Technical Report

IEEE 802.15.4 for Wireless Sensor Networks: A Technical Overview

Anis Koubâa

Mário Alves

Eduardo Tovar

TR-050702

Version: 1.0

Date: 14 July 2005

IEEE 802.15.4 for Wireless Sensor Networks: A Technical Overview

Anis KOUBAA, Mário ALVES, Eduardo TOVAR

IPP-HURRAY!

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8340509

E-mail: {akoubaa, ffp}@dei.isep.ipp.pt

http://www.hurray.isep.ipp.pt

Abstract

Low-rate low-power consumption and low-cost communication are the key points that lead to the specification of the IEEE 802.15.4 standard. This paper overviews the technical features of the physical layer and the medium access control sub-layer mechanisms of the IEEE 802.15.4 protocol that are most relevant for wireless sensor network applications. We also discuss the ability of IEEE 802.15.4 to fulfill the requirements of wireless sensor network applications.

IEEE 802.15.4 for Wireless Sensor Networks: A Technical Overview

ANIS KOUBAA, MÁRIO ALVES, EDUARDO TOVAR
IPP-HURRAY! GROUP, POLYTECHNIC INSTITUTE OF PORTO (ISEP-IPP), PORTUGAL
akoubaa@dei.isep.ipp.pt

Abstract

Low-rate low-power consumption and low-cost communication are the key points that lead to the specification of the IEEE 802.15.4 standard. This paper overviews the technical features of the physical layer and the medium access control sub-layer mechanisms of the IEEE 802.15.4 protocol that are most relevant for wireless sensor network applications. We also discuss the ability of IEEE 802.15.4 to fulfil the requirements of wireless sensor network applications.

Keywords*: IEEE 802.15.4, MAC sub-layer, Physical Layer, Wireless Sensor Networks*

1. INTRODUCTION

The IEEE 802.15.4 protocol specifies the Medium Access Control (MAC) sub-layer and physical layer for Low-Rate Wireless Private Area Networks (LR-WPAN). Even thought this standard was not specifically developed for wireless sensor networks, it is intended to be suitable for them since sensor networks can be built up from LR-WPANs. In fact, the IEEE 802.15.4 protocol targets low-data rate, low power consumption, low cost wireless networking, with typically fits the requirements of sensor networks.

The IEEE 802.15.4 protocol is very much associated with the ZigBee protocol. In fact, the ZibBee Alliance, which is an organization with over 150 member companies (ref. April 2005) has been working in conjunction with IEEE (task group 4) in order to specify a full protocol stack for low cost, low power, low data rate wireless communications, as well as to foster its use worldwide. The ZigBee Specification, released in December 2004 and recently turned publicly available, specifies the protocol layers above IEEE 802.15.4, i.e. the network (including security services) and the application (including device objects and profiles) layers. A snapshot of the ZigBee/IEE 802.15.4 protocol architecture is presented in Fig. 1.

This Technical Report presents the most relevant characteristics of the IEEE 802.15.4 protocol, in the context of wireless sensor networks.

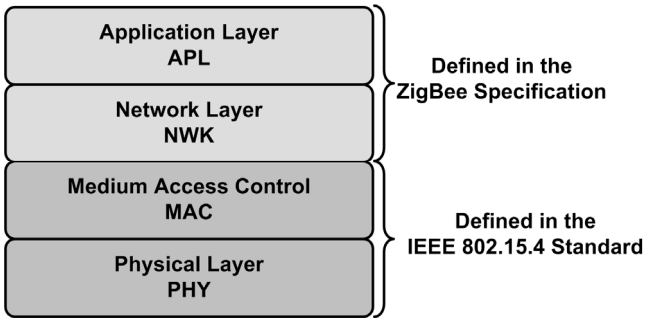

Fig.1. IEEE820.15.4/ZigBee protocol stack architecture

2. GENERAL DESCRIPTION OF IEEE 802.15.4

2.1 *Network Devices*

According to the IEEE 802.15.4 standard, a LR-WPAN supports two different types of devices:

- ***Full Function Device*** (FFD): a FFD is a device that can support three operation modes, serving as:
  - A *Personal Area Network (PAN) Coordinator*: the principal controller of the PAN. This device identifies its own network, to which other devices may be associated.
  - A *Coordinator*: provides synchronization services through the transmission of beacons. Such a coordinator must be associated to a PAN coordinator and does not create its own network.
  - A simple *device*: a device which does not implement the previous functionalities.
- ***Reduced Function Device*** (RFD): the RFD is a device operating with minimal implementation of the IEEE 802.15.4 protocol. An RFD is intended for applications that are extremely simple, such as a light switch or a passive infrared sensor; they do not have the need to send large amounts of data and may only associate with a single FFD at a time.

