GEOPRIV Working Group
INTERNET-DRAFT
Obsoletes: 3825 (if approved)
Category: Standards Track
Expires: May 7, 2011
7 November 2010

J. Polk
Cisco Systems
J. Schnizlein
Individual Contributor
M. Linsner
Cisco Systems
M. Thomson
Andrew
B. Aboba (ed)
Microsoft Corporation

Dynamic Host Configuration Protocol Options for Coordinate-based Location Configuration Information

draft-ietf-geopriv-rfc3825bis-13.txt

Abstract

This document specifies Dynamic Host Configuration Protocol Options (both DHCPv4 and DHCPv6) for the coordinate-based geographic location of the client. The Location Configuration Information (LCI) includes Latitude, Longitude, and Altitude, with resolution or uncertainty indicators for each. Separate parameters indicate the reference datum for each of these values.

Status of This Memo

Table of Contents

1. Introduction

The physical location of a network device has a range of applications. In particular, emergency telephony applications rely on knowing the location of a caller in order to determine the correct emergency center.

The location of a device can be represented either in terms of geospatial (or geodetic) coordinates, or as a civic address. Different applications may be more suited to one form of location information; therefore, both the geodetic and civic forms may be used simultaneously.

This document specifies Dynamic Host Configuration Protocol v4 (DHCPv4) [RFC2131] and DHCPv6 [RFC3315] options for the coordinate-based geographic location of the client, to be provided by the server. "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information" [RFC4776] specifies DHCP options for civic addresses.

The geodetic coordinate options defined in this document and the civic address options defined in RFC 4776 [RFC4776] enable a DHCP client to obtain its location. For example, a wired Ethernet host might use these options for location determination. In this case, the location information could be derived from a wiremap by the DHCP server, using the Circuit-ID Relay Agent Information Option (RAIO) defined (as Sub-Option 1) in RFC 3046 [RFC3046]. The DHCP server could correlate the Circuit-ID with the geographic location where the identified circuit terminates (such as the location of the wall

jack).

The mechanism defined here may also be utilized to provide location to wireless hosts. DHCP relay agent sub-options (RAIO) [RFC3046] is one method a DHCP server might use to perform host location determination. Currently, the relay agent sub-options do not include data sets required for device level location determination of wireless hosts. In cases where the DHC server uses RAIO for location determination, a wireless host can use this mechanism to discover location of the radio access point, or the area of coverage for the radio access point.

An important feature of this specification is that after the relevant DHCP exchanges have taken place, the location information is stored on the end device rather than somewhere else, where retrieving it might be difficult in practice.

1.1. Conventions used in this document

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].

1.2. Resolution and Uncertainty

The DHCP options defined in this document include fields quantifying the resolution or uncertainty associated with a target location. No inferences relating to privacy policies can be drawn from either uncertainty or resolution values.

As utilized in this document, resolution refers to the accuracy of a reported location, as expressed by the number of valid bits in each of the Latitude, Longitude and Altitude fields.

In the context of location technology, uncertainty is a quantification of errors. Any method for determining location is subject to some sources of error; uncertainty describes the amount of error that is present. Uncertainty might be the coverage area of a wireless transmitter, the extent of a building or a single room.

Uncertainty is usually represented as an area within which the target is located. In this document, each of the three axes can be assigned an uncertainty value. In effect, this describes a rectangular prism, which may be used as a coarse representation of a more complex shape that fits within it. See Section 2.3.2 for more detail on the correspondence between shapes and uncertainty.

When representing locations from sources that can quantify uncertainty, the goal is to find the smallest possible rectangular prism that this format can describe. This is achieved by taking the minimum and maximum values on each axis and ensuring that the final encoding covers these points. This increases the region of uncertainty, but ensures that the region that is described encompasses the target location.

The DHCPv4 option format defined in this document supports both resolution and uncertainty parameters. Version 0 of the DHCPv4 option format includes a resolution parameter for each of the dimensions of location. Since this resolution parameter need not apply to all dimensions equally, a resolution value is included for each of the three location elements. Since version 0 of the DHCPv6 option format is not defined, the DHCPv6 option does not support a resolution parameter. Version 1 of the DHCPv4 and DHCPv6 option format utilizes an uncertainty parameter. Appendix A describes the mapping of DHCP option values to GML. Appendix B of this document

provides examples showing the calculation of resolution values. Appendix C provides an example demonstrating calculation of uncertainty values.

Since the PIDF-LO format [RFC4119][RFC5491] is used to conveying location and the associated uncertainty within an emergency call [Convey], a mechanism is needed to convert the information contained within the DHCPv4 and DHCPv6 options to PIDF-LO. This document describes the following conversions:

version 0 to PIDF-LO
version 1 to PIDF-LO
PIDF-LO to version 1

Conversion to PIDF-LO does not increase uncertainty; conversion from

PIDF-LO to version 1 increases uncertainty by less than a factor of 2 in each dimension. Since it is not possible to translate an arbitrary PIDF-LO to version 0 with a bounded increase in uncertainty, the conversion to version 0 is not specified.

2. DHCP Option Format

This section defines the format for the DHCPv4 and DHCPv6 options. These options utilize a similar format, differing primarily in the option code.

2.1. DHCPv6 Option

The DHCPv6 [RFC3315] option format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Option Code (TBD)       |          OptLen (16)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  LatUnc   |                   Latitude                        +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Lat (cont'd)  |  LongUnc  |              Longitude            +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Longitude (cont'd)         | AType |   AltUnc  |  Altitude +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Altitude (cont'd)                |Ver| Res |Datum|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code: GEOCONF_GEODETIC (16 bits).

OptLen: Option Length (16). This option has a fixed length, so that the value of this octet will always be 16.

LatUnc: 6 bits. When the Ver field = 1, this field represents latitude uncertainty. The contents of this field is undefined for other values of the Ver field.

Latitude: a 34 bit fixed point value consisting of 9 bits of integer and 25 bits of fraction, interpreted as described in Section 2.3.

LongUnc: 6 bits. When the Ver field = 1, this field represents longitude uncertainty. The contents of this field is undefined for other values of the Ver field.

