Project	**IEEE 802.16e Broadband Wireless Access Working Group <http://ieee802.org/16>**	
Title	**Byte-alignment and Clean-up of MAC Management Messages**	
Date Submitted	**2006-07-06, Version: 1.0**	
Source(s)	**Samsung Electronics Co.**	Voice: +82-31-279-5587 E-mail: sungwook98.park@samsung.com
	POSDATA	Voice: +1-408-986-1140 E-mail: yehur@posdata-usa.com
Re:	Call for Maintenance Change Requests on IEEE Std 802.16	
Abstract	This document suggests changes in TGe Draft Document IEEE 802.16e-2005 as Byte-alignment and Clean-up of MAC Management message	
Purpose	Adopt into the current Maint TG draft	
Notice	This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.	
Release	The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.	
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:chair@wirelessman.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.	

Byte Alignment and Clean-up of MAC Mangement Messages

Sungwook Park*, Yeongmoon Son*, Yerang Hur**

Samsung Electronics*, Posdata**

1. Background

The current published standard [1] includes some MAC management messages not byte-aligned. Examples are MOB_SLP-RSP, MOB_BSHO-REQ, MOB_BSHO-RSP.

Moreover the format of some messages is incorrect. For examples, an enclosing brace is missing, the number of bits is incorrect or missing, and obsoleted message field still remains. We found such examples in MOB_SLP-REQ, MOB_SLP-RSP, MOB_SCN-REP, MOB_MSHO-REQ, and MBS-MAP.

Proposed correction is based on the following rationale:

a. Management messages are byte-aligned before if-clause, for-loop, or TLV starts if the modification does not increase the message size.
b. Enclosing braces are matched.
c. The number of bits of each message field is consistent with the text.
d. Missing message fields are inserted.
e. Obsoleted message fields are deleted.
f. Incorrect positions of the message fields are corrected.

Section 2 describes text changes to correct management message format.

2. Proposed Changes

[Modify Table 109c--Sleep-Request (MOB_SLP-REQ) message format on Page 112 as follows]

```
MOB_SLP-REQ_Message_format() {
    Management message type = 50                          : 8 bits
    Number of Classes                                     : 8 bits
    for (i=0; i< Number of Classes; i++) {
        Definition                                        : 1 bit
        Operation                                         : 1 bit
        Power_Saving_Class_ID                             : 6 bits
        if (Operation = 1) {
            ~~Start_frame_number~~                        : ~~6 bits~~
            ~~Reserved~~                                  : ~~2 bits~~
            Start_frame_number                            : 8 bits
        }
        if (Definition = 1) {
            Power_Saving_Class_Type                       : 2 bits
            Direction                                     : 2 bits
            Traffic_triggered_wakening_flag               : 1 bit
            Reserved                                      : 3 bits
            initial-sleep window                          : 8 bits
            listening-window                              : 8 bits
            final-sleep window base                       : 10 bits
            final-sleep window exponent                   : 3 bits
            Number_of_~~Sleep_~~CIDs                      : 3 bits
            for (i=0; i<Number_of_~~Sleep_~~CIDs; i++ {
                CID                                       : 16 bits
            }
        }
    }                                                     : (inserted bracket)
    TLV encoded information                               : variable
}
```

[Modify Table 109d--Sleep-Response (MOB_SLP-RSP) message format on Page 114 as follows]

```
MOB_SLP-RSP_Message_format() {
    Management message type = 51                          : 8 bits
    Number of Classes                                     : 8 bits
    for (i = 0; i < Number_of_Classes; i++) {
        …
        if (Sleep Approved =~~=~~ 1) {
            if (Operation = 1) {
                ~~Start_frame_number~~                    : ~~6 bits~~
```

```
                    Reserved                                    : 2 bits
                    Start_frame_number                          : 8 bits
                }
                if (Definition = 1) {
                    Power_Saving_Class_Type                     : 2 bits
                    Direction                                   : 2 bits
                    initial-sleep window                        : 8 bits
                    listening window                            : 8 bits
                    final-sleep window base                     : 10 bits
                    final-sleep window exponent                 : 3 bits
                    TRF-IND required                            : 1 bit
                    Traffic_triggered_wakening_flag             : 1 bit
                    Power_Saving_Class_Type                     : 2 bits
                    Direction                                   : 2 bits
                    TRF-IND required                            : 1 bit
                    Reserved                                    : 1 bit
                    if (TRF-IND required) {
                        SLPID                                   : 10 bits
                        Reserved                                : 2 bits
                    }
                    Number_of_CIDs                              : 4 bits
                    for (i = 0; i < Number_of_CIDs; i++) {
                        CID                                     : 16 bits
                    }
                    if (TRF-IND required) {
                        SLPID                                   : 10 bits
                        Reserved                                : 2 bits
                    }

                    …
                } ( end of definition =1)
                Padding                                         : variable
                …
            } else {  In case Sleep Approved == 0
                REQ-duration                                    : 8 bits
            }
        }                                                       : (inserted bracket)
        TLV encoded information                                 : variable
}
```

