

Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)

Submission Title: Analysis of inter-frame spacing in 802.15.4

Date Submitted: March 2014

Source: Cristina Seibert (Silver Spring Networks)

Contact Information: cseibert @ silverspringnet.com

Re: Presentation on inter-frame spacing in 802.15.4

Abstract: This contribution is prepared to support 802.15.4 revision.

Purpose:

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.

Today's 802.15.4

- Assumptions (stated or implied):
 - MAC needs time to process data received by the PHY
 - MAC may need more time if the frame is long compared to short, where short is defined as $\leq aMaxSIFSFrameSize = 18$ octets
 - MAC may be busy during this time, i.e. cannot receive another packet

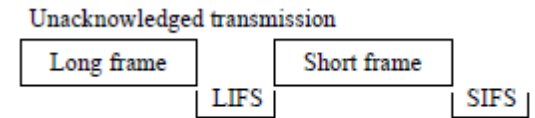
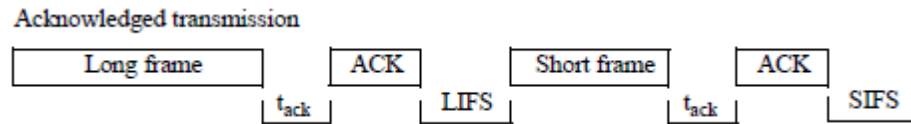


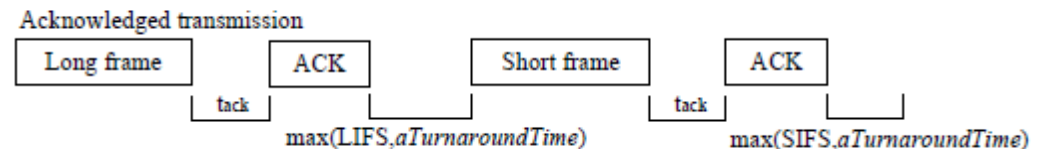
Figure 10—IFS

- Hence concept of Short Inter Frame Spacing, Long Inter Frame Spacing
- Minimum SIFS = 12 symbols, Minimum LIFS = 40 symbols, with further qualifications for some PHYs

- Gap required also in acknowledged case, following the ACK:



- 4m amendment fix to ensure time no shorter than $aTurnaroundTime$:



Interpretation

- Seems logical to think some MACs may “black out” while processing a frame, for time amounts proportional to the frame size.
- Some limitations can be buffer pipeline logic, e.g. delay from moving the frame to different memory location for upper layer processing, or inability to free-up while MAC processing completes, etc.
- MAC still busy for same amount after ACK generation? Perhaps could have been shorter (e.g. CRC calculation already completed) but timing simplified to reuse of already defined IFSSs?
- Evidently this doesn’t protect for the case where frames sent by multiple devices to the same receiver – those received during “blackout” could get dropped on the floor?
- *However, another interpretation is: diversity in IFSSs sizing is a **means to control channel access fairness!** In other words: a device that has just used up the channel with a long frame needs to wait a longer period (LIFS) to re-access the channel.*

Summary of Proposed Changes

- Some suggestions to consolidate LIFS and SIFS into one IFS. Rationale appears to be along the lines of “spec clean-up” – e.g. why keep features that are not needed?
- A further question can be why need IFS in the first place? Devices today can support back-to-back packet reception.
- At some point device would run out of buffer memory, but that can be the case even when IFS are used..
- Devices have more storage than in the old days, but is there a need for protection still, and does it need to be dependent on the packet size received?
- Depending on the PHY, the maximum supported PSDU size is 127 or 2047 octets. Should there be any dependencies on *aMaxPHYPacketSize*?
- Yet, it seems desirable to have diversity in IFS sizing as a form of channel access fairness.

Recommendation

- If IFS sizing changed, concern on compatibility with older devices whose receivers may have been built to expect the IFS timing in the current specification.
- The change would not qualify as “fixing a feature that is broken”.
- Unlike frame exchanges where “frame version” can signal “old” or “new” specification, IFS are trickier to “redo”.
- Could consider tying IFS timing to version of frames exchanged and for those, use a new IFS, e.g. any two frames with frame version ≥ 3 may be spaced by the new IFS.
- However, with consolidation, would lose benefits of IFS diversity.
- *Recommendation: No change at this time.*