Longitude: a 34 bit fixed point value consisting of 9 bits of integer and 25 bits of fraction, interpreted as described in Section 2.3.

AType: Altitude Type (4 bits).

AltUnc: 6 bits. When the Ver field = 1, this field represents altitude uncertainty. The contents of this field is undefined for other values of the Ver field.

Altitude: A 30 bit value defined by the AType field.

Ver: The Ver field is two bits, providing for four potential versions. This specification defines the behavior of version 1. The Ver field is always located at the same offset from the beginning of the option, regardless of the version in use.

Res: The Res field which is 3 bits, is reserved. These bits have been used by [IEEE-802.11y], but are not defined within this specification.

Datum: 3 bits. The Map Datum used for the coordinates given in this Option.

2.2. DHCPv4 Option

The DHCPv4 option format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Code 123    |    Length     |   LatUnc  |     Latitude      +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Latitude (cont'd)               | LongUnc  |   +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Longitude                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AType |   AltUnc  |                Altitude                   +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Alt.(cont'd)  |Ver| Res |Datum|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code: 8 bits. The code for the DHCPv4 option (123).

Length: 8 bits. The length of the DHCPv4 option, in octets. For versions 0 and 1, the option length is 16.

LatUnc: 6 bits. When the Ver field = 0, this field represents latitude resolution. When the Ver field = 1, this field represents latitude uncertainty.

Latitude: a 34 bit fixed point value consisting of 9 bits of signed integer and 25 bits of fraction, interpreted as described in Section 2.3.

LongUnc: 6 bits. When the Ver field = 0, this field represents longitude resolution. When the Ver field = 1, this field represents longitude uncertainty.

Longitude: a 34 bit fixed point value consisting of 9 bits of signed integer and 25 bits of fraction, interpreted as described in Section 2.3.

AType: Altitude Type (4 bits).

AltUnc: 6 bits. When the Ver field = 0, this field represents altitude resolution. When the Ver field = 1, this field represents altitude uncertainty.

Altitude: A 30 bit value defined by the AType field.

Ver: The Ver field is two bits, providing for four potential

versions. This specification defines the behavior of version 0 (originally specified in [RFC3825]) as well as version 1. The Ver field is always located at the same offset from the beginning of the option, regardless of the version in use.

Res: The Res field which is 3 bits, is reserved. These bits have been used by [IEEE-802.11y], but are not defined within this specification.

Datum: 3 bits. The Map Datum used for the coordinates given in this Option.

2.2.1. Version Support

2.2.1.1. Client Version Support

DHCPv6 clients implementing this specification MUST support receiving version 1 responses. DHCPv4 clients implementing this specification MUST support receiving responses of versions 0 and 1. It is required that DHCPv4 client implementations support version 1 so the versioning capability added by this document does not cause errors interpreting the Latitude, Longitude and Altitude values. Since this specification utilizes the same DHCPv4 option code as [RFC3825], the option format does not provide a means for the DHCPv4 client to indicate the highest version that it supports to the DHCPv4 server.

Moving forward, DHCPv4 and DHCPv6 clients not understanding a datum value MUST assume a World Geodesic System 1984 (WGS84) [WGS84] datum (EPSG [EPSG] 4326 or 4979, depending on whether there is an Altitude value present) and proceed accordingly. Assuming that a less accurate location value is better than none, this ensures that some (perhaps less accurate) location is available to the client.

2.2.1.2. Server Version Selection

DHCPv6 servers implementing this specification MUST support sending version 1 responses. A DHCPv4 server implementing this specification MUST support sending version 1 responses and SHOULD support sending version 0 responses. A DHCPv4 server that provides location information cannot provide options with both version 0 and version 1 formats in the same response. This is not useful since receiving two copies of the same Option (either in the same response or a separate response) causes a DHCPv4 client to replace the information in the old Option with the information in the new Option.

A DHCPv4 server uses configuration to determine which version to send in a response. For example, where a mixture of version 0 and version

1 clients are expected, the DHCPv4 server could be configured to send version 0 or version 1 depending on configuration (possibly making the choice based on information such as the client MAC address). Where few version 0 clients are expected, the DHCPv4 server could be configured to send only version 1 responses. Version 0 options will provide resolution, while version 1 options will provide an area of uncertainty.

An RFC 3825 DHCPv4 client that receives a version 1 option, as defined in this document, will either reject the Option or will not understand the additions to the Datum field and will misinterpret the LongUnc, LatUnc, and AltUnc values. If the RFC 3825 DHCPv4 client does not reject the option and utilizes the location data it will most likely assume a datum. Assuming one of the RFC 3825 datums causes correct interpretation of Latitude/Longitude/Altitude values. The values for LongUnc/LatUnc/AltUnc are mistakenly interpreted as representing significant digits. The resultant location value will be in error up to a full degree of latitude and longitude, and a full increment of altitude.

This results in a version 0-only DHCPv4 client either not obtaining location information (with no ability to indicate to the server that version 1 was unsupported), or misinterpreting the option. Therefore, in situations where some DHCPv4 clients are known to support only version 0, by default the DHCPv4 server SHOULD send a version 0 response.

2.3. Latitude and Longitude Fields

The Latitude and Longitude values in this specification are encoded as 34 bit, twos complement, fixed point values with 9 integer bits and 25 fractional bits. The exact meaning of these values is determined by the datum; the description in this section applies to the datums defined in this document. This document uses the same definition for all datums it specifies.

Latitude values encoded by the DHCP server MUST be constrained to the range from -90 to +90 degrees. Location consumers MUST be prepared to normalize values outside this range. Values outside the range are

normalized by clamping (e.g. values less than -90 degrees are set to -90; values greater than 90 degrees are set to +90). Positive latitudes are north of the equator and negative latitudes are south of the equator.

Longitude values encoded by the DHCP server MUST be normalized to the range from -180 to +180 degrees. Location consumers MUST be prepared to normalize values outside this range. Values outside the range are normalized by wrapping (e.g. adding or subtracting 360 until they

fall within the range of -180 to 180). Positive longitudes are east of the Prime Meridian (Greenwich) and negative (2s complement) longitudes are west of the Prime Meridian.

