IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 802.11bi D2.0 CID 2035 | | | | |
| Date: 2025 - Sept | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Graham SMITH | SRT BlueHalo | Sunrise, FL, USA. | 916 799 9563 | gsmith@bluehalo.com |

Abstract

Proposed resolution on D2.0 related to CID 2035.

The proposal is to make clear that the positive offset delay ΔIT(*n*) can be zero.

Rev 0 presented 10/8/2025

Rev 1 as agreed after discussions at meeting 10/8/2025

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page | Line | Comment | Proposed | Resolution |
| 2035 | 99 | 35 | "The IT(n) is a positive offset delaying the effective start of an EPP epoch boundary by a pseudo random value,". This may on the face of it seem a good idea, but on reflection, is it? When the transition occurs, a 3rd party can tell which new MAC addresses are due to epoch as, e.g., no 4 w HS. Also, the natural variation of traffic patterns on devices will effectively spread out the apparent epoch boundary. The basic idea is to hide amoung all the STAs changing addresses, not when they do it. So it is debatable if the conciderable effort required to do this randomizing is really acheiving anything. | Do we need this complication? I would suggest no. Let's discuss. | REVISED Incorporate the changes as described in this document. |

**DISCUSSION**

In presentation 25/1632r0, it was argued that FA might be simplified. For example, it should be easy to set the offset delay ΔIT(*n*) to zero. This is particularly true for larger networks.

The formula for calculating ΔIT(*n*) is

ΔIT(*n*) = int (KDF-*Hash*-*Length*(PGTK, “ERCM”, Seed + (*n* × EpochInterval))) mod TimeRangeTU

Where TimeRangeTU is “the value in TU corresponding to the value of the Epoch Start Time Variation Range field multiplied by the number of TU in the Epoch Interval Unit field of the EPP Epoch Settings field”

The idea was that in the EPP Epoch Settings field, the Epoch Start Time Variation Range could be set to zero, and then ΔIT(*n*) = 0.

Unfortunately, this is not true. x mod 0, is indeterminate (divide by zero). This is supported in Clause “1.5 Terminology for mathematical, logical and bit operations”, which covers the MOD function:

*“x* mod *y* is the remainder when *x* is divided by *y*; this operator is not used in this standard if *y* is negative; theresult is positive even if *x* is negative. For example, 5 mod 3 is 2 and –5 mod 3 is 1.”

However, it is easily rectified, by stating that setting the Epoch Start Time Variation Range to zero, sets ΔIT(*n*) = 0.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CID 2035 RESOLUTION

REVISED

At P57L14, add edits as shown:

The Epoch Start Time Variation Range field contains the range of values, expressed in epoch interval units as defined in Table 9-129s (Epoch Interval Units and epoch durations), used by the AP MLD and each non- AP MLDs that is a member of the EPP group to determine a random delay added to the EPP epoch planned start time as defined in 10.71.2.4 (EPP Epoch Start Time Computation). A value of 0 indicates that no random delay is used.

At P99L61, add text as shown:

**10.71.2.4 EPP Epoch Start Time Computation**

To prevent an eavesdropper from easily determining the EPP Epoch start time, a pseudo random offset is computed and used by both the AP and each non-AP STAs of the EPP group.

At P100L24, add edits as shown.

EpochTSFStartTime(*n*) = PlannedEpochTSFStartTime(*n*) for the link + ΔIT(*n*)

Where,

PlannedEpochTSFStartTime(*n*) = FirstPlannedEpochTSFStartTime + (*n* – EpochNumberOffset) × EpochInterval mod 264

For all values of Epoch Start Time Variation Range field other than zero:

ΔIT(*n*) = int (KDF-*Hash*-*Length*(PGTK, “ERCM”, Seed + (*n* × EpochInterval))) mod TimeRangeTU

If the value of the Epoch Start Time Variation Range field is 0, then no computation of ΔIT(*n*) is needed, and ΔIT(*n*) = 0.