[Modify Table 109j—MOB_SCN-REP message format on Page 132 as follows]

```
MOB_SCN-REP_Message_format() {
    Management Message Type = 60                                : 8 bits
    Report Mode                                                 : 1 bit
    Comp_NBR_BSID_IND                                           : 1 bit
    N_current_BSs                                               : 3 bits.
```

Reserved : ~~3~~ 4 bits
Report metric : 8 bits
….
…..
N_Neighbor_BS_Index : 8 bits
If (N_ Neighbor_BS_Index!=0){
    Configuration change count for MOB_NBR-ADV : 8 bits
}
For(j=0;j<N_Neighbor_BS_Index;j++) {
    Neighbor_BS_Index : 8 bits
    If(Report metric[Bit 0]==1)
        BS CINR mean : 8 bits
    If(Report metric[Bit 1]==1)
        BS RSSI mean : 8 bits
    If(Report metric[Bit 2]==1)
        Relative delay : 8 bits
}
N_ Neighbor_BS_Full : 8 bits
For(j=0;j<N_Neighbor_BS_Full;j++) {
    Neighbor BSID : 48 bits
    If(Report metric[Bit 0]==1)
        BS CINR mean : 8 bits
    If(Report metric[Bit 1]==1)
        BS RSSI mean : 8 bits
    If(Report metric[Bit 2]==1)
        Relative delay : 8 bits
}
TLV encoded information : *variable*
}

[Modify Table 109m—MOB_MSHO-REQ message format on Page 145 as follows]

MOB_MSHO-REQ_Message_format() {
    Management Message Type = 57 : 8 bits
    …
    N_New_BS_Full **8 bits**
    For(j=0;j< N_New_BS_Full;j++){
        Neighbor_BS_ID ~~: 8 bits~~ **48** bits
        Preamble index/ Preamble Present and Subchannel Index : 8 bits
        If (Report metric [Bit#0] == 1)
            BS CINR mean : 8 bits
        If(Report metric[Bit#1]==1)
            BS RSSI mean : 8 bits
        If(Report metric[Bit#2]==1)
            Relative delay : 8 bits
        Service level prediction : 3 bits

Arrival Time Difference Indication : 1 bit
If (Arrival Time Difference Indication == 1) {
Arrival Time Difference (t) : 4 bits
}
}
…
}

[Modify Table 109n—BS HO Response (MOB_BSHO-RSP) message on Page 149 as follows]

MOB_BSHO-RSP_Message_format() {
Management Message Type = 58 :8 bits
Mode :3 bits
Reserved :5 bits.
If (Mode == 0b000) {
~~**HO operation mode** : 1 bit~~
N_Recommended :8 bits
HO operation mode :1 bit
Resource Retain Flag :1 bit
~~***Reserved*** : 7 bits~~
Reserved :6 bits
For (j=0 ; j<N_Recommended ; j++) {
Neighbor BSID :48 bits
Preamble index/ Preamble Present and Subchannel Index :8 bits.
Service level prediction :8 bits
HO process optimization :8 bits
Network Assisted HO supported :1 bit
HO_ID_included_indicator :1 bit
HO_authorization policy indicator :1 bit
Reserved :5 bits
If (HO_ID_included_indicator == 1) {
HO_ID :8 bits
}
~~**HO_authorization policy indicator** 1 bit~~
~~*Reserved* 4 bits~~.

