HTTP Enabled Location Delivery (HELD)
draft-winterbottom-http-location-delivery-01.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on January 19, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This document describes a means by which an IP-device may request location from a Location Server using HTTP as a GeoPriv 'using protocol'.
Table of Contents

1. Introduction .............................................. 4
   1.1 Paradigm ........................................... 4
2. Terminology ............................................... 5
3. Overview ................................................ 6
4. Location Server Discovery .............................. 8
   4.1 DHCPv4 ............................................ 8
   4.2 DHCPv6 ............................................ 8
   4.3 DNS Service Record ................................ 9
5. Protocol Specifics ....................................... 10
   5.1 HELED Protocol Stack .............................. 10
   5.2 Location Information Request Parameters ........ 10
      5.2.1 exact ......................................... 12
      5.2.2 locationURI ................................... 12
      5.2.3 locationType .................................. 12
      5.2.4 updateURI ...................................... 13
      5.2.5 updateResponseTime ............................ 13
      5.2.6 pidf-lo ......................................... 13
      5.2.7 rulesetURI ...................................... 14
      5.2.8 ruleset ......................................... 15
      5.2.9 retentionURI .................................... 15
      5.2.10 responseTime ................................... 16
      5.2.11 signed .......................................... 16
      5.2.12 ip ............................................. 16
      5.2.13 hwaddr ......................................... 17
6. IP-device References ..................................... 18
7. Location Assertion ....................................... 19
8. Location Information Response Message ............... 20
9. Authentication ........................................... 21
   9.1 IP-device Authentication ............................ 21
   9.2 OBO Authentication .................................. 21
   9.3 Location Client Authentication ..................... 21
10. Session Management ...................................... 22
11. Security Considerations ............................... 23
12. IANA Considerations .................................... 24
13. Acknowledgements ...................................... 25
14. Normative references ................................... 25
   Authors’ Addresses ..................................... 25
A. Examples .................................................. 28
   A.1 Simple Location Request ............................ 28
      A.1.1 Request for Any Location ...................... 28
      A.1.2 Request for Civic Location .................... 29
   A.2 Location URI Request .............................. 31
      A.2.1 Request locationURI Specifying rulesetURI .... 31
      A.2.2 Requesting locationURI Specifying ruleset .... 32
      A.2.3 Location-Client Using locationURI ............ 34
   A.3 Location Assertion .................................... 35
1. Introduction

This document describes a protocol that employs secure HTTP as a GeoPriv 'using protocol' to deliver location information relating to a IP-device from a Location Server. The IP-device discovers or is informed of the Location Server and establishes a session with the server. The Location Server constructs a PIDF-LO document and returns this to the IP-device. This protocol facilitates requesting specific forms of location including geodetic, postal civic, jurisdictional civic and that the Location Server sign the location as specified in [I-D.thomson-domain-auth].

This document concentrates solely on the provision of location information, it does not describe how location may be generated. The advantage of this is that this enables the use of this protocol in many different networks. This aligns this document with the model described in [RFC3693] by separating the Location Server and Location Generator functions and focussing on only the Location Server.

Currently proposed location delivery mechanisms, such as those defined in [RFC3825] and [I-D.ietf-geopriv-dhcp-civil], tightly couple location determination (generation) and location acquisition. While this coupling offers some benefits, it also has drawbacks, particularly where DHCP is not employed to any great degree, or if the form of location or its container evolve to encompass additional information, for example signatures as proposed in [I-D.thomson-domain-auth] or the existing provided-by field, where support for these will necessitate changes to the base protocol.

1.1 Paradigm

The philosophical approach for requesting a location on which this document is based, is that the network does not know the identity of a user requesting a location, but it can have some assurances as to which access point the location request is coming from. That is, location is a resource or an attribute being used in a network and the network can identify the resource being used. This is somewhat analogous to an IP address being allocated via DHCP, that is to say that the DHCP server does not know the identity of the user requesting the address, but it can be sure that it has allocated an address in response to a request from an entity physically present in the access network it serves.

The Location Server is assumed to have the facilities of a Location Generator as part of its own implementation or available in the access networks serviced by the Location Server. The form and nature of the Location Generator are not discussed in this document.
2. Terminology

The key conventions and terminology used in this document are defined as follows:

This document reuses the terms Location Server, Location Generator and Using-Protocol as defined in [RFC3693]. In some specification the Location Server is referred to as a Location Information Server or LIS.

Target The device that the location is attributed to.

IP-device Used interchangeably with Target.

OBO On-Behalf-Of. A network node that is able to request location on-behalf-of an IP-device.

Location-Client A third-party requesting the location of an IP-device using a locationURI.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
3. Overview

This document describes how an IP-device requests location information from a Location Server, including how a Location Server is identified.

Location Server discovery is achieved through the use of a new DHCP option, a new DNS SRV record or static configuration. A Location Server is identified by a URI that identifies the Location Server as an HTTP service using the "https" scheme.

