**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Wireless Personal Area Networks | |
| Title | **15.4 MAC Timing Resolution Proposed Approach** | |
| Date Submitted | 1st May, 2014 | |
| Source | Clint Powell [PWC LLC]  Benjamin Rolfe [BCA] Noriyuki Sato [OKI] | Voice: [] Fax: [ ] E-mail: [cpowell@ieee.org]  [ben.rolfe@ieee.org]  [sato652@oki.com] |
| Re: | 802.15.4 Revision Preparation: Proposed Approach to MAC Timing Resolution | |
| Abstract | Submission to Maintenance standing committee: Summary of small group call and proposed approach for resolution of MAC timing. | |
| Purpose | Help resolve MAC timing issue. | |
| Notice | This document has been prepared to assist the IEEE P802.15. 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 acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. | |

**Notes from Call on 15.4 Revision - Timing Issues**

Timing issues to be resolved in the 802.15.4 revision are highlighted in red in doc. 15-14-0111. These include:

Constants

* *aMaxSIFSFrameSize*
* *aTurnaroundTime*

MAC Attributes

* *macAckWaitDuration*
* *macLIFSPeriod*
* *macSIFSPeriod*

In Beijing it was decided, based on the recommendation from doc. 15-14-0118,that *macLIFSPeriod* will be left as is. This leaves 4 items remaining to be addressed.

Based on the discussion during the call and email following the call the following proposal was captured.

**Proposal**

It was proposed that we proceed (as suggested at one time, possibly by James) to:

Specify the time to generate an ACK, and to let the determination of how long to wait for it to be defined by the implementer. This means:

*macAckWaitDuration* would be removed

*macEnhAckWaitDuration* would also potentially be removed

These values are used in several equations, which would need addressing or removed.

Referencing 802.15.4 2011, we find that:

* MAC ACK timing is specified in 5.1.6.4.2 and is where the cleanup is needed - at the very least we need to start here.
* 15.4 amendments e, g, k, and m all have changes to clause 5.1.6.4.2. So we need to see all of the text pulled together in one place first and then go from there.
* Clause 6.4.3 may also be involved.

**Rational Behind Proposal**

We currently specify quite clearly in 5.1.6.4.2 when the acknowledgement is transmitted, and with well defined reference points:

"The transmission of an acknowledgment frame in a nonbeacon-enabled PAN or in the CFP shall commence ZZ after the reception of the last symbol of the data or MAC command frame"

The only problems we have with this specification are:

1. nothing in the real world ever happens EXACTLY
2. ZZ is defined differently depending on the PHY and/or amendment you look at, and it really shouldn't be.

We really should specify a window one way or another. As it is now we have an implied window about the specified time and leave the allowable tolerance to be defined by whoever writes a test specification.

With this specification (and a spec. for how much variation we tolerate) one can design the logic to handle waiting for the ACK.  As an example, one would set a timer that starts after the last symbol of the data frame is transmitted, and stops as soon as it thinks it has received a preamble/SFD.  When the receiver locks either it is the expected acknowledgement, or it is something else. If it is something else, within the time the ACK is expected, then the ACK is missed and the transfer is declared a failure, and a retry is initiated.  If the timer expires, i.e. nothing is received within ZZ + tolerance, then the logic is finished.

This should be all we need to enable inter-operation.  We know when we can stop waiting for the ACK, turn off the receiver, and commence the retry logic. We also know how to test that the ACK is generated on time. What else do we need to know?

The problem we have with specifying the wait period the way we have it is that we are trying to take into account the transmission duration of the acknowledgement, and that is a variable that cannot be known since the ACK may have payload, IEs, a variable length preamble, etc. (none of which was true in 2006).

Also, we have specified that the device expecting the ACK MUST give up at specified time and can't be more generous in waiting. This causes a lot of problems doing conformance testing, and makes no real sense: why should a device "fail" conformance testing if it is more tolerant than required of timing errors?  If we get rid of *macAckWaitDuration* then we solve that problem, too.