When encoding, Latitude and Longitude values are rounded to the nearest 34-bit binary representation. This imprecision is considered acceptable for the purposes to which this form is intended to be applied and is ignored when decoding.

2.3.1. Latitude and Longitude Resolution

The Latitude (LatUnc), Longitude (LongUnc) and Altitude (AltUnc) Uncertainty fields are encoded as 6 bit, unsigned integer values. In the version 0 DHCPv4 Option, the LatUnc, LongUnc and AltUnc fields are used to encode the number of bits of resolution. The resolution sub-fields accommodate the desire to easily adjust the precision of a reported location. Contents beyond the claimed resolution MAY be randomized to obscure greater precision that might be available.

In the version 0 DHCPv4 Option, the LatUnc value encodes the number of high-order latitude bits that should be considered valid. Any bits entered to the right of this limit should not be considered valid and might be purposely false, or zeroed by the sender. The examples in Appendix B illustrate that a smaller value in the resolution field increases the area within which the device is located. A value of 2 in the LatUnc field indicates a precision of no greater than 1/6th that of the globe (see the first example of Appendix B). A value of 34 in the LatUnc field indicates a precision of about 3.11 mm in latitude at the equator.

In the version 0 DHCPv4 Option, the LongUnc value encodes the number of high-order longitude bits that should be considered valid. Any

bits entered to the right of this limit should not be considered valid and might be purposely false, or zeroed by the sender. A value of 2 in the LongUnc field indicates precision of no greater than 1/6th that of the globe (see the first example of Appendix B). A value of 34 in the LongUnc field indicates a precision of about 2.42 mm in Longitude (at the equator). Because lines of longitude converge at the poles, the distance is smaller (better precision) for locations away from the equator.

2.3.2. Latitude and Longitude Uncertainty

In the DHCPv6 option and the version 1 DHCPv4 option, the Latitude and Longitude Uncertainty fields (LatUnc and LongUnc) quantify the amount of uncertainty in each of the Latitude and Longitude values respectively. A value of 0 is reserved to indicate that the uncertainty is unknown; values greater than 34 are reserved.

A point within the region of uncertainty is selected to be the encoded point; the centroid of the region is often an appropriate choice. The value for uncertainty is taken as the distance from the selected point to the furthest extreme of the region of uncertainty on that axis. This is demonstrated in the figure below, which shows a two-dimensional polygon that is projected on each axis. In the figure, "X" marks the point that is selected; the ranges marked with "U" is the uncertainty.

```
        ___          ___________
        ^ |         /           |
        | |        /            |
        | |       /             |
        U |      /              |
        | |     (               |
        V |     |               |
        --X     |         X     |
          |     |               `---------.
          |     |                         |
          |     |                         |
          |     |                         |
          _     `-------------------------'

                |---------X---------------|
                          |<------U------>|
```

Key

V, ^ = vertical arrows, delimiting the vertical uncertainty range.
<>   = horizontal arrows, delimiting the horizontal uncertainty range.

Uncertainty applies to each axis independently.

The amount of uncertainty can be determined from the encoding by taking 2 to the power of 8, less the encoded value. As is shown in the following formula, where "x" is the encoded integer value:

    uncertainty = 2 ^ (8 - x)

The result of this formula is expressed in degrees of latitude or longitude. The uncertainty is added to the base latitude or longitude value to determine the maximum value in the uncertainty range; similarly, the uncertainty is subtracted from the base value to determine the minimum value. Note that because lines of longitude converge at the poles, the actual distance represented by this uncertainty changes with the distance from the equator.

If the maximum or minimum latitude values derived from applying uncertainty are outside the range of -90 to +90, these values are trimmed to within this range. If the maximum or minimum longitude values derived from applying uncertainty are outside the range of -180 to +180, then these values are normalized to this range by adding or subtracting 360 as necessary.

The encoded value is determined by subtracting the next highest whole integer value for the base 2 logarithm of uncertainty from 8. As is shown by the following formula, where uncertainty is the midpoint of the known range less the lower bound of that range:

    x = 8 - ceil(log2(uncertainty))

Note that the result of encoding this value increases the range of uncertainty to the next available power of two; subsequent repeated encodings and decodings do not change the value. Only increasing uncertainty means that the associated confidence does not have to

decrease.

2.4. Altitude

The Altitude value is expressed as a 30 bit, fixed point, twos complement integer with 22 integer bits and 8 fractional bits. How the Altitude value is interpreted depends on the Altitude Type (AType) value and the selected datum. Three Altitude Type values are defined in this document: unknown (0), meters (1) and floors (2).

2.4.1. No Known Altitude (AType = 0)

In some cases, the altitude of the location might not be provided. An Altitude Type value of zero indicates that the altitude is not given to the client. In this case, the Altitude and Altitude Uncertainty fields can contain any value and MUST be ignored.

2.4.2. Altitude in Meters (AType = 1)

If the Altitude Type has a value of one, Altitude is measured in meters, in relation to the zero set by the vertical datum.

2.4.3. Altitude in Floors (AType = 2)

A value of two for Altitude Type indicates that the Altitude value is measured in floors. This value is relevant only in relation to a building; the value is relative to the ground level of the building. In this definition, numbering starts at ground level, which is floor 0 regardless of local convention.

Non-integer values can be used to represent intermediate or sub-floors, such as mezzanine levels. For instance, a mezzanine between floors 4 and 5 could be represented as 4.1.

2.4.4. Altitude Resolution

In the version 0 DHCPv4 Option, the Altitude Uncertainty (AltUnc) value encodes the number of high-order altitude bits that should be considered valid. Values above 30 (decimal) are undefined and reserved.

If the Altitutde Type value is one (AType = 1), an AltUnc value 0.0 would indicate unknown Altitude. The most precise altitude would have an AltUnc value of 30. Many values of AltUnc would obscure any variation due to vertical datum differences.