The IP-device initiates an HTTP connection with the Location Server. If the IP-device only requires location information with no particular constraints it uses the HTTP GET method, otherwise it includes request parameters using the HTTP POST method.

The Location Server identifies the IP-device based on its IP address. The Location Server then identifies a Location Generator for the IP-device, which determines the location of the IP-device. The Location Server will take the generated location and produce a PIDF-LO that is subsequently returned to the requesting IP-device.

In addition to returning a location to an IP-device requesting its own location, the protocol provides support for a location URI, which enables by-reference distribution of location information by the IP-device. This has a number of benefits which are described in more detail by [I-D.winterbottom-location-uri].

An option exists for specially authorized entities to request location on-behalf-of (OBO) an IP-device. In this situation the requester includes the IP address of the target IP-device; the Location Server uses this IP address to determine location, rather than the source IP address of the request message. The Location Server in this circumstance must have a pre-arranged trust relationship with the requester otherwise the request MUST fail.
4. Location Server Discovery

Location Server discovery MAY occur using a DHCP option, a DNS SVC record, or a preconfigured URL; in order of preference.

4.1 DHCPv4

The DHCPv4 option includes a list of URIs; the first URI must be attempted first and subsequent URIs contacted only in the event of a problem in retrieving location information. Each URI must reference a service that is able to provide the IP-device with location information.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  LOCSERV_URI  |    Length     |    URI List ...               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code LOCSERV_URI (TBD)

Length The length of the URI list in octets

URI List A list of one or more URIs separated by the space character

4.2 DHCPv6

The DHCPv6 option is similarly formatted to the DHCPv4 option.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      OPTION_LOCSERV_URI       |         option-len            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           URI List ...                        .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

option-code OPTION_LOCSERV_URI (TBD)

option-len The length of the URI list in octets

URI List A list of one or more URIs separated by the space character
4.3 DNS Service Record

A new service is to be created "locserv+http" such that the querying entity can recover the FQDN for the local Location Server. This service MUST be configured such that an HTTP POST to this address will result in the Location Server application being invoked. The DNS server entry will therefore look similar to:

\_locserv+http\_tcp SRV 1 0 0 location-server.example.com.

The new service record needs to be registered with IANA, see Section 12 for more details.

The DNS response indicates a fully qualified domain name for the Location Server. In order to derive a URI from this FQDN, the "https" scheme SHOULD be used with the root path ("/"). This allows the corresponding client to construct a URL as follows:

https://location-server.example.com/
5. Protocol Specifics

5.1 HELED Protocol Stack

HELED is a web services application that uses the POST and GET methods of HTTP. TLS is used as a secure transport layer to ensure that HELED meets the requirements of a GEOPRIV 'using protocol'. TLS for transporting HTTP is defined [RFC2818].

```
+------------+
|    HELED   |
+------------+
|    HTTP    |
+------------+
|    TLS     |
|    TCP     |
+------------+
|    IP      |
+------------+
```

HELED relies on the security features of TLS to ensure that location information is properly protected. Specifically, the encryption and authentication provided by TLS must be enabled.

5.2 Location Information Request Parameters

A HELED location information request is either an HTTP GET or POST [RFC2616] to the discovered Location Server URI. Where the requesting entity needs to request specific information from, or provide specific information to the Location Server the POST method MUST be used.

A request may be made by one of three types of entity: an IP-device wishing to determine its own location, a location-client requesting the location of a IP-device, or a element requesting location on-behalf-of (OBO) an IP-device. Request parameters are included in the body of a POST request using either URL-encoding or the "multipart/form-data" encoding defined in [RFC1867]. The same type of request is made from the IP-device, an OBO and a location-client, the only difference being the URI to which the requests are posted.

The Location Server identifies the IP-device by a different means for each request source. For requests from the IP-device, the source IP address of the connection is used. Requests from the location-client SHOULD be to a unique URI that includes information that the Location Server can use to identify a particular IP-device. Where an OBO
requests the location of an IP-device, the IP address of the IP-device is included as a parameter and additional information MAY be included as necessary.

The protocol structure of HELD suitably lends itself to support the cascading of location information requests between Location Servers. This accommodates complex network structures by enabling distribution of location functions across servers.

HELD does not explicitly prohibit any parameter from being included in any request, however the practical applications of the parameters dictate their suitability more to specific request types. The following table lists the parameters that may have practically meaning to the Location Server when passed from the requesting node types:

<table>
<thead>
<tr>
<th></th>
<th>IP-device</th>
<th>OBO</th>
<th>Location-Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>locationURI</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>locationType</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>updateURI</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>updateResponseTime</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>responseTime</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>pidf-lo (multipart)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ruleset (multipart)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rulesetURI</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>retentionInterval</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ip</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>hwaddr</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>signed</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

The following sub-sections describe each of the parameters in detail.
5.2.1 exact

The "exact" parameter may have a value of "true" or "false". If not included the default value is "false".

