IEEE P802.11
Wireless LANs

|  |
| --- |
| CR for EDP Epoch Start Time |
| Date: 2025-01-15 |
| Author(s): |
| Name | Affiliation | Address | Phone | Email |
| Stéphane Baron | Canon Research centre France |  |  | Stephane.baron@crf.canon.fr |
| Julien Sevin |  |  | Julien.sevin@crf.canon.fr |
| Patrice Nezou |  |  | Patrice.nezou@crf.canon.fr |
|  |  |  |  |
| Po-Kai Huang | Intel |  |  |  |

Abstract

Abstract

This submission proposes resolutions and discussions for 23 CIDs number:

1031, 1050, 1051, 1052, 1080, 1244, 1245, 1246, 1247, 1248, 1249, 1250, 1251, 1252, 1253, 1254, 1256, 1259, 1260, 1265, 1266, 1267, 1518 on 802.11bi D0.4:

Revisions:

* Rev 0: Initial version of the document.
* Rev 1 : Addition of Option A (Link Offset based) and B (Epoch number offset based) in the discussion part.
* Rev 2 : update of the discussion part after merge of Option A and B, udpate of the comment resolution accordingly. + Clean-up of comment resolution instruction to editor.
* Rev 3 : Editorial modifications

1. Introduction

Interpretation of a Motion to Adopt

A motion to approve this submission means that the editing instructions and any changed or added material are actioned in the TGbi Draft. The introduction and the explanation of the proposed changes are not part of the adopted material.

