needs access to precise location information, for example, if it is acting as a backup for one of the location provider's normal PSAPs.

6. Acknowledgements

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. Thanks to Hannes Tschofenig and Martin Thomson for detailed reviews of this document.

7. Security Considerations

The use of imprecise location provides a security trade-off for location providers. When location providers are required to provide location in support of emergency services, they have to balance that requirement against the risk that location information will be disclosed to an unauthorized party. The use of location configuration protocols inherently introduces some risk that an entity other than the device will be able to masquerade as the device (e.g., another host behind the same NAT or malicious software on the host) [11]. In some cases, the location provider may not authorize the device itself to access precise location. At the same time, because endpoints can roam between networks, it is operationally difficult to have strong client authentication for LCPs.

Using of rough location to support emergency calling enables a location provider to provide low-precision location with low assurance (e.g., without client authentication)and high-precision location with higher assurance. Because lower-precision location generally has lower value -- to location providers and LBS providers as a commercial asset, and to devices 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, location providers cannot provide only low-precision location; they also have to provide

PSAPs with access to high-precision location information. Because PSAPs require high-precision location for emergency response, a location provider that normally provides imprecise location to clients MUST also provide them 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.

The use of location by reference introduces some risk that the reference will be used by an attacker to gain unauthorized access to the device's location. These risks are not specific to emergency service, however; general risks and mitigations for location by reference are discussed in [12]

As described in Section 4 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 some privacy 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. An algorithm for securely fuzzing a device's location can be found in [13]; for emergency services, the additional constraint must be added that the fuzzed location must remain in the same filter region as the original.

8. IANA Considerations

This document requests that IANA register a new PIDF-LO 'method' token in the registry defined by RFC 4119 [4]

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

9. References

9.1. Normative References

[1] Rosen, B. and J. Polk, "Best Current Practice for Communications Services in support of Emergency Calling", draft-ietf-ecrit-phonebcp-20 (work in progress), September 2011.

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

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

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

9.2. Informative References

[5] Rosen, B., Schulzrinne, H., Polk, J., and A. Newton, "Framework for Emergency Calling Using Internet Multimedia", RFC 6443, December 2011.

[6] Schulzrinne, H., Liess, L., Tschofenig, H., Stark, B., and A. Kuett, "Location Hiding: Problem Statement and Requirements", RFC 6444, January 2012.

[7] Schulzrinne, H., "Location-to-URL Mapping Architecture and Framework", RFC 5582, September 2009.

[8] Thomson, M. and J. Winterbottom, "Representation of Uncertainty and Confidence in PIDF-LO", draft-thomson-geopriv-uncertainty-07 (work in progress), March 2012.

[9] Barnes, R., Thomson, M., Winterbottom, J., and H. Tschofenig, "Location Configuration Extensions for Policy Management", draft-barnes-geopriv-policy-uri-02 (work in progress), November 2010.

[10] Polk, J., Rosen, B., and J. Peterson, "Location Conveyance for the Session Initiation Protocol", RFC 6442, December 2011.

[11] Tschofenig, H. and H. Schulzrinne, "GEOPRIV Layer 7 Location Configuration Protocol: Problem Statement and Requirements", RFC 5687, March 2010.

[12] Marshall, R., "Requirements for a Location-by-Reference

Mechanism", RFC 5808, May 2010.

[13] Schulzrinne, H., Tschofenig, H., Cuellar, J., Polk, J., Morris, J., and M. Thomson, "Geolocation Policy: A Document Format for Expressing Privacy Preferences for Location Information", draft-ietf-geopriv-policy-26 (work in progress), June 2012.

[14] Thomson, M. and K. Wolf, "Describing Boundaries for Civic Addresses", draft-thomson-ecrit-civic-boundary-02 (work in progress), January 2011.

Authors' Addresses

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