When set to "true" all parameters included in the location request MUST be satisfied by the Location Server, or the response from the Location Server MUST indicate failure.

5.2.2 locationURI

The "locationURI" parameter indicates if a location URI is required. The value space of this parameter is the superset of the duration type specified in [W3C.REC-xmlschema-2-20041028], plus the string "unbounded". When omitted, or set to a zero or negative duration, no location URI SHALL be generated or returned.

When a location URI is requested the Location Server provides a URI that references the location of the IP-device. Uses and reasons for location by-reference are described in [I-D.winterbottom-location-uri].

A location URI is provided in the body of the response using the "text/plain" MIME type. If a "locationURI" is requested then the Location Server MUST return a "locationURI" or fail.

The value of this parameter indicates the maximum amount of time that the requester wishes the location URI to be valid for; a Location Server MAY specify a shorter duration, but MUST NOT provide a longer duration than that requested.

The inclusion of a "locationURI" parameter overrides the inclusion of the "locationType" and "exact" parameters. The response MUST yield a "locationURI" or an error.

5.2.3 locationType

A form parameter that may have multiple values. The valid values are:

any The Location Server may return any information it has available. This value is assumed as the default if no "locationType" is specified.

geodetic The Location Server SHOULD return a geodetic location for the IP-device.
jurisdictionalCivic The Location Server SHOULD return a jurisdictional civic address for the IP-device.

postalCivic The Location Server SHOULD return a postal civic address for the IP-device.

civic The Location Server SHOULD return a civic address for the IP-device. Any type of civic address may be returned. The location server SHOULD ignore this value if a request for "jurisdictionalCivic" or "postalCivic" has been made and can be satisfied.

The Location Server SHOULD attempt to provide the location in the requested form unless the request cannot be properly satisfied. If the "exact" parameter is set to "true" then the Location Server MUST provide the location in the requested form or fail.

The Location Server SHOULD provide location-information types in the same order in which they are requested.

5.2.4 updateURI

The "updateURI" parameter MAY be provided by an IP-device that is capable of determining its own location. This provides the Location Server with a means of interrogating an IP-device to determine the IP-device’s current location.

A location update URI SHOULD use the "https" scheme so that HELD may be employed to request information.

5.2.5 updateResponseTime

A form parameter provided to the Location Server in conjunction with an "updateURI" as an indication of how long an IP-device could take to respond to a request for location information.

The "updateResponseTime" is indicative only and is a positive decimal value representing the approximate number of seconds that it could take to obtain a response. The Location Server MUST ignore this value if no corresponding "updateURI" is provided. It is RECOMMENDED that systems support millisecond precision for this parameter.

5.2.6 pidf-lo

A file describing a location provided by an IP-device, and the general format of PIDF-LOs that the IP-device wishes the Location Server use when providing location information.
The "pidf-lo" is transferred to the location server using the multipart file transfer, with the MIME type of "application/pidf+xml".

The Location Server SHOULD perform a location assertion against ALL locations provided in the PIDF-LO document. The Location Server MUST only include location information that has been successfully asserted. If the "exact" parameter is set to "true" then ANY location failing an assertion test results in the entire request failing.

The Location Server will replace the following entities in any provided PIDF-LO with values that it derives or determines:

- "presentity"
- "timestamp"
- "retention-expiry"
- "method"
- "provided-by"

If the "exact" parameter is set to "true", then these values MAY be changed, but the Location Server MUST NOT insert new elements if they are not already present.

The "pidf-lo" SHOULD contain a "method" element.

If the IP-device wishes explicit rules to be included in any PIDF-LO documents containing its location, then it MUST include them in the "pidf-lo" sent to the Location Server. The Location Server will preserve, but not adhere to, any of these rules. The Location Server will adhere to rules provided in either the "rulesetURI" or "ruleset" parameters discussed in subsequent sections.

5.2.7 rulesetURI

A form parameter that defines a URI to a set of policies and rules describing how and to whom an IP-device’s location may be provided.

The "rulesetURI" is the preferred means by which an IP-device informs the Location Server to whom the IP-device’s location may be divulged. If a "rulesetURI" is provided to the Location Server then the Location Server MUST adhere to the rules referenced. If a "rulesetURI" and a "ruleset" are both provided then ONLY the "rulesetURI" MUST be used.
Rules documents MUST comply with [I-D.ietf-geopriv-common-policy] and [I-D.ietf-geopriv-policy]. It is RECOMMENDED that a ruleset URI use an "https" scheme.

If the IP-device does not provide a "rulesetURI", or a "ruleset" then the Location Server MUST apply a default ruleset. The default "ruleset" SHALL prohibit the Location Server from providing location information to any unauthorized location-client. The default "ruleset" MAY include specific third-parties as mandated by local regulations, for example emergency services.

If the IP-device provides a "rulesetURI" that is not accessible, contains invalid rules, or cannot be parsed by the Location Server then the Location Server SHOULD reject the request and return an error.