The AltUnc field SHOULD be set to maximum precision when AType = 2 (floors) when a floor value is included in the DHCP Reply, or when AType = 0, to denote that the floor isn't known. An altitude coded as AType = 2, AltRes = 30, and Altitude = 0.0 is meaningful even outside a building, and represents ground level at the given latitude and longitude.

2.4.5. Altitude Uncertainty

In the DHCPv6 option or the version 1 DHCPv4 option, the AltUnc value quantifies the amount of uncertainty in the Altitude value. As with LatUnc and LongUnc, a value of 0 for AltUnc is reserved to indicate that Altitude Uncertainty is not known; values above 30 are also reserved. Altitude Uncertainty only applies to Altitude Type 1.

The amount of Altitude Uncertainty can be determined by the following formula, where x is the encoded integer value:

```
Uncertainty = 2 ^ ( 21 - x )
```

This value uses the same units as the associated altitude.

Similarly, a value for the encoded integer value can be derived by the following formula:

```
x = 21 - ceil( log2( uncertainty ) )
```

2.5. Datum

The Datum field determines how coordinates are organized and related to the real world. Three datums are defined in this document, based on the definitions in [OGP.Geodesy]:

1: WGS84 (Latitude, Longitude, Altitude):
   The World Geodesic System 1984 [WGS84] coordinate reference system.

This datum is identified by the European Petroleum Survey Group (EPSG)/International Association of Oil & Gas Producers (OGP) with the code 4979, or by the URN "urn:ogc:def:crs:EPSG::4979". Without Altitude, this datum is identified by the EPSG/OGP code 4326 and the URN "urn:ogc:def:crs:EPSG::4326".

2: NAD83 (Latitude, Longitude) + NAVD88:
This datum uses a combination of the North American Datum 1983 (NAD83) for horizontal (Latitude and Longitude) values, plus the North American Vertical Datum of 1988 (NAVD88) vertical datum.

This datum is used for referencing location on land (not near tidal water) within North America.

NAD83 is identified by the EPSG/OGP code of 4269, or the URN "urn:ogc:def:crs:EPSG::4269". NAVD88 is identified by the EPSG/OGP code of 5703, or the URN "urn:ogc:def:crs:EPSG::5703".

3: NAD83 (Latitude, Longitude) + MLLW:
This datum uses a combination of the North American Datum 1983 (NAD83) for horizontal (Latitude and Longitude) values, plus the Mean Lower Low Water (MLLW) vertical datum.

This datum is used for referencing location on or near tidal water within North America.

NAD83 is identified by the EPSG/OGP code of 4269, or the URN "urn:ogc:def:crs:EPSG::4269". MLLW does not have a specific code or URN.

All hosts MUST support the WGS84 datum (Datum 1).

3. Security Considerations

Geopriv requirements (including security requirements) are discussed in "Geopriv Requirements" [RFC3693]. A threat analysis is provided in "Threat Analysis of the Geopriv Protocol" [RFC3694].

Since there is no privacy protection for DHCP messages, an eavesdropper who can monitor the link between the DHCP server and requesting client can discover this LCI.

To minimize the unintended exposure of location information, the LCI option SHOULD be returned by DHCP servers only when the DHCP client

has included this option in its 'parameter request list' (section 3.5 [RFC2131]).

Where critical decisions might be based on the value of this option, DHCP authentication as defined in "Authentication for DHCP Messages" [RFC3118] and "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)" [RFC3315] SHOULD be used to protect the integrity of the DHCP options.

Link layer confidentiality and integrity protection may also be employed to reduce the risk of location disclosure and tampering.

4. IANA Considerations

IANA has assigned a DHCPv4 option code of 123 for the GeoConf option defined in this document. Assignment of a DHCPv6 option code is requested.

The GeoConf Option defines two fields for which IANA maintains a registry: The Altitude Type (AType) field and the Datum field (see Section 2). The datum indicator MUST include specification of both horizontal and vertical datum. New values for the Altitude Type (AType) and Datum fields are assigned through "Standards Action" [RFC5226]. New Altitude Types MUST define the way that the 30 bit altitude values and the associated 6 bit uncertainty are interpreted. New datums MUST define the way that the 34 bit values and the respective 6 bit uncertainties are interpreted. The initial values of the Altitude registry are as follows:

AType = 0 No known altitude.

AType = 1 meters of altitude defined by the vertical datum specified.

AType = 2 building floors of altitude.

Datum = 1 denotes the vertical datum WGS 84 as defined by the EPSG as their CRS Code 4327; CRS Code 4327 also specifies WGS 84 as the vertical datum.

Datum = 2 denotes the vertical datum NAD83 as defined by the EPSG as their CRS Code 4269; North American Vertical Datum of 1988 (NAVD88) is the associated vertical datum for NAD83.

Datum = 3 denotes the vertical datum NAD83 as defined by the EPSG as their CRS Code 4269; Mean Lower Low Water (MLLW) is the associated vertical datum for NAD83.

This document defines the Ver field for the DHCPv4 and DHCPv6 options. New values for the Ver field are assigned through "Standards Action" [RFC5226]. Initial values are as follows:

0: DHCPv4 Implementations conforming to [RFC3825]
1: Implementations of this specification (for both DHCPv4 and DHCPv6)

5. Acknowledgments

The authors would like to thank Randall Gellens, Patrik Falstrom, Ralph Droms, Ted Hardie, Jon Peterson, Robert Sparks and Nadine Abbott for their inputs and constructive comments regarding this document. Additionally, the authors would like to thank Greg Troxel for the education on vertical datums, as well as Carl Reed. Thanks to Richard Barnes for his contribution on GML mapping for resolution.

6. References

6.1. Normative References

[EPSG] European Petroleum Survey Group, http://www.epsg.org/ and http://www.epsg-registry.org/

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C. and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[WGS84] US National Imagery and Mapping Agency, "Department of

Defense (DoD) World Geodetic System 1984 (WGS 84), Third Edition", NIMA TR8350.2, January 2000, https://www1.nga.mil/PRODUCTSSERVICES/GEODESYGEOPHYSICS/WORLDGEODETICSYSTEM/Pages/default.aspx and http://www.ngs.noaa.gov/faq.shtml#WGS84

6.2. Informational References

[Convey] Polk, J., Rosen, B. and J. Peterson, "Location Conveyance for the Session Initiation Protocol", Internet draft (work in progress), draft-ietf-sipcore-location-conveyance-04.txt, October 25, 2010.