The baseline for this text is 802.11 REVme D7.0, and 802.11 TGbi draft D0.7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CID | Page | Line | Comment | Proposed Change | Resolution |
| 1031 | 41 | 26 | It's not necessary to use PGTK to derive the GETn+1. The information is secured since the element is carried in a unicast protected action frame. | A simple pseudorandom integer drawn from a uniform distribution over the interval [0,RandTR] should be fine. | RejectedThe start times of the group EDP epochs are not exchanged but computed locally by the EDP AP MLDs and EDP non-AP MLDs. A shared secret is then needed, as input of the computation, in order to determine same value by both the AP MLD and the non-AP MLD.  |
| 1050 | 41 | 23 | Regarding the Next EDP Epoch Starting Time computed, is it needed to be synchronised between the AP and the STA? if so, there cannot be two different computation algorithms | If the calculation needs to be in sync, choose one algorithm. If the second one is used, we do not need to distribute PGTK | Revised.Agree in principle with the commenter. The calculation between the AP and the STA needs to be synchronized. For sake of simplicity, only one way to make the computation is proposed now. ***Instructions to the editor:*****Please make the changes as shown under CID 1050 in 11-24/1999r3.** |
| 1051 | 41 | 23 | GTn is not defined | GT is defined as the reference start time of the EDP Epoch, how do you know this value? | Revised.GTn is now replaced by PlannedTSFStartTime(n), and defined in clause 10.71.2.5 : GTn is replaced by n as the seed of the pseudo random generation***Instructions to the editor:*** **Please make the changes as shown under CID 1051 in 11-24/1999r3.** |
| 1052 | 41 | 30 | GEI cannot be added/multiplied/divided (see lines 30, 35 and 38) as defined | GEI as defined, Epoch Interval Duration, cannot be added as it is, since it has units which are not considered. | Revised.The unit of GEI (now EpochInterval) is the value expressed in TU of the Epoch Interval Duration field originally expressed in Epoch Interval Unit of the Epoch Interval field as specified in section 9.4.1.76 EDP Epoch Setting fields ***Instructions to the editor:*** **Please make the changes as shown under CID 1052 in 11-24/1999r3.** |
| 1080 | 41 | 48 | The term "GT" is not used | Please remove "GT" | Revised.Agree in principle, GT should be GTn and is now renamed as PlannedTSFStartTime(n) and defined in 10.71.2.5 (see CID 1051 resolution).***Instructions to the editor:*****Please make the changes as shown under CID 1051 in 11-24/1999r3.** |
| 1244 | 41 | 23 | "GTn+1" is not defined | As it says in the comment | Revised.Agree in principle, GTn+1 is now not used anymore and GTn is renamed by PlannedTSFStartTime(n) that is defined in 10.71.2.5 (see CID 1051 resolution).***Instructions to the editor:*****Please make the changes as shown under CID 1051 in 11-24/1999r3.** |
| 1245 | 41 |   | All this computational stuff shouldn't be in Clause 9 | As it says in the comment | Revised.Agree in principle with the commenter, computation is now in moved in new clause 10.71.2.5.***Instructions to the editor:*****First apply 1844r5 changes then apply modification** then **Please make the changes as shown under CID 1245 in 11-24/1999r3** |
| 1246 | 41 | 25 | What is \? | As it says in the comment | Revised.Agree in principle but the formula is not present anymore in the document***Instructions to the editor:*****No change needed** |
| 1247 | 41 | 38 | What is x? | As it says in the comment | Revised.Agree in principle with the commenter, change the letter “x” by ““Formula is revised according to the 802.11 editing style.***Instructions to the editor:*****Please make the changes as shown under CID 1247 in 11-24/1999r3.** |
| 1248 | 41 | 33 | I have no idea what "Or" means here | As it says in the comment | Revised.Agree in principle with the commenter, only one way to make the computation is proposed now. Alternative implementation corresponding to the "Or” part, is now removed.***Instructions to the editor:*****Please make the changes as shown under CID 1248 in 11-24/1999r3.**  |
| 1249 | 41 | 25 | I think PRF returns a bitstring, not a number, so it's not clear how you apply mod to it | As it says in the comment | Revised.Agree in principle with the commenter. “Int()” cast has been added according to clause 1.5, in addition, usage of PRF has been replaced by the KDF-Hash function***Instructions to the editor:*****Please make the changes as shown under CID 1249 in 11-24/1999r3.** |
| 1250 | 41 |   | Hyphens should be minuses in maths | As it says in the comment | Revised.Agree with the commenter. Minus now replaces hyphen.***Instructions to the editor:*****Please make the changes as shown under CID 1250 in 11-24/1999r3.** |
| 1251 | 41 | 50 | "the Epoch Interval Duration" missing "field" | As it says in the comment | Revised.Agree in principle with the commenter. “field” added after “Epoch Interval Duration”.***Instructions to the editor:*****Please make the changes as shown under CID 1251 in 11-24/1999r3.** |
| 1252 | 41 | 55 | Do we have TSF counters as opposed to TSF times? | As it says in the comment | Revised.Assuming commenter is referring to TSF timers, “TSF counter” replaced by “TSF timer” across the chapter***Instructions to the editor:*****Please make the changes as shown under CID 1252 in 11-24/1999r3.** |
| 1253 | 41 | 60 | "of EDP Epoch Settingsfield" missing "the" | As it says in the comment | Revised.Agree in principle with the commenter. “of EDP Epoch Settings” replaced by “of the EDP Epoch Settings”.***Instructions to the editor:*****Please make the changes as shown under CID 1253 in 11-24/1999r3.** |
| 1254 | 41 |   | It's not clear what an iteration of an EDP epoch sequence is | As it says in the comment | Revised.Iteration of an EDP Epoch is now replaced by the EDP epoch number that is the element number of a sequence, with first element number equal to 0 (see also CID1260 resolution).***Instructions to the editor:*****Please make the changes as shown under CID 1254 in 11-24/1999r3.** |
| 1256 | 42 | 9 | "GET" is not defined | As it says in the comment | Revised.GET is now replaced by EpochTSFStartTIme, and defined as follow in clause 10.71.2.5EpochTSFStartTime(n) : the link TSF timer value corresponding to the start time of the EDP epoch number n***Instructions to the editor:*****Please make the changes as shown under CID 1256 in 11-24/1999r3.** |
| 1259 | 42 | 13 | "The time range field is" should be "The Time Range field contains" and the units are not given. Also "the range used by the stations" is not clear. Also is "the stations" different from "the STAs", and which are "the stations" anyway? | As it says in the comment | Revised.As per discussion belowThe unit is the Epoch Interval Unit to be coherent with the unit selected for the duration of an EDP epochReplace "the stations" by non-AP MLD***Instructions to the editor:*****Please make the changes as shown under CID 1259 in 11-24/1999r3.**  |
| 1260 | 41 | 60 | "Next Epoch Start Time field" should be defined better than a passing mention in a "where" for some equations | As it says in the comment | Revised.Field name is now First Epoch TSF Start Time and a definition is provided***Instructions to the editor:*****Please make the changes as shown under CID 1260 in 11-24/1999r3.** |
| 1265 | 42 | 3 | This should be part of the where, i.e. indented | As it says in the comment | Revised.Agree in principle, indentation has been added in the new formula “where” statement.***Instructions to the editor:*****Please make the changes as shown under CID 1265 in 11-24/1999r3.** |
| 1266 | 42 | 9 | "If the effective start time GET of an EDP Epoch occurs during an ongoing TXOP, the Epoch starts at the end of this TXOP." -- how do receivers know this, if they missed all the frames in the TXOP? | As it says in the comment | Revised.Agree in principle with the commenter. A sentence indicating the behaviour if a start time occurs during an on going TXOP ***Instructions to the editor:*****Please make the changes as shown under CID 1266 in 11-24/1999r3.** |
| 1267 | 42 | 9 | Anyway, what is "the effective start time"? GET is previously defined as " the start time [...] of the next EDP Epoch of the sequence", with no effectiveness | As it says in the comment | Revised.The new formula account for resolution of this comment. “Effective” has been removed and GET renamed by EpochTSFStartTime. ***Instructions to the editor:*****Same resolution as CID 1256****Please make the changes as shown under CID 1256 in 11-24/1999r3.** |
| 1518 |   |   | A single offset calcualtion reduces operation overheads and improves STA and AP efficiency. |   | RejectedCommenter do not clearly indicate what calculation it is referring to.  |