5.2.8 ruleset

A file containing policies and rules describing how and to whom an IP-device’s location may be provided. It is transferred to the Location Server using multipart file transfer, with a MIME type of "application/auth-policy+xml".

If a "ruleset" is provided to the Location Server then the Location Server MUST adhere to the rules, unless it is also provided with a "rulesetURI", in which case the rules referenced by the "rulesetURI" are used. Rules documents MUST comply with [I-D.ietf-geopriv-common-policy] and [I-D.ietf-geopriv-policy].

If the IP-device does not provide a "rulesetURI", or a "ruleset" then the Location Server MUST apply a default ruleset. The default "ruleset" SHALL prohibit the Location Server from providing location information to any unauthorized location-client. The default "ruleset" MAY include specific third-parties as mandated by local regulations, for example emergency services.

If the IP-device provides a "ruleset" that is invalid, or cannot be parsed by the Location Server then the request MUST be rejected and an error returned.

5.2.9 retentionInterval

A form parameter provided to the Location Server to specify the length of time that should be allowed for the "retention-expiry" element that SHOULD be applied to all locations provided for this IP-device.

The "retentionInterval" is an positive decimal value specifying a
duration in seconds. When a Location Server serves a request for location, the resulting PIDF-LO document will have a "retention-expiry" element set to the specified number of seconds after the issue of the location information.

If no "retentionInterval" is specified the Location Server SHOULD provide a default value for the "retention-expiry" element in all PIDF-LO documents. Nominally this default SHOULD be 24 hours from the receipt of the location request as specified in [I-D.ietf-geopriv-pidf-lo]. The Location Server MAY use a different value as circumstances dictate; for example, where the access network supports mobility, this value can be reduced.

5.2.10 responseTime

A form parameter that is sent to the Location Server indicating how long a requester is prepared to wait for location information.

The "responseTime" is a positive decimal value representing the time in seconds that a client is prepared to wait for a location response from the Location Server. It is RECOMMENDED that systems support millisecond precision for this parameter.

The Location Server SHOULD provide the most accurate location that it can within the response time requested. The Location Server may use this parameter to aid it in making decisions about which location determination method it will invoke if more than one is supported. If this parameter is not provided then the Location Server may use any mechanism for location determination at its disposal. The value used in this parameter is indicative to the Location Server only, and the Location Server is under no obligation to strictly adhere to it, any strict enforcement of behaviour around this time is left to the requester.

5.2.11 signed

A form parameter that may have a value of "true" or "false". If not included the default value is "false".

The "signed" parameter allows the device to request a signed location object as defined in [I-D.thomson-domain-auth]. A value of "true" indicates a request for a signed location. If not present a value of "false" MUST be assumed, and the location MUST NOT be signed.

5.2.12 ip

A form parameter used by an OBO to specify the IP address of the IP-device for which it is requesting location information.
Any entity including this parameter in a location request MUST have an explicit trust relationship with the Location Server. Any request from a node not meeting these requirements MUST fail.

5.2.13 hwaddr

A form parameter used by an OBO to specify the hardware address of an IP-device for which it is requesting location information.

Any entity including this parameter in a location request MUST have an explicit trust relationship with the Location Server. Any request from a node not meeting these requirements MUST fail.
6. IP-device References

The location URI is an identifier generated by the Location Server so that the IP-device’s location may be retrieved by-reference. Location URIs are discussed in more detail in Appendix A.2.2.

There is no requirement for the Location Server to know the identity of the user of an IP-device. However the Location Server is responsible for providing assurances that location information is attributed to a specific IP-device at a given point in time. To satisfy these requirements, the Location Server MUST generate a pseudonym for the presentity URI associated with the IP-device. Any pseudonym created by the Location Server SHOULD NOT reveal the true identity of the IP-device or user.

The Location Server MUST be able to uniquely identify the IP-device based on a location URI or presentity pseudonym. Identity matching SHOULD occur through internal correlation only; it SHOULD NOT be possible to perform a computational transform on these data to identify the IP-device.
7. Location Assertion

Location assertion is most applicable where an IP-device can
determine its own location. In this situation the device may be able
to provide additional or more precise information than the Location
Generator available to the Location Server. The location assertion
mechanism allows the IP-device to tell the Location Server where it
thinks it is and by comparing this location against that provided by
a Location Generator, the Location Server may assert that the
location provided by the IP-device is better, or more precise.
Conversely if the location provided by the IP-device is significantly
different to that provided by the Location Generator then the
Location Server SHOULD fail the assertion.

If the Location Server is unable to assert the location provided by
the IP-device as being valid then the Location Server MUST do one of
two things depending on the value of the "exact" parameter. If
"exact" parameter is set to "true" then the Location Server MUST
return an error to the IP-device. If the "exact" parameter is set to
"false", then the Location Server MUST return the location provided
by the Location Generator.
8. Location Information Response Message

The location information response message contains either a "text/plain" body containing a location URI or a "application/pidf+xml" body containing a valid PIDF-LO document. HTTP features SHOULD be used to provide additional information, including the "Content-Type", "Expires" and "Date" headers.