[GeoShape] Thomson, M. and C. Reed, "GML 3.1.1 PIDF-LO Shape Application Schema for use by the Internet Engineering Task Force (IETF)", Candidate OpenGIS Implementation Specification 06-142, Version: 0.0.9, December 2006.

[IEEE-802.11y] Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 3: 3650-3700 MHz Operation in USA, November 2008.

[NENA] National Emergency Number Association (NENA) www.nena.org NENA Technical Information Document on Model Legislation Enhanced 911 for Multi-Line Telephone Systems.

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

[RFC3693] Cuellar, J., Morris, J., Mulligan, D., Peterson, J. and J. Polk, "Geopriv Requirements", RFC 3693, February 2004.

[RFC3694] Danley, M., Mulligan, D., Morris, J. and J. Peterson, "Threat Analysis of the Geopriv Protocol", RFC 3694, February 2004.

[RFC3825] Polk, J., Schnizlein, J. and M. Linsner, "Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information", RFC 3825, July 2004.

[RFC4119] Peterson, J., "A Presence-based GEOPRIV Location Object Format", RFC 4119, December 2005.

[RFC4776] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information", RFC 4776, November 2006.

[RFC5139] Thomson, M. and J. Winterbottom, "Revised Civic Location Format for Presence Information Data Format Location Object (PIDF-LO)", RFC 5139, February 2008.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, May 2008.

[RFC5491] Winterbottom, J., Thomson, M. and H. Tschofenig, "GEOPRIV PIDF-LO Usage Clarification, Considerations, and Recommendations ", RFC 5491, March 2009

Appendix A. GML Mapping

The GML representation of a decoded DHCP option depends on what fields are specified. The DHCP format for location logically describes a geodetic prism, rectangle, or point, depending on whether Altitude and uncertainty values are provided. In the absence of uncertainty information, the value decoded from the DHCP form can be expressed as a single point; this is true regardless of whether the version 0 or version 1 interpretations of the uncertainty fields are used. If the point includes Altitude, it uses a three dimensional CRS, otherwise it uses a two dimensional CRS. If all fields are included along with uncertainty, the shape described is a rectangular prism. Note that this is necessary given that uncertainty for each axis is provided independently.

If Altitude or Altitude Uncertainty (AltUnc) is not specified, the shape is described as a rectangle using the "gml:Polygon" shape. If Altitude is available, a three dimensional CRS is used, otherwise a two dimensional CRS is used.

For Datum values of 2 or 3 (NAD83), there is no available CRS URN that covers three dimensional coordinates. By necessity, locations described in these datums can be represented by two dimensional shapes only; that is, either a two dimensional point or a polygon.

If the Altitude Type is 2 (floors), then this value can be represented using a civic address object [RFC5139] that is presented alongside the geodetic object.

This Appendix describes how the location value encoded in DHCP format for geodetic location can be expressed in GML. The mapping is valid for the DHCPv6 option as well as versions 0 and 1 of the DHCPv4 option, and for the currently-defined datum values (1, 2, and 3). Further version or datum definitions should provide similar mappings.

These shapes can be mapped to GML by first computing the bounds that are described using the coordinate and uncertainty fields, then encoding the result in a GML Polygon or Prism shape.

A.1. GML Templates

If Altitude is provided in meters (Altitude Type 1) and the datum value is WGS84 (value 1), then the proper GML shape is a Prism, with the following form (where $value$ indicates a value computed from the DHCP option as described below):

```
<gs:Prism srsName="urn:ogc:def:crs:EPSG::4979"
          xmlns:gs="http://www.opengis.net/pidflo/1.0"
```



```
          xmlns:gml="http://www.opengis.net/gml">
  <gs:base>
    <gml:Polygon>
      <gml:exterior>
        <gml:LinearRing>
          <gml:posList>
            $lowLatitude$ $lowLongitude$ $lowAltitude$
            $lowLatitude$ $highLongitude$ $lowAltitude$
            $highLatitude$ $highLongitude$ $lowAltitude$
            $highLatitude$ $lowLongitude$ $lowAltitude$
            $lowLatitude$ $lowLongitude$ $lowAltitude$
          </gml:posList>
```

```
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </gs:base>
    <gs:height uom="urn:ogc:def:uom:EPSG::9001">
      $highAltitude - lowAltitude$
    </gs:height>
  </gs:Prism>
```

The Polygon shape is used if Altitude is omitted or specified in floors, or if either NAD83 datum is used (value 2 or 3). The corresponding GML Polygon has the following form:

```
  <gml:Polygon srsName="$2D-CRS-URN$"
               xmlns:gml="http://www.opengis.net/gml">>
    <gml:exterior>
      <gml:LinearRing>
        <gml:posList>
          $lowLatitude$ $lowLongitude$
          $lowLatitude$ $highLongitude$
          $highLatitude$ $highLongitude$
          $highLatitude$ $lowLongitude$
          $lowLatitude$ $lowLongitude$
        </gml:posList>
      </gml:LinearRing>
    </gml:exterior>
  </gml:Polygon>
```

The value "2D-CRS-URN" is defined by the datum value: If the datum is WGS84 (value 1), then the 2D-CRS-URN is "urn:ogc:def:crs:EPSG::4326". If the datum is NAD83 (value 2 or 3), then the 2D-CRS-URN is "urn:ogc:def:crs:EPSG::4269".

A Polygon shape with the WGS84 three-dimensional CRS is used if the datum is WGS84 (value 1) and the Altitude is specified in meters (Altitude type 1), but no Altitude uncertainty is specified (that is,

AltUnc is 0). In this case, the value of the Altitude field is added after each of the points above, and the srsName attribute is set to the three-dimensional WGS84 CRS, namely "urn:ogc:def:crs:EPSG::4979".