If (HO_authorization policy indicator == 1) {
HO_authorization_policy_support :8 bits
}
}
…
else if (Mode == 0b101) {
N_new_BSs :3 bits
N_CIDs :8 bits.
N_SAIDs :8 bits
for (i= 0; i < N_new_BSs; i++) {
Neighbor BSID :48 bits

```
                    TEMP_BSID                          :3 bits
                    for (j= 0;j<N_CIDs;j++) {
                        New CID for BS_i               :16 bits.
                    }
                    For(i=0; i<N_SAIDs; i++){
                        New SAID for BS_i              :16 bits.
                    }
                }
                N_current_BSs                          :3 bits
                for (i=0;i< N_current_BSs;i++) {
                    TEMP_BSID                          :3 bits.
                }
                TEMP_BSID_Anchor                       :3 bits.
                AK Change Indicator                    :1 bit.
            }
            else if (Mode == 0b110) {
                N_new_BSs                              :3 bits.
                N_CIDs                                 :8 bits.
                N_SAIDs                                :8 bits.
                for ( i= 0; i < N_new_BSs; i++) {
                    Neighbor BSID                      :48 bits
                    TEMP_BSID                          :3 bits
                    for (j= 0;j<N_CIDs;j++) {
                        New CID for BS_i               :16 bits
                    }
                    For(i=0; i<N_SAIDs; i++){
                    For(k=0; k<N_SAIDs; k++){
                        New SAID for BS_i              :16 bits
                    }
                …
}
```

[Modify Table 109o—HO Indication (MOB_HO-IND) message on Page 157 as follows]

```
MOB_HO-IND_Message_format() {
    Management Message Type = 59                       :8 bits
    Reserved                                           :6 bits
    Mode                                               :2 bits
    if (Mode == 0b00) {
        HO_IND_type                                    :2 bits
        Ranging_Params_valid_indication                :2 bits
        Reserved                                       :4 bits

        if (HO_IND_type == 0b00) {
            Target_BS_ID                               :48 bits
            Preamble index/ Subchannel Index           :8 bits
        }
```

```
        }
        if (Mode == 0b01) {
            MDHOFBSS_IND_Type                   :2 bits
            if (MDHOFBSS_IND_Type == 0b00) {
                Anchor BSID                     :3 bits
                Action time                     :8 bits
            }
        }
        if (Mode == 0b10) {
            MDHOFBSS_IND_Type                   :2 bits
            if (MDHOFBSS_IND_Type == 0b00) {
                Diversity Set Included Indicator    :1 bit
                if (Diversity Set Included Indicator ==1) {
                    Anchor BSID                 :3 bits
                    N_BSs                       :3 bits
                    For (j=0 ; j<N_BSs ; j++) {
                        Temp BSID               :3 bits
                    }
                }
                Action time                     :8 bits
            }
        }
        ~~Preamble index/ Subchannel Index~~ ~~8 bits~~
        Padding                                 :variable
        TLV encoded information                 :variable
}
```

[Modify Table 109q—MBS MAP message on Page 161 as follows]

```
MBS_MAP Message format (){
        Management Message Type = 62            : ~~4 bits~~ 8 bits
        Frame number                                :4 bits
        MBS_DIUC_Change_Count                   :8 bits
        #MBS_DATA_IE                            :4 bits
        For (i = 0; i<n; i++){  n = #MBS_DATA_IE
            MBS_DATA_IE                         :variable
        }
        #Extended_MBS_DATA_IE                   :4 bits
        For(i=0;i<n;i++) {  n = #Extended_MBS_DATA_IE
            Extended_MBS_DATA_IE()              :variable
        }
        #MBS_DATA_Time_Diversity_IE             :4 bits
        For(i=0; i<m; i++){  m = #MBS_DATA_Time diversity_IE
            MBS_DATA_Time_Diversity_IE()        :variable
        }
        If(!byte boundary){
```

	Padding Nibble	:4 bits
}		
TLV encoding element		
}		

[Modify Table 105--HARQ Compact_UL-MAP IE format for UIUC subchannel on Page 106 as follows]

Compact_UL-MAP_IE () {		
UL-MAP Type =4	:3 bits	
Reserved	:1 bit	
UIUC	:4 bits	
RCID_IE	:variable	
if (UIUC == 12) {		
OFDMA symbol offset	:8 bits	
Subchannel offset	:7 bits	
No. OFDMA symbols	:7 bits	
No. Subchannels	:7 bits	
Ranging method	:2 bits	
Reserved	:1 bit	
} else if (UIUC == 14) {		
CDMA_Allocation_IE()	:~~32 bits~~ 40 bits	
~~}~~		
} else if (UIUC == 15) {		
Extended UIUC dependent IE()	:variable	
} else {		
No. Subchannels	:8 bits	
Repetition coding indication	:2 bits	
Reserved	:2 bits	
}		
H-ARQ_Control_IE	:variable	
}		