The following HTTP status codes MAY be used to convey additional information about the request:

200 OK The Location Server MAY return this code if the request was processed successfully; the body of the response contains the requested data.

400 Bad request The Location Server MAY return this code if the request was badly formatted, or required parameters were incorrect, out of range or missing. This code MAY be returned if any data type does not correctly parse, including any XML content.

401 Unauthorized The Location Server MAY return this code if a requester is not authorized to access the requested service. For example, a Location Server may return this if a requester uses client identification parameters without the required permissions.

403 Forbidden The Location Server MAY return this code if a location client does not meet the criteria specified in the authorization policy. A Location Server SHOULD use the 404 code instead of 403 to prevent a location client from discovering that a location URI is in use.

404 Not Found The Location Server MAY return this code when the Location Server URI or location URI is not correct. A Location Server MAY also use this return code when location information cannot be determined, or a session has expired.

406 Not Acceptable The Location Server MAY return this code where the client uses "Accept" header that does not allow for the content characteristics that the Location Server is capable of providing. A request for location URI MUST include an "Accept" header that includes "text/plain", or "*". A request for location information MUST include an "Accept" header that includes "text/xml", "application/xml" "application/pidf+xml", or "*".

501 Not Implemented The Location Server MAY return this code if it does not support a requested function. The body of the response should include some indication about the feature that is not implemented.
9. Authentication

9.1 IP-device Authentication

The Location Server identifies that a request is comes from an IP-device by the incoming URI and the accompanying HELD parameters. For these requests, the source IP address of the request message is used to identify the IP-device. The location determined is for the device that is using the IP address. Location information is also returned to this device. Therefore there is no explicit requirement for an IP-device to authenticate itself with in the access network, over and above what is needed to gain access to the network in the first place.

9.2 OBO Authentication

OBOs require an explicit trust relationship with the Location Server. How this trust relationship is managed with in a specific installation is implementation dependent, but may be implemented with white lists or other such mechanisms.

9.3 Location Client Authentication

Authentication of location-clients requesting the location of an IP-device through a location URI MAY be achieved either through the use of client-side certificates attached to the TLS session, or through the use of Security Assertion Markup Language (SAML). When client-side X.509 certificates are used the corresponding common name SHOULD be lifted from the certificate and used to match against the identity components in the usage ruleset provided by the IP-device. If the location-client request fails authentication then a "404 Not Found" status code SHOULD be returned.
10. Session Management

The protocol described thus far implies in some instances where a session or state is maintained by the Location Server. A Location Server may issue HTTP cookies to provide a persistent session between subsequent requests. When a cookie is used, certain information MUST be maintained by the Location Server. All clients of the Location Server MAY assume that the following information is retained:

- Any PIDF-LO format provided as a "pidf-lo" parameter.
- The authorization policy.
- The location update URI and associated response time.
- Any client identification information provided by the OBO, including IP address.

Only the latest value of any of these data need be maintained.

Session duration MAY be increased by the IP-device re-requesting location information from the Location Server. The Location Server SHOULD maintain IP-device session data until a cookie expires, at which time session data MUST be purged.

An IP-device MAY change its authorization policy at any time. It does this by providing the Location Server with a new "rulesetURI" or "ruleset". Changes to authorization policy are immediate. Where a policy excludes access to location information from a location-client in an existing session, that session MUST be terminated.
11. Security Considerations

All requests for location and subsequent location responses MUST be done using secure HTTP (https).

Exactly how validation and authentication of location-clients using client-side certificates and SAML identity session is implemented is beyond the scope of this document but consideration should be given to employing best current practices.

State information associated with a session at the Location Server MUST be purged when a session expires. Lengthy expiry times SHOULD be avoided.

Mechanisms for protection of location URIs to minimize the likelihood and impact of theft are described in Appendix A.2.2.

The creation of a location update URI by a IP-device SHOULD be done using a randomly selected port.
12. IANA Considerations

IANA has assigned a DHCPv4 option code of TBD for the Location Server URI (LOCSERV_URI) option defined in this document.

IANA has assigned a DHCPv6 option code of TBD for the Location Server URI (OPTION_LOCSERV_URI) option defined in this document.

A new DNS services (SVR) field type of LOCSERV+HTTP with a protocol type of TCP is required in order for a IP-device to discover the Location Server in the access network.
13. Acknowledgements

The authors wish to thank Rohan Mahy and Hannes Tschofenig for their early comments and guidance.

14. Normative references


[I-D.ietf-geopriv-pdif-lo-profile]
Tschofenig, H., "GEOPRIV PIDF-LO Usage Clarification, Considerations and Recommendations",
draft-ietf-geopriv-pdif-lo-profile-00 (work in progress), July 2005.

[I-D.thomson-domain-auth]
Thomson, M. and J. Winterbottom, "Domain Authorization for PIDF-LO",
draft-thomson-domain-auth-01 (work in progress), June 2005.