A simple point shape is used if either Latitude uncertainty (LatUnc)

or Longitude uncertainty (LongUnc) is not specified. With Altitude, this uses a three-dimensional CRS; otherwise, it uses a two-dimensional CRS.

```
<gml:Point srsName="$CRS-URN$"
           xmlns:gml="http://www.opengis.net/gml">
  <gml:pos>$Latitude$ $Longitude$ $[Altitude]$</gml:pos>
</gml:Point>
```

A.1.1. Finding Low and High Values using Uncertainty Fields

The uncertainty fields (LatUnc, LongUnc, AltUnc) indicate the bounds of the location region described by a DHCP location object. For version 0 of the DHCPv4 option, the uncertainty fields represent resolution, indicating how many bits of a value contain information. Any bits beyond those indicated can be either zero or one. For the DHCPv6 option and version 1 of the DHCPv4 option, the LatUnc, LongUnc and AltUnc fields indicate uncertainty distances.

The two sections below describe how to compute the Latitude, Longitude, and Altitude bounds (e.g., $lowLatitude$, $highAltitude$) in the templates above. The first section describes how these bounds are computed in the "resolution encoding" (version 0), while the second section addresses the "uncertainty encoding" (version 1).

A.1.1.1. Resolution Encoding

Given a number of resolution bits (i.e., the value of a resolution field), if all bits beyond those bits are set to zero, this gives the lowest possible value. The highest possible value can be found setting all bits to one.

If the encoded value of Latitude/Longitude and resolution (LatUnc, LongUnc) are treated as 34-bit unsigned integers, the following can be used (where ">>" is a bitwise right shift, "&" is a bitwise AND, "~" is a bitwise negation, and "|" is a bitwise OR).

```
mask = 0x3ffffffff >> resolution
lowvalue = value & ~mask
highvalue = value | mask + 1
```

Once these values are determined, the corresponding floating point numbers can be computed by dividing the values by 2^25 (since there

are 25 bits of fraction in the fixed-point representation).

Alternatively, the lowest possible value can be found by using resolution to determine the size of the range. This method has the advantage that it operates on the decoded floating point values. It is equivalent to the first mechanism, to a possible error of 2^-25 (2^-8 for altitude).

```
scale = 2 ^ ( 9 - resolution )
lowvalue = floor( value / scale ) * scale
highvalue = lowvalue + scale
```

Altitude resolution (AltUnc for v0) uses the same process with different constants. There are 22 whole bits in the Altitude encoding (instead of 9) and 30 bits in total (instead of 34).

A.1.1.2. Uncertainty Encoding

In the uncertainty encoding, the uncertainty fields (LongUnc/LatUnc in version 1) directly represent the logarithms of uncertainty distances. So the low and high bounds are computed by first computing the uncertainty distances, then adding and subtracting these from the value provided. If "uncertainty" is the unsigned integer value of the uncertainty field and "value" is the value of the coordinate field:

```
distance = 2 ^ (8 - uncertainty)
lowvalue = value - distance
highvalue = value + distance
```

Altitude uncertainty (AltUnc in version 1) uses the same process with different constants:

```
distance = 2 ^ (21 - uncertainty)
lowvalue = value - distance
```

Appendix B. Calculations of Resolution

The following examples for two different locations demonstrate how the Resolution values for Latitude, Longitude, and Altitude (used in the version 0 DHCPv4 option) can be calculated. In both examples, the geo-location values were derived from maps using the WGS84 map datum, therefore in these examples, the Datum field would have a value = 1 (00000001, or 0x01).

B.1. Location Configuration Information of "White House" (Example 1)

The grounds of the White House in Washington D.C. (1600 Pennsylvania

Ave. NW, Washington, DC 20006) can be found between 38.895375 and 38.898653 degrees North and 77.037911 and 77.035116 degrees West. In this example, we assume that we are standing on the sidewalk on the north side of the White House, between driveways. Since we are not inside a structure, we assume an Altitude value of 15 meters, interpolated from the US Geological survey map, Washington Washington West quadrangle.

The address was NOT picked for any political reason and can easily be found on the Internet or mapping software, but was picked as an easily identifiable location on our planet.

In this example, the requirement of emergency responders in North America via their NENA Model Legislation [NENA] could be met by a LatUnc value of 21 and a LongUnc value of 20. This would yield a geo-location that is Latitude 38.8984375 north to Latitude 38.8988616 north and Longitude -77.0371094 to Longitude -77.0375977. This is an area of approximately 89 feet by 75 feet or 6669 square feet, which is very close to the 7000 square feet requested by NENA. In this example, a service provider could enforce that a device send a Location Configuration Information with this minimum amount of resolution for this particular location when calling emergency services.

An approximate representation of this location might be provided using the version 0 encoding as follows:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Code (123)  |  OptLen (16)  |  LatUnc   |     Latitude      .
 |0 1 1 1 1 0 1 1|0 0 0 1 0 0 0 0|0 1 0 0 1 0|0 0 0 1 0 0 1 1 0 1.
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .               Latitude (cont'd)             |  LongUnc  |    .
 .1 1 0 0 1 0 1 1 1 0 0 1 1 0 0 0 0 1 1 0 0 0 1 1|0 1 0 0 0 1|1 1.
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .                    Longitude (cont'd)                         |
 .0 1 1 0 0 1 0 1 1 1 1 0 1 1 0 1 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | AType |   AltUnc  |                Altitude                   .
 |0 0 0 1|0 1 0 0 0 1|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1.
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
 .  Alt (cont'd) |Ver| Res |Datum|
 .0 0 0 0 0 0 0 0|0 0|0 0 0|0 0 1|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In hexadecimal, this is 7B10484D CB986347 65ED42C4 1440000F 0001.

Decoding Location Configuration Information with Resolution

Decoding this option gives a latitude of 38.897647 (to 7 decimal places) with 18 bits of resolution; a longitude of -77.0366000 with 17 bits of resolution; an altitude type of meters with a value of 15 and 17 bits of resolution; version 0 (resolution) and the WGS84 datum.

