IEEE P802.11  
Wireless LANs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proposed spec text for D0.1 | | | | |
| Date: 2018-01-05 | | | | |
| Author(s): | | | | |
| Name | Affiliation | Address | Phone | email |
| Alfred Asterjadhi | Qualcomm Inc. | 5775 Morehouse Dr, San Diego, CA 92109 | +1-858-658-5302 | aasterja@qti.qualcomm.com |
|  |  |  |  |  |
|  |  |  |  |  |

Abstract

This submission proposes resolutions for

*The WUR frame has the following format:*

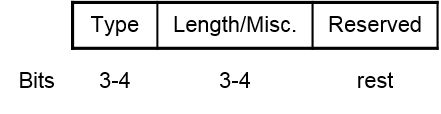
|  |  |  |
| --- | --- | --- |
| *MAC Header* | *Frame Body* | *FCS* |

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Frame Control* | *Address* | *TD Control* |
| *Bits* | *~~TBD~~ 8* | *~~TBD~~ 12* | *~~TBD~~ 12* |

* *The length of the MAC header is fixed.*
* *Whether the Address field contains more than one identifier is TBD.*

*The Frame Control field is 8 bits and contains:*

* *A Type subfield that identifies the type and, together with the Length field differentiates between constant length (CL) and variable length (VL) WUR frames*
* *A Length/Misc field which contains:*
  + *The length of the Frame Body field for a VL WUR frame*
  + *Bits that are expected to be used for other purposes for a CL WUR frames*
    - *No Length field is present in CL WUR frames*

**

*A Type subfield identifies the WUR frame type:*

* *The Type subfield is contained in the Frame Control field of the MAC header.*
* *One Type subfield value assigned to WUR Beacon and one to wake-up frame.*

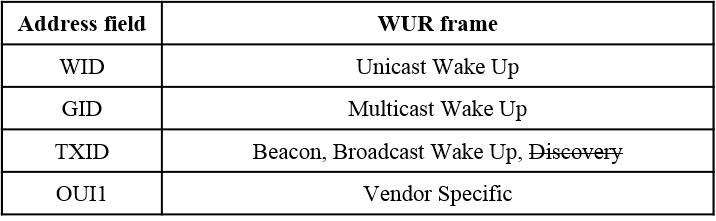
*TBD bits Type field is included in the Frame control field of MAC header with the following mapping of the Type field:*

* *0 assigned to WUR Beacon*
* *1 assigned to Wake-Up frame*
* *2 assigned to Vendor specific frame*

*The Address field is 12 bits, and the TD Control field is 12 bits*

*[Motion 2, Nov 2017, see [6] [48]]*

*The contents of the Address field are as defined below:*

**

* *Where:*
  + *WID is the WUR ID provided by the AP and identifies one WUR STA*
  + *GID is the GROUP ID provided by the AP and identifies one or more WUR STAs*
  + *TXID is a transmitter identifier that is decided by the AP*
    - *Which bits, from where, and how they are selected is TBD*
  + *OUI1 is the 12 MSBs of the OUI*

*The Type Dependent (TD) Control field in the MAC header contains type dependent control information.*

*The WUR frame has an optionally present Frame Body field:*

* *It is optional for a STA to support reception of a frame with nonzero length Frame Body.*

*When the Frame Body field is present in a WUR frame then:*

* *The length of the Frame Body field is indicated by the Length subfield in the Frame Control field*
* *The length is in units of TBD octets, and is up to 8 or 16 (TBD) octets.*

*The WUR frame has a Frame Check Sequence (FCS) that carries the CRC of the frame:*

* *Length and computation of FCS is TBD.*

*[Motion 6, Sep 2017, see [4] [37]]*

*Define a type of WUR frame as WUR Discovery frame to assist the STAs to discover the BSS.*

*[Motion, Nov 2017, see [6] [47]]*

Revisions:

* Rev 0: Initial version of the document.

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 TGax Draft. This introduction is not part of the adopted material.

***Editing instructions formatted like this are intended to be copied into the TGba Draft (i.e. they are instructions to the 802.11 editor on how to merge the text with the baseline documents).***

***TGba Editor: Editing instructions preceded by “TGba Editor” are instructions to the TGba editor to modify existing material in the TGba draft. As a result of adopting the changes, the TGba editor will execute the instructions rather than copy them to the TGba Draft.***

**TGba Editor: *Change the paragraphs below of this subclause as follows (#CID):***

* MAC frame format for Wake Up Radio (WUR) frames

**9.10.1 Basic components**

Each Wake Up Radio (WUR) frame consists of the following basic components:

* A *MAC header*, which comprises frame control, address, and type dependent (TD) control fields;
* A variable-length *frame body*, which contains information specific to the frame *type*;
* An *FCS*,which contains an IEEE *TBD*-bit CRC.