Discussion: Simultaneous epoch start time across links.

For an MLD, there are different TSF values in each link, and those are the only TSF values that are maintained by an MLD. However, it is not clear that based on the current texts in the epoch operation which link that the TSF that the indication is referring to.



Fundamentally, epoch operation in each link is like TWT, and we should continue to have operations that can be done independently in each link so that the operations will be independent of reconfigurations.

**Proposed standardization direction:**

The proposal here provides simplified version of operation below without the need to refer back to the centralized timer and align with existing TWT operation so that we have aligned EPOCH across links

Step 1:

An AP MLD selects one of its links, sends directly the First planned Epoch Start time based on the TSF of the link for the exchange, epoch interval, and the index for the next epoch in the EDP epoch setup frame.

Upon reception of the EDP epoch setup frame, the non-AP STA of a non-AP MLD will store the Next planned epoch start time, the epoch interval, and the received Epoch number n for that link.

Note that today, all the timing indication is based on the indication of the TSF of a link, and this operation aligns with existing indication.



Step 2:

Non-AP MLD then constructs the corresponding time of other links.

The next planned epoch start time of link 1 and link 2 for epoch number n is as follows.

Next planned epoch TSF start time\_link\_1 = Next epoch TSF start time\_link\_0 + TSF difference of link 1 and link 0

Next planned epoch TSF start time\_link\_2 = Next epoch TSF start time\_link\_0 + TSF difference of link 2 and link 0

Note that TSF difference is calculated based on the TSF offset indicated by AP MLD across links



Step 3:

From now on, the non-AP STAs of each link can compute epoch TSF start time value independently.

For a link, future epoch number and associated start time is the following.