For the latitude value, 18 bits of resolution allow for values in the range from 38.8964844 to 38.8984375. For the longitude value, 17 bits of resolution allow for values in the range from -77.0390625 to -77.0351563. Having 17 bits of resolution in the altitude allows for values in the range from 0 to 32 meters.

GML Representation of Decoded Location Configuration Information

The following GML shows the value decoded in the previous example as a point in a three dimensional CRS:

```
 <gml:Point srsName="urn:ogc:def:crs:EPSG::4979"
            xmlns:gml="http://www.opengis.net/gml">
   <gml:pos>38.897647 -77.0366 15</gml:pos>
 </gml:Point>
```

This representation ignores the values included in the resolution parameters. If resolution values are provided, a rectangular prism can be used to represent the location.

The following example uses all of the decoded information from the previous example:

```
<gs:Prism srsName="urn:ogc:def:crs:EPSG::4979"
    xmlns:gs="http://www.opengis.net/pidflo/1.0"
    xmlns:gml="http://www.opengis.net/gml">
  <gs:base>
```

```
    <gml:Polygon>
     <gml:exterior>
        <gml:LinearRing>
         <gml:posList>
            38.8964844 -77.0390625 0
            38.8964844 -77.0351563 0
            38.8984375 -77.0351563 0
            38.8984375 -77.0390625 0
            38.8964844 -77.0390625 0
          </gml:posList>
        </gml:LinearRing>
      </gml:exterior>
    </gml:Polygon>
```



```
  </gs:base>
  <gs:height uom="urn:ogc:def:uom:EPSG::9001">
    32
  </gs:height>
</gs:Prism>
```

B.2. Location Configuration Information of "Sears Tower" (Example 2)

Postal Address:
Sears Tower
103rd Floor
233 S. Wacker Dr.
Chicago, IL 60606

Viewing the Chicago area from the Observation Deck of the Sears Tower.

```
Latitude 41.87884 degrees North (or +41.87884 degrees)
Using 2s complement, 34 bit fixed point, 25 bit fraction
Latitude = 0x053c1f751,
Latitude = 0001010011110000011111011101010001
Longitude 87.63602 degrees West (or -87.63602 degrees)
Using 2s complement, 34 bit fixed point, 25 bit fraction
Longitude = 0xf50ba5b97,
Longitude = 1101010000101110100101101110010111

Altitude 103
```

In this example, we are inside a structure, therefore we will assume an Altitude value of 103 to indicate the floor we are on. The Altitude Type value is 2, indicating floors. The AltUnc field would indicate that all bits in the Altitude field are true, as we want to accurately represent the floor of the structure where we are located.

```
AltUnc = 30, 0x1e, 011110
AType = 2, 0x02, 000010
Altitude = 103, 0x00006700, 000000000000000110011100000000
```

For the accuracy of the Latitude and Longitude, the best information available to us was supplied by a generic mapping service that shows a single geo-loc for all of the Sears Tower. Therefore we are going to show LatUnc as value 18 (0x12 or 010010) and LongUnc as value 18 (0x12 or 010010). This would be describing a geo-location area that is Latitude 41.8769531 to Latitude 41.8789062 and extends from -87.6367188 degrees to -87.6347657 degrees Longitude. This is an area of approximately 373412 square feet (713.3 ft. x 523.5 ft.).

Appendix C. Calculations of Uncertainty

The following example demonstrates how uncertainty values for Latitude, Longitude, and Altitude (LatUnc, LongUnc and AltUnc used in the DHCPv6 Option as well as the version 1 DHCPv4 option) can be calculated.

C.1. Location Configuration Information of "Sydney Opera House" (Example 3)

This section describes an example of encoding and decoding the geodetic DHCP Option. The textual results are expressed in GML [OGC.GML-3.1.1] form, suitable for inclusion in PIDF-LO [RFC4119].

These examples all assume a datum of WGS84 (datum = 1) and an Altitude type of meters (AType = 1).

C.1.1. Encoding a Location into DHCP Geodetic Form

This example draws a rough polygon around the Sydney Opera House. This polygon consists of the following six points:

```
33.856625 S, 151.215906 E
33.856299 S, 151.215343 E
33.856326 S, 151.214731 E
33.857533 S, 151.214495 E
33.857720 S, 151.214613 E
33.857369 S, 151.215375 E
```

The top of the building 67.4 meters above sea level, and a starting Altitude of 0 meters above the WGS84 geoid is assumed.

The first step is to determine the range of Latitude and Longitude values. Latitude ranges from -33.857720 to -33.856299; Longitude ranges from 151.214495 to 151.215906.

For this example, the point that is encoded is chosen by finding the middle of each range, that is (-33.8570095, 151.2152005). This is encoded as (1110111100010010010011011000001101, 0100101110011011100010111011000011) in binary, or (3BC49360D, 12E6E2EC3) in hexadecimal notation (with an extra 2 bits of leading padding on each). Altitude is set at 33.7 meters, which is 000000000000000010000110110011 (binary) or 000021B3 (hexadecimal).

The Latitude Uncertainty (LatUnc) is given by inserting the difference between the center value and the outer value into the formula from Section 2.3.1. This gives:


```
x = 8 - ceil( log2( -33.8570095 - -33.857720 ) )
```

The result of this equation is 18, therefore the uncertainty is encoded as 010010 in binary.

Similarly, Longitude Uncertainty (LongUnc) is given by the formula:

```
x = 8 - ceil( log2( 151.2152005 - 151.214495 ) )
```

The result of this equation is also 18, or 010010 in binary.

Altitude Uncertainty (AltUnc) uses the formula from Section 2.4.4:

```
x = 21 - ceil( log2( 33.7 - 0 ) )
```

The result of this equation is 15, which is encoded as 001111 in binary.

