IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LCI corrections | | | | |
| Date: 2014-05-5 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Gabor Bajko | Mediatek |  |  | Gabor.bajko@mediatek.com |
|  |  |  |  |  |

Abstract

This document corrects some errors introduced by 14/0032 and corrects some additional text which was missed in 14/0032. It also includes instructions on MIB changes missing from 0032.

Baseline is 11mc v2.7

***Editor: add the following to section 1.5:***

The Round(x) function is the integer closest to x. Positive elements with a fractional part of 0.5 round up to the nearest positive integer. Negative elements with a fractional part of −0.5 round down to the nearest negative integer.

***Editor: make the following corrections to section 8.4.2.21.10:***

NOTE—This example shows how to encode the coordinates of the Sydney Opera House using the encoding defined in

IETF RFC 6225. The building is a polygon with the following (latitude, longitude) coordinates:

( −33.856625°, +151.215906°)

(−33.856299°, +151.215343°)

(−33.856326°, +151.214731°)

(−33.857533°, +151.214495°)

(−33.857720°, +151.214613°)

(−33.857369°, +151.215375°)

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°), which is encoded as (1110111100010010010011011000001101,

0100101110011011100010111011000011) (2s complement, 9 bits before binary point, 25 after, MSB first).

These values can be derived as 234 – round (33.8570095 × 225) and round (151.2152005 × 225) respectively.

The latitude uncertainty (LatUnc) is given by inserting the difference between the center value and the outer value into

the formula from Section 2.3.2 of IETF RFC 6225. This gives:

LatUnc = 8 − ceil( log2( −33.8570095 − −33.857720 ) ). The result is 18, which is encoded as 010010.

Similarly, longitude uncertainty (LongUnc) is given by the formula:

LongUnc = 8 − ceil( log2( 151.2152005 − 151.214495 ) ). The result is also 18, which is encoded as

010010.

The top of the building is 67.4 meters above sea level, and a starting altitude of 0 meters above sea level is assumed. At the GPS coordinates of the Sydney Opera House, the sea level is approximately 22.5m below the WGS84 ellipsoid. Therefore, the lowest altitude of the building is -22.5m, while the highest altitude is (67.4-22.5)=44.9m from the WGS84 ellipsoid. The middle of the range is (44.9-22.5)/2=11.2m, which is encoded as 000000000000000000101100110011

(22 bits before binary point, 8 after, MSB first).

Altitude uncertainty is given by the height of the building. Since the building is 67.4 m above sea level and basement is assumed to be at sea level, the uncertainty to be encoded is 67.4 m;

Altitude uncertainty (AltUnc) uses the formula from Section 2.4.5 from IETF RFC 6225:

AltUnc = 21 − ceil( log2( 33.7 − 0 ) ), the result is 15, which is encoded as 001111 (MSB first).

The Altitude Type field is set to 1, indicating that the altitude is specified in meters.

The Datum field is set to 1, indicating WGS84 coordinates.

The RegLoc Agreement field is set to 0 for this example.

The RegLoc DSE field is set to 0 for this example.

The Dependent STA field is set to 0 for this example.

The Version field is set to 1, as that is the only value currently defined in IETF RFC 6225.

The LCI configuration information report for this example is encoded as (where underscores are used as field or octet delimiters):

010010\_1110111100010010010011011000001101\_010010\_0100101110011011100010111011000011\_0001

\_001111\_000000000000000000101100110011\_001\_0\_0\_0\_01 (binary, MSB first per field)

010010\_1011000001101100100100100011110111\_010010\_1100001101110100011101100111010010\_1000

\_111100\_110011001101000000000000000000\_100\_0\_0\_0\_10 (binary, LSB first per field)

01001010\_11000001\_10110010\_01001000\_11110111\_01001011\_00001101\_11010001\_11011001\_1101001

0\_10001111\_00110011\_00110100\_00000000\_00000000\_10000010 (binary, rearranged into octets, with LSB

first per octet)

01010010\_10000011\_01001101\_00010010\_11101111\_11010010\_10110000\_10001011\_10011011\_0100101

1\_11110001\_11001100\_00101100\_00000000\_00000000\_01000001 (binary, MSB first, per octet)

52 83 4d 12 ef d2 b0 8b 9b 4b f1 cc 2C00 00 41 (order over the PHY SAP, see 7.3.5 and 8.2.2)

−

***Editor: delete this text from the end of section 8.4.2.51:***

***Editor: replace instances of IETF RFC 6625 with IETF RFC 6225 throughout the draft document***

***Editor: replace all instances of ‘RFC6225’ and ‘RFC 6225’ with ‘IETF RFC 6225’ throughout the draft document***

***Editor: for the text approved in 14/0032 and 14/0541, use real minus symbols for negative numbers (and for subtraction), not hyphens.***

***Editor: Modify the MIB variables in Annex C.3 as follows (\* refers to either Longitude, Latitude, or Altitude):***

* Delete MIB variables dot11RMRqstLCI\*Resolution and delete the entries from Dot11RMRequestEntry; except for dot11RMRqstLCIAzimuthResolution
* Rename dot11LCIDSE\*Resolution to dot11LCIDSE\*Uncertainty
* Replace “\* resolution is 6 bits indicating the number of valid bits in the fixed-point value of \*.” with “\* uncertainty is defined in IETF RFC 6225” in the MIB variable description.
* Rename dot11LCI\*Resolution to dot11LCI\*Uncertainty, except for dot11LCIAzimuthResolution
* Replace “This attribute indicates the \* resolution as 6 bits indicating the number of valid bits in the fixed-point value of \*.” with “This attribute indicates the \* uncertainty as 6 bits.” In the MIB variable description
* Replace “resolution” with “uncertainty” in the dot11STALCI\*Uncertainty MIB variables’ description.
* Rename dot11APLCI\*Resolution to dot11APLCI\*Uncertainty, except for dot11APLCIAzimuthResolution
* Rename dot11LCI\*Resolution to dot11LCI\*Uncertainty, except dot11LCIAzimuthResolution
* Delete all MIB variables ending with ‘AltitudeFraction’
* Rename all MIB variables \*AltitudeInteger to \*Altitude