**9.10.2 General WUR frame format**

Figure 9-AA1(WUR frame format) depicts the general MAC frame format for WUR frames.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B0   B7 | B8  B19 | B20  B31 |  |  |
| Frame Control | Address | TD Control | Frame Body | FCS |
| 8 | 12 | 12 | optional | *TBD* |
| Figure 9-AA1 – WUR frame format | | | | | |

The MAC header of the WUR frame consists of the Frame Control, Address, and TD Control fields, and is defined in 9.10.2.1 (MAC header).

The Frame Body field is optionally present in certain WUR frame types, and is defined in 9.10.2.4 (Frame Body field).

The FCS field is defined in 9.10.2.5 (FCS field).

The MAC header and the last field (FCS) constitute the minimal WUR frame format and are present in all WUR frames, including reserved types.

A WUR frame that does not have a Frame Body field is referred to as a constant-length (CL) WUR frame. A WUR frame that has a Frame Body field is referred to as a variable-length (VL) WUR frame.

NOTE—A CL WUR frame can be sent to any WUR STA while a VL WUR frame can be sent only to a WUR STA that has declared support of its reception (see X.Y.Z)

**9.10.2.1 MAC header**

**9.10.2.1.1 Frame Control field**

The general format of the Frame Control field of the WUR frame is illustrated in Figure AA2 (Frame Control field format of WUR frame).

|  |  |  |
| --- | --- | --- |
| B0   BX | BX+1  BY |  |
| Type | Length/Misc | Reserved |
| 3-4 | 3-4 |  |
| Figure 9-AA2 – Frame Control field format of WUR frame | | |

The Type field contains the type of the WUR frame, as defined in Table JK1 (WUR frame types).

|  |  |
| --- | --- |
| Table JK1-- WUR frame types | |
| Type | Type description |
| 0 | WUR Beacon |
| 1 | WUR Wake Up |
| 2 | WUR Vendor Specific |
| 3 | WUR Discovery |
| 4-*TBD* | Reserved |

The Length/Misc field contains the Length field for a VL WUR frame and the Misc field for a CL WUR frame.

The Length field contains the length of the Frame Body field as defined in 9.10.2.4 (Frame Body field). The Misc field contains bits that are expected to be used for other purposes and is *TBD*.

**9.10.2.2 Address field**

The Address field contains an identifier for the WUR frame, which is selected from Table JK2 (Identifiers of WUR frames). The identifier depends on the type of WUR frame (see 9.10.3 (Format of individual WUR frame types).

|  |  |
| --- | --- |
| Table JK2—Identifiers of WUR frames | |
| Address field | Identifier description |
| Transmit ID | Identifier of the transmitting AP (provided by the AP to the WUR STAs as defined in *TBD*) |
| Group ID | Identifier of a group of receiving WUR STAs (provided by the AP to the group of WUR STAs as defined in *TBD*) |
| Wake Up ID | Identifier of an individual receiving WUR STA (provided by the AP to the WUR STA as defined in *TBD*) |
| OUI1 | The 12 MSBs of the OUI (see 9.4.1.32 (Organization Identifier field)) |

**9.10.2.3 TD Control field**

The Type Dependent (TD) Control field contains control information that dependends on the WUR frame type (see 9.10.3 (Format of individual WUR frame types).

**9.10.2.4 Frame Body field**

The Frame Body field is a variable-length field that contains information specific to specific individual WUR frame types. The Frame Body field is not present within CL WUR frames and is present VL WUR frames.

The length of the Frame Body field is in units of *TBD* octets and is equal to *TBD* x (*L* + 1), where *L* is the value of the Length subfield in the Frame Control field. The maximum length of the Frame Body field is 8 or 16 octets (*TBD*).

**9.10.2.5 Frame Check Sequence (FCS) field**

The FCS carries the CRC of the frame and its length and computation is TBD.

**9.10.3 Format of individual WUR frame types**

**9.10.3.1 WUR Beacon frame format**[Rojan]

**9.10.3.2 WUR Wake Up frame format**

[Jeongki]

**9.10.3.3 WUR Discovery frame format**

[Guoqing]

**9.10.3.4 WUR Vendor Specific frame format**

[Po-Kai]