Adding an Altitude Type of 1 (meters) and a Datum of 1 (WGS84), this gives the following DHCPv4 form:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Code (123)  |  OptLen (16)  |  LatUnc   |      Latitude     .
 |0 1 1 1 1 0 1 1|0 0 0 1 0 0 0 0|0 1 0 0 1 0|1 1 1 0 1 1 1 1 0 0.
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .                Latitude (cont'd)              |  LongUnc  |   .
 .0 1 0 0 1 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 1 1 0 1|0 1 0 0 1 0|0 1.
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .                       Longitude (cont'd)                      |
 .0 0 1 0 1 1 1 0 0 1 1 0 1 1 1 0 0 0 1 0 1 1 1 0 1 1 0 0 0 0 1 1|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | AType |   AltUnc  |                Altitude                     .
 |0 0 0 1|0 0 1 1 1 1|0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1.
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .  Alt (cont'd) |Ver| Res |Datum|
 .1 0 1 1 0 0 1 1|0 1|0 0 0|0 0 1|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In hexadecimal, this is 7B104BBC 49360D49 2E6E2EC3 13C00021 B341. The DHCPv6 form only differs in the code and option length portion.

C.1.2. Decoding a Location from DHCP Geodetic Form

If receiving the binary form created in the previous section, this section describes how that would be interpreted. The result is then represented as a GML object, as defined in [GeoShape].

A Latitude value of 1110111100010010010011011000001101 decodes to a value of -33.8570095003 (to 10 decimal places). The Longitude value of 0100101110011011100010111011000011 decodes to 151.2152005136.

Decoding Tip: If the raw values of Latitude and Longitude are placed in integer variables, the actual value can be derived by the following process:

1. If the highest order bit is set (i.e. the number is a twos complement negative), then subtract 2 to the power of 34 (the total number of bits).

2. Divide the result by 2 to the power of 25 (the number of fractional bits) to determine the final value.

The same principle can be applied when decoding Altitude values, except with different powers of 2 (30 and 8 respectively).

The Latitude and Longitude Uncertainty are both 18, which gives an uncertainty value using the formula from Section 2.3.1 of 0.0009765625. Therefore, the decoded Latitudes is -33.8570095003 +/- 0.0009765625 (or the range from -33.8579860628 to -33.8560329378) and the decoded Longitude is 151.2152005136 +/- 0.0009765625 (or the range from 151.2142239511 to 151.2161770761).

The encoded Altitude of 000000000000000010000110110011 decodes to 33.69921875. The encoded uncertainty of 15 gives a value of 64, therefore the final uncertainty is 33.69921875 +/- 64 (or the range from -30.30078125 to 97.69921875).

C.1.2.1. GML Representation of Decoded Locations

The following GML shows the value decoded in the previous example as a point in a three dimensional CRS:

```
<gml:Point srsName="urn:ogc:def:crs:EPSG::4979"
           xmlns:gml="http://www.opengis.net/gml">
  <gml:pos>-33.8570095003 151.2152005136 33.69921875</gml:pos>
</gml:Point>
```

The following example uses all of the decoded information from the previous example:

```
<gs:Prism srsName="urn:ogc:def:crs:EPSG::4979"
    xmlns:gs="http://www.opengis.net/pidflo/1.0"
    xmlns:gml="http://www.opengis.net/gml">
  <gs:base>
    <gml:Polygon>
```

```
          <gml:exterior>
            <gml:LinearRing>
              <gml:posList>
                -33.8579860628 151.2142239511 -30.30078125
                -33.8579860628 151.2161770761 -30.30078125
                -33.8560329378 151.2161770761 -30.30078125
                -33.8560329378 151.2142239511 -30.30078125
                -33.8579860628 151.2142239511 -30.30078125
              </gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </gs:base>
      <gs:height uom="urn:ogc:def:uom:EPSG::9001">
        128
      </gs:height>
    </gs:Prism>
```

Note that this representation is only appropriate if the uncertainty is sufficiently small. [GeoShape] recommends that distances between polygon vertices be kept short. A GML representation like this one is only appropriate where uncertainty is less than 1 degree (an encoded value of 9 or greater).

Appendix D. Changes from RFC 3825

This section lists the major changes between [RFC3825] and this document. Minor changes, including style, grammar, spelling and editorial changes are not mentioned here.

- o Section 1 now includes clarifications on wired and wireless uses.
- o The former Sections 1.2 and 1.3 have been removed. Section 1.2 now defines the concepts of uncertainty and resolution, as well as conversion between the DHCP option format and PIDF-LO.
- o A DHCPv6 option is now defined (Section 2.1) as well as a DHCPv4 option (Section 2.2).
- o The former Datum field has been split into three fields: Ver, Res and Datum. These fields are used in both the DHCPv4 and DHCPv6 options.
- o Section 2.2.1 has been added, describing Version support.
- o Section 2.3 has been added, describing the Latitude and Longitude fields.
- o Section 2.3.1 has been added, covering Latitude and Longitude resolution.
- o Section 2.3.2 has been added, covering Latitude and Longitude uncertainty.
- o Section 2.4 has been added, covering values of the Altitude field (Sections 2.4.1, 2.4.2 and 2.4.3), Altitude resolution (Section 2.4.4), and Altitude uncertainty (Section 2.4.5).
- o Section 2.5 has been added, covering the Datum field.
- o Section 3 (Security Considerations) has added a recommendation on link layer confidentiality.
- o Section 4 (IANA Considerations) has consolidated material relating to parameter allocation for both the DHCPv4 and DHCPv6 option parameters.
- o The material formerly in Appendix A has been updated and shortened and has been moved to Appendix B.
- o An Appendix A on GML mapping has been added.
- o Appendix C has been added, providing an example of uncertainty encoding.
- o Appendix D has been added, detailing the changes from RFC 3825.

Authors' Addresses

James M. Polk
Cisco Systems
2200 East President George Bush Turnpike
Richardson, Texas 75082 USA
USA

EMail: jmpolk@cisco.com

John Schnizlein

EMail: john@schnizlein.org

Marc Linsner
Cisco Systems
Marco Island, FL 34145 USA
USA

EMail: marc.linsner@cisco.com

Martin Thomson
Andrew
PO Box U40
Wollongong University Campus, NSW 2500
AU

EMail: martin.thomson@andrew.com

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052 USA
USA

EMail: bernarda@microsoft.com