Planned EpochTSFStartTime(m) for the link = Planned EpochTSFStartTime(n) for the link + (m-n)EpochInterval

Actual EpochTSFStartTime(m) of the link = Planned EpochTSFStartTime(m) for the link + ΔIT

With:

ΔIT = int (KDF-Hash*-Length* (PGTK, "ERCM", m)) mod RandTR.

Note that if client wakes up from power save, client can continue to compute the next planned Epoch start time until the next epoch start time is larger than the current time.

***TGbi editor: Modify clause 9.4.1.76 as follow***

* EDP Epoch Setting fields (#1070)

The EDP Epoch Settings field includes the information regarding the actual parameters of an Epoch.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | EDP Epoch Settings Control | Group ID | Epoch Interval | First Epoch Start Time | Epoch number Offset (#1256) | Time Range | Epochs Remaining | Minimum Epoch Pacing Parameters  | Number Of participating Affiliated STAs |
| Bits: | 8 | 0 or 8 | 16 | 0 or 64 | 0 or 64 | 0 or 16 | 0 or 8 | 0 or 16 | 0 or 8 or 16 or 24 |

* EDP Epoch Settings field format(#Ed)

The EDP Epoch Settings field contains the EDP epoch parameters of an EDP epoch sequence for the non-AP MLD.

The Group ID field signals an identifier of the EDP group. Value 0 indicates the default group. Value 255 is reserved.

The EDP Epoch Settings Control is defined as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Group ID Present | First EpochTSF(#1260) Start TimePresent | Time RangePresent | Epochs RemainingPresent | Participating Affiliated STAs Count Present(#Ed) | Participating Affiliated STAs Percentage Present(#Ed) | Minimum Epoch Pacing Parameters Present | Reserved |
| Bits: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

* EDP Epoch Settings Control field format(#Ed)

Each of the bits of the EDP Epoch Settings Control field indicate the presence of the corresponding field in the EDP Epoch Settings field when set to 1 and its absence when set to 0.

The Epoch Interval field is defined as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Epoch Interval Unit | Epoch Interval Length | Reserved |
| Bits: | 3 | 11 | 2 |

* Epoch Interval field format(#Ed)

The Epoch Interval Length field contains the length of the EDP epoch, expressed in Epoch Interval Units, shown in Table 9-129g (Epoch Interval Units and epoch durations). Epoch Interval Length value 0 is reserved.

|  |  |  |
| --- | --- | --- |
| Epoch Interval Unit field value | Epoch Interval Unit | Max Epoch Duration (approx.) |
| 0 | 1000 s | 23 d 16 h 36 min 40 s |
| 1 | 1 s | 34 min 7 s |
| 2 | Reserved | N/A |
| 3 | Reserved | N/A |
| 4 | Reserved | N/A |
| 5 | Reserved | N/A |
| 6 | Reserved | N/A |
| 7 | Reserved | N/A |

* Epoch Interval Units and epoch durations

(#1260)The First Epoch TSF Start Time filed contains the value of the TSF timer of the receiving link at the start time of the first EDP epoch of the sequence (EDP epoch number = Epoch number Offset).

(#1256)

The Epoch number offset field value contains the offset between the AP epoch number and the non-AP STA epoch number . (10.71.2.6 : EDP epoch start time).

The Minimum Epoch Pacing field signals the minimum epoch duration value that the non-AP MLD can support. The format of the Minimum Epoch Pacing element is the same as the Epoch Interval field.

The Time Range(#Ed) field contains (#1259) the range of values, expressed in Epoch Interval Units (#1259) as defined in Table 9-129g (Epoch Interval Units and epoch durations), used by the AP and each non-AP stations member of the EDP group (#1259) to determine a random delay added to the EDP epoch planned start time (PlannedTSFStartTime) as defined in 10.71.2.6 (EDP epoch start time) (#1259).