[I-D.winterbottom-location-uri]
Winterbottom, J., "Rationale for Location by Reference",
draft-winterbottom-location-uri-00 (work in progress), July 2005.

[W3C.REC-xmlschema-2-20041028]
W3C REC REC-xmlschema-2-20041028, October 2004.

Authors’ Addresses

James Winterbottom
Nortel
PO Box U87
University of Wollongong, NSW 2500
AU
Email: winterb@nortel.com

Martin Dawson
Nortel
PO Box U87
University of Wollongong, NSW 2500
AU
Email: mdawson@nortel.com
Martin Thomson
Nortel
PO Box U87
University of Wollongong, NSW 2500
AU

Email: martin.thomson@nortel.com
Appendix A.  Examples

The following subsections represent a few examples of location requests and subsequent responses.

A.1  Simple Location Request

In this example an IP-device makes a request for any location from the Location Server.

The Location Server determines the location and subsequently returns it to the IP-device, which may then communicate its location to other entities.

```
+--------+         +-----------+         +----------+
|   IP   |         | Location  |         | Location |
| Device |         |  Server   |         |  Client  |
+----+---+         +-----+-----+         +----+-----+
|                   |                    |
|   LocReq(ANY)     |                    |
|------------------>|                    |
|                   |                    |
|               Generate                          |
|               Location                          |
|                   |                    |
|     Location      |                    |
|<------------------|                    |
|                   |                    |
|              Conveyance                          |
| ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ >               |
```

A.1.1  Request for Any Location

In this example the IP-device sends a GET to the Location Server URL.

```
GET / HTTP/1.1
Host: lis.example.com
Accept: application/pidf+xml,text/xml;q=0.8,application/xml;q=0.8
Accept-Charset: UTF-8,*
```
A positive response from the Location Server to the IP-device would look similar to:

HTTP/1.x 200 OK
Server: Example LIS
Set-Cookie: httplocationID=dhf2W6hfx53; expires: Wed, 13 Jul 2005 02:54:47 GMT;secure
Date: Wed, 13 Jul 2005 01:54:47 GMT
Expires: Wed, 13 Jul 2005 01:59:47 GMT
Content-Type: application/pidf+xml
Content-Length: 745

```xml
<?xml version="1.0"?>
<pres xmlns="urn:ietf:params:xml:ns:pidf"
   xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
   xmlns:gml="http://www.opengis.net/gml"
   entity="pres:3650n87934c@lis.example.com">
   <tuple id="3b650sf789nd">
      <status>
         <gp:geopriv>
            <gp:location-info>
               <gml:position>
                  <gml:Point srsName="urn:ogc:def:crs:EPSG:6.6:4326">
                     <gml:pos>-34.407 150.88001</gml:pos>
                  </gml:Point>
               </gml:position>
            </gp:location-info>
            <gp:usage-rules/>
         </gp:geopriv>
      </status>
      <timestamp>2005-07-13T01:59:45+00:00</timestamp>
   </tuple>
</pres>
```

### A.1.2 Request for Civic Location

In this example the IP-device specifically wants a civic location, so it posts a "locationType" of "civic" to the Location Server.

POST / HTTP/1.1
Host: lis.example.com
Accept: application/pidf+xml,text/xml;q=0.8,application/xml;q=0.8
Accept-Charset: UTF-8,*
Content-Type: application/x-www-form-urlencoded
Content-Length: 18
locationType=civic

A civic location response from the Location Server follows:

HTTP/1.1 200 OK
Server: Example LIS
Set-Cookie: httplocationID=789pnLw36T489cse;
        expires: Wed, 13 Jul 2005 02:54:47 GMT;secure
Date: Wed, 13 Jul 2005 01:54:47 GMT
Expires: Wed, 14 Jul 2005 01:54:47 GMT
Content-Type: application/pidf+xml
Content-Length: 809

<?xml version="1.0"?>
<pres xmlns="urn:ietf:params:xml:ns:pidf"
     xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
     entity="pres:Rvi46wB4fd345@lis.example.com">
  <tuple id="38mvwfq54">
    <status>
      <gp:geopriv>
        <gp:location-info>
          <cl:civilAddress>
            <cl:country>AU</cl:country>
            <cl:A1>NSW</cl:A1>
            <cl:A3>Wollongong</cl:A3>
            <cl:A4>North Wollongong</cl:A4>
            <cl:A6>Northfields</cl:A6>
            <cl:STS>Avenue</cl:STS>
            <cl:LMK>University of Wollongong</cl:LMK>
            <cl:LOC>Building 3</cl:LOC>
          </cl:civilAddress>
        </gp:location-info>
        <gp:usage-rules>
          <gp:retransmission-allowed>no</gp:retransmission-allowed>
        </gp:usage-rules>
        <gp:method>DHCP</gp:method>
      </gp:geopriv>
    </status>
    <timestamp>2005-07-13T01:54:47Z</timestamp>
  </tuple>
</pres>
A.2 Location URI Request

In these examples the IP-device requests a location URI from the Location Server. Two examples show the way in which an IP-device may specify either a "rulesetURI" or a "ruleset" to control to whom the Location Server divulges their location information. A third example shows how a location-client may request the location of an IP-device using a location URI.

A.2.1 Request locationURI Specifying rulesetURI

In this example the IP-device specifies a "rulesetURI" when requesting a "locationURI".
A.2.2 Requesting locationURI Specifying ruleset

In this example the IP-device provides an explicit "ruleset" in a request for a "locationURI" in the request.
Note that the location request contains a cookie identifying the session created from a previous request.

POST / HTTP/1.1  
Host: lis.example.com  
Accept: application/pidf+xml,text/xml;q=0.8,application/xml;q=0.8  
Accept-Charset: UTF-8,*  
Cookie: httplocationID=F168jiwdjw92jds09  
Content-Type: multipart/form-data; boundary=-29733  
Content-Length: 1866

---29733
Content-Disposition: form-data; name="locationURI"

unbounded
---29733
Content-Disposition: form-data; name="ruleset"; filename="rules.xml"  
Content-Type: application/auth-policy+xml

<?xml version="1.0"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">
    <rule id="rul123">  
        <conditions>  
            <identity>  
                <id>helpme@roadside.assistance.com</id>  
            </identity>  
        </conditions>  
        <actions/>  
        <transformations/>  
    </rule>  
</ruleset>

---29733--

The Location Server accepts the request and responds with a "locationURI".

HTTP/1.1 200 OK  
Server: Example LIS  
Date: Wed, 13 Jul 2005 01:54:47 GMT  
Expires: Wed, 14 Jul 2005 01:54:47 GMT  
Content-Type: text/plain  
Content-Length: 41

https://lis.example.com/362b0e75du908n1
A.2.3 Location-Client Using locationURI

This example shows how a location-client makes a request to a "locationURI", and the subsequent response.

GET /362b0e75du908n1 HTTP/1.1  
Host: lis.example.com  
Accept: application/pidf+xml,text/xml;q=0.8,application/xml;q=0.8  
Accept-Charset: UTF-8,*

The example illustrates a request for arbitrary information. A POST to the "locationURI" with parameters is also allowed where specific location formats are required.

HTTP/1.x 200 OK  
Server: Example LIS  
Date: Wed, 13 Jul 2005 01:54:47 GMT  
Expires: Wed, 13 Jul 2005 01:59:47 GMT  
Content-Type: application/pidf+xml  
Content-Length: 745

<?xml version="1.0"?>  
<pres xmlns="urn:ietf:params:xml:ns:pidf"  
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"  
xmlns:gml="http://www.opengis.net/gml">  
<tuple id="3b650sf789nd">  
<status>  
<gp:geopriv>  
<gp:location-info>  
<gml:position>  
<gml:Point srsName="urn:ogc:def:crs:EPSG:6.6:4326">  
<gml:pos>-34.407 150.88001</gml:pos>  
</gml:Point>  
</gml:position>  
</gp:location-info>  
<gp:usage-rules/>  
</gp:geopriv>  
</status>  
<timestamp>2005-07-13T01:59:45+00:00</timestamp>  
</tuple>  
</presence>
A.3 Location Assertion

In the example the IP-device contains an on-board GPS so that the device can accurately determine where it is. The IP-device asserts its location to the Location Server to obtain a valid PIDF-LO.
POST / HTTP/1.1
Host: lis.example.com
Accept: application/pidf+xml,text/xml;q=0.8,application/xml;q=0.8
Accept-Charset: UTF-8,*
Content-Type: multipart/form-data; boundary=-29733
Content-Length: 1866

---29733
Content-Disposition: form-data; name="locationType"

geodetic
---29733
Content-Disposition: form-data; name="pidf-lo";
   filename="gpsloc.xml"
Content-Type: application/pidf+xml

<?xml version="1.0"?>
<pres xmlns="urn:ietf:params:xml:ns:pidf"
   xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
   xmlns:gml="http://www.opengis.org/gml"
   entity="pres:user@example.com">
   <tuple id="38mvwfq54">
      <status>
         <gp:geopriv>
            <gp:location-info>
               <gml:position>
                  <gml:Point srsName="urn:ogc:def:crs:EPSG:6.6:4326">
                     <gml:pos>-34.407 150.88001</gml:pos>
                  </gml:Point>
               </gml:position>
            </gp:location-info>
            <gp:usage-rules>
               <gp:retransmission-allowed>no</gp:retransmission-allowed>
            </gp:usage-rules>
            <gp:method>GPS</gp:method>
         </gp:geopriv>
      </status>
      <timestamp>2005-07-13T01:54:32Z</timestamp>
   </tuple>
</presence>