The Epochs Remaining field indicates the number of EDP Epochs left in the sequence after the current epoch finishes, except 255, which means that the sequence duration is unlimited. The length of the Epoch Sequence Duration field is 1 octet.

The Number of Participating Affiliated STAs field is optional. When present, the field signals an indication of the number of affiliated STAs currently participating to this group EDP epoch on the current link.

|  |  |  |
| --- | --- | --- |
|  | Participating Affiliated STAs Count | Participating Affiliated STAs Percentage |
| Octets: | 2 | 1 |

* Number of Participating Affiliated STAs field format(#Ed)

The Participating Affiliated STAs Count field represents an indication of the number of affiliated STAs participating in the signaled group on the link. The Participating Affiliated STAs Percentage field, with values in the range of 0 to 100, represents an indication of the percentage of the associated affiliated STAs participating to the signalled group on the link. Values 101-255 are reserved.

***TGbi editor: Modify clause 10.71.2.3 as follow***

12.71.2.3 EDP epoch transitions operations (#1116)

[…]

(#1347, #1348)The next epoch boundary is derived (as described in 10.71.2.5 (EDP epoch start time computation (#1091))) from the value of the first epoch start time(#1349, #1095, #1116) defined in the EDP Epoch

Settings field of the Group Enhanced Privacy element of the (Re)Association Response frame or the EDP

epoch setting action response frame. The Epoch Interval Duration field of the same fields and frames defines

the interval of the following Group EDP epochs sequence.

(#1096, #1353)

A CPE non-AP MLD belonging to an EDP group(#1096) and the CPE AP MLD may calculate the new OTA

values to be used for the non-AP MLD in the next(#1354) group EDP epoch(#1030).

At the start of the new group(#1030) EDP epoch, the new anonymization parameters are used to anonymize

the selected OTA fields of all new(#1175) individual frames transmitted during the epoch(#1355).

To account for clock drifts, the CPE non-AP MLD and CPE AP MLD shall begin to accept individually

addressed frames that use the new anonymization parameters for a dot11EpochStartTimeMargin before the

start of the(#1356) new epoch. The CPE non-AP MLD and CPE AP MLD shall accept individually

addressed frames with the old anonymization parameters for dot11EpochTransitionTime after the start of

the new epoch. The rules of 10.71.2.1 (General(#1091)) apply for frame retransmissions and acknowledgments.

(#1245)***TGbi editor: Create clause 10.71.2.5 as follow***

10.71.2.5 EDP epoch start time computation (#1245)

To avoid an easy determination of the epoch stat time by an eavesdropper in a link, the start time of each EDP epoch in a link is determined by introducing a pseudo random variation around a planned start time occurring at a regular interval.

Upon reception on a link of an EDP Epoch Request frame or an (Re)Association Request frame, the AP may send in response to the requesting non-AP STA, an EDP element including the First planned Epoch Start time based on the TSF of the link, the epoch interval, and the Epoch number Offset set to the next epoch number of the EDP epoch sequence of the EDP group assigned to the non-AP STA.

Upon reception of an EDP Epoch Response frame, or of a (Re)Association Response frame containing an EDP element on a link, the non-AP STA of a non-AP MLD shall:

* Store the First planned epoch start time, the epoch interval, and set its epoch number for this Epoch to the value of the received Epoch number offset for that link.
* Constructs the corresponding First planned epoch start time of its other links according to the formula:

First planned epoch TSF start time of another link= First epoch TSF start time of the receiving link + TSF Offset value between the other link and the receiving link

Note: the TSF Offset value is the value received in the latest Basic Multi-Link element exchange.