---29733--

If the Location Server is able to satisfy the assertion request then a "200 OK" status is returned to the IP-device along with the PIDF-LO providing the location.
HTTP/1.x 200 OK
Server: Example LIS
Set-Cookie: httplocationID=789pnLw36T489cse;
    expires: Wed, 13 Jul 2005 02:54:47 GMT;secure
Date: Wed, 13 Jul 2005 01:54:47 GMT
Expires: Wed, 14 Jul 2005 01:54:47 GMT
Content-Type: application/pidf+xml
Content-Length: 745

<?xml version="1.0"?>
<pres xmlns="urn:ietf:params:xml:ns:pidf"
     xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
     xmlns:gml="http://www.opengis.org/gml"
     entity="pres:wv46sny894G9@lis.example.com">
  <tuple id="38mvwfq54">
    <status>
      <gp:geopriv>
        <gp:location-info>
          <gml:position>
            <gml:Point srsName="urn:ogc:def:crs:EPSG:6.6:4326">
              <gml:pos>-34.407 150.88001</gml:pos>
            </gml:Point>
          </gml:position>
        </gp:location-info>
        <gp:usage-rules>
          <gp:retransmission-allowed>no</gp:retransmission-allowed>
        </gp:usage-rules>
        <gp:method>GPS</gp:method>
        <gp:provided-by>
          <mydp:myProvidedBy
            xmlns:mydp="http://www.example.org/schema/myDataProvider">
            <dp:DataProviderID>99999</dp:DataProviderID>
            <dp:TelURI>tel:555-99</dp:TelURI>
            <dp:URL>https://www.example.com</dp:URL>
          </mydp:myProvidedBy>
        </gp:provided-by>
      </gp:geopriv>
    </status>
    <timestamp>2005-07-13T01:54:32Z</timestamp>
  </tuple>
</presence>

Note that the "provided-by" element has also been populated.
If the location assertion had failed in the above example then the Location Server would return the location that the Location Generator generated for the IP-device. If the "exact" parameter was set to "true" then the request would fail.

A.4 Requests Using Update and Location URIs

In this example the IP-device requests a location URI from the Location Server that includes a "rulesetURI" and an "updateURI". The Location Server generates a location URI and returns it to the IP-device.

```
+--------+               +-----------+      +----------+
|   IP   |               | Location  |      | Location |
| Device |               |  Server   |      |  Client  |
+----+---+               +-----+-----+      +----+-----+
|                         |                 |
| LocReq(URI,upURI,rules) |                 |
|------------------------>|                 |
|                         |                 |
| Generate               |                 |
| LocationURI            |                 |
|<------------------------|                 |
| Conveyance(locationURI) |                 |
| ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~> |
| LocReq(ANY)            |                 |
|<------------------------|                 |
| Authenticate           |                 |
| LocReq                 |                 |
|<------------------------|                 |
| Generate               |                 |
| Location               |                 |
|------------------------|                 |
| Assert                 |                 |
| Location               |                 |
|<------------------------|                 |

The IP-device sends the location URI to the location-client. The location-client uses the location URI to request a location from the Location Server.
The Location Server uses the update URI to request location information from the IP-device. The IP-device determines its location and responds to the Location Server.

The Location Server asserts that the provided location information is valid and then returns a PIDF-LO to the location-client.

POST / HTTP/1.1
Host: lis.example.com
Accept: text/plain
Accept-Charset: UTF-8,*
Content-Type: application/x-www-form-urlencoded
Content-Length: 17

locationURI=unbounded&updateURI=https://1.2.3.4:9974/locationUpdate

The Location-Server response and subsequent location-client request are the same as shown in Appendix A.2.2 and are not repeated here. The Location Server request to the IP-device using the update URI is shown below. The response from the IP-device will be a PIDF-LO or an error.

GET /locationUpdate HTTP/1.1
Host: 1.2.3.4
Accept: application/pidf+xml,text/xml;q=0.8,application/xml;q=0.8
Accept-Charset: UTF-8,*

A.5 OBO Location Request Examples

The example shows an OBO requesting location on-behalf-of an IP-device. While not directly shown, the OBO is expected to be in a communication path between the IP-device and the location-client.
The OBO requests a location URI from the Location Server. It uses the "ip" and "hwaddr" parameters in the request to identify the IP-device.

POST / HTTP/1.1  
Host: lis.example.com  
Accept: text/plain  
Accept-Charset: UTF-8,*  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 46

    ip=1.2.3.4&hwaddr=abcdef012345&locationURI=true

The Location Server generates a location URI and returns this to the OBO. The response is not shown here but similar to examples in
Appendix A.2.2.

The OBO includes the location URI in the IP-device’s communications to the location-client, which is not shown here but similar to examples in Appendix A.2.2.

The location-client uses the location URI to request location from the Location Server, which is not shown here but similar to examples in Appendix A.2.2.

If the location-client satisfies the default ruleset imposed by the Location Server, then a location is returned, which is not shown here but similar to examples in Appendix A.2.2.
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

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

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

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