At any point of time, for a given link, for any EDP epoch(#1243) (#1254) number n ( n > 0) in an EDP epoch(#1101) sequence, the link TSF timer value corresponding to the start time of the EDP epoch number n is called EpochTSFStartTime(n) (#1256), and is computed according to the formula:

(#1256) EpochTSFStartTime(n) = PlannedTSFStartTime(n) for the link + ΔIT

With

PlannedTSFStartTime(n) = First planned epoch TSF start time + (n (#1250) Epoch number offset) (#1247) EpochInterval (#1051)(#1244)

ΔIT = int (#1249) (KDF-Hash*-Length*(PGTK, "ERCM", n )) mod RandTR

(#1248)(#1050) (#1051)and where:

 n is the current number (#1254) of the EDP epoch in the EDP epoch sequence.

 PlannedTSFStartTime(n) is the TSF timer(#1252) value of the link corresponding to the start time of the EDP epoch number n in the EDP epoch sequence. (#1051)

 Epoch number offset is the value indicated in the Epoch number Offset field of EDP Epoch Settings Field

 EpochInterval is the value in TU corresponding to the (#1052)Epoch Interval Duration field (#1251) of the EDP Epoch Settings field

 KDF-Hash-*Length* is the key derivation function as defined in 12.7.1.6.2 (Key derivation function (KDF)) using the hash algorithm identified by the AKM suite selector (see Table 9-190 (AKM suite selectors)) .(#1249).

 *Length* is the number of bits to derive. 64-bits are derived for ΔIT.(#1249)

 First planned epoch TSF start time is the value of the First planned epoch TSF start time, computed upon reception of an EDP element by the STA based on the First Epoch Start Time value of thes EDP element of the (#1253) received EDP Epoch Settings Field.

 RandTR is the value in TU corresponding to the Time Range field, of the EDP Epoch Settings field

 PGTK (for Privacy GTK) (#1265)is the cryptographic key assigned by an EDP AP MLD that is used to manage the group EDP epoch, distributed to the EDP non-AP MLDs associated with the EDP AP MLD.

If the start time of an EDP epoch occurs during an ongoing TXOP, the FA parameters corresponding to the new EDP Epoch applies at the end of this TXOP. (#1266)

***TGbi editor: Modify clause 10.71.3 as follow***

**10.71.3 Establishing frame anonymization parameter sets** (#1002)

[…]

For a given EDP epoch, the EDP FA block shall be generated as:

EDP FA block =*KDF*-*Hash*-*Length*( KDK, "EDP CPE frame anonymization", n (#1051))

where:

EDP FA block is the block of bits which is partitioned into the sets of all possible values for each

EDP frame anonymization parameter

KDF-*Hash*-*Length* is the key derivation function as defined in 12.7.1.6.2 (Key derivation function

(KDF)) using the hash algorithm identified by the AKM suite selector (see Table

9-190 (AKM suite selectors))

KDK is the Key Derivation Key

n is the current number (#1254) of the EDP epoch in the EDP epoch sequence as defined in 10.71.2.5 EDP epoch start time computation *Length* is the total number of bits to derive. A total of 1728 bits are derived for a EDP

FA block.

***TGbi editor: Modify clause 10.71.4 as follow***

**10.71.4 Establishing BPE frame anonymization parameter sets(#1521)**

All associated BPE non-AP MLDs and the BPE AP MLD shall generate EDP BPE frame anonymization

parameters for a given EDP epoch by computing a single pseudorandom EDP BPE FA block which is partitioned

into a(#Ed) set of EDP BP frame anonymization parameters according to the following tables.(#Ed)

For a given EDP epoch, the EDP FA block shall be generated as:

EDP\_BPE\_FA\_block = KDF-*Hash*-*Length* (PGTK, "EDP BPE frame anonymization", n (#1051)),

where:

KDF-*Hash*-*Length* is the key derivation function as defined in 12.7.1.6.2 (Key derivation function

(KDF)) using the hash algorithm identified by the AKM suite selector (see Table

9-190 (AKM suite selectors)); Length is equal to 872 bits for offset calculation.

PGTK is the Privacy Group Transient Key.

n is the current number (#1051) of the EDP epoch in the EDP epoch sequence as defined in 10.71.2.5 EDP epoch start time computation.