A LR-WPAN must include at least one FFD acting as a PAN coordinator that provides global synchronization services to the network and manages potential FFDs and RFDs.

2.2 *Network Topologies*

Two basic types of network topologies are defined in the IEEE 802.15.4 standard, according to the networking requirements of the applications: the *star* topology and the *peer-to-peer* topology. A third type of topology – the *cluster-tree* topology – can be considered as a particular case of a peer-to-peer topology, but we will address it separately.

a. The Star Topology

In the star topology (Fig.2), a unique node operates as a PAN coordinator. For instance, if an FFD is activated it may establish its own network and become its PAN coordinator. The PAN coordinator chooses a PAN identifier, which is not currently used by any other network in the sphere of influence.

The communication paradigm is the star topology is centralized i.e., each device (FFD or RFD) joining the network and willing to communicate with other devices must send its data to the PAN coordinator, which dispatch them to the adequate destination devices.

Due to the power-consuming tasks of the PAN coordinator in the star topology, the IEEE 802.15.4 standard mentions that the PAN coordinator may be mains powered while other devices are more likely to be battery powered.

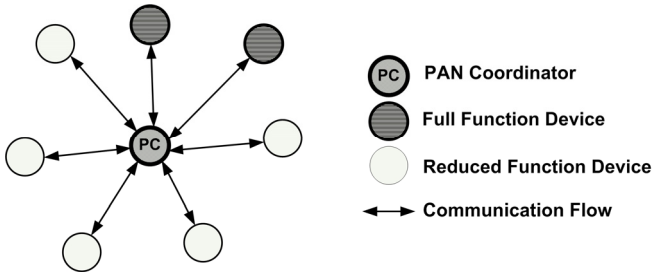

Fig.2. Star Topology Model

As a consequence, the star topology seems to be not adequate for traditional wireless sensor networks, since all sensor nodes are supposed to be battery-powered and thus very energy-constrained. A sensor node selected as a PAN coordinator will get its battery resources rapidly ruined. A potential bypass to this problem is to have a dynamic PAN coordinator based on remained battery supplies in sensor nodes, such as made in LEACH protocol [5]. However, this solution seems to be quite complex, since the dynamic election of a PAN coordinator among a large number of sensor nodes is not practical.

Taking these issues into consideration, the IEEE 802.15.4 standard recommends the star topology for applications such as home automation, personal computer peripherals, toys and games.

b. The Peer-to-Peer Topology

The peer-to-peer topology also includes a PAN coordinator, which is nominated, for instance, by virtue of being the first device to communicate on the channel. However, the communication paradigm in the peer-to-peer topology is decentralized, where each device can directly communicate with any other device in its radio range.

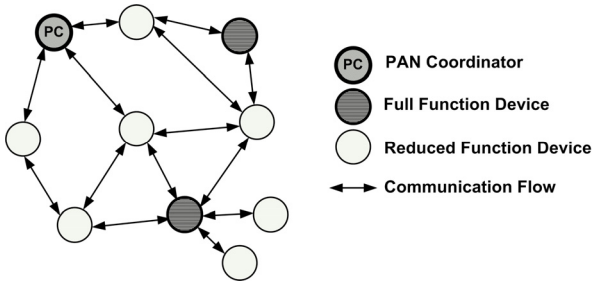

Fig. 3. Peer-to-Peer Topology Model

This mesh topology enables enhanced networking flexibility, but it induces an additional complexity for providing an end-to-end connectivity between all devices in the network. Basically, the peer-to-peer topology operates in ad hoc fashion and allows multiple hops to route data from any device to any other device. However, these functions must be defined at the Network Layer and therefore are not considered in the IEEE 802.15.4 specification.

Wireless Sensor Networks are one of the potential applications that may profit from such a topology. In contrast with the star topology, the resource usage is fairer in the peer-to-peer topology since the communication process does not rely on a particular node..

c. The Cluster-Tree Topology

The Cluster-Tree is a special case of a peer-to-peer network in which most devices are FFDs.

- One (and only one) coordinator is nominated as the PAN coordinator, which identifies the entire network.
- Any FFD may act as a coordinator and provide synchronization services to other devices or other coordinators,
- An RFD connects to a cluster-tree as a leave node at the end of a branch and associates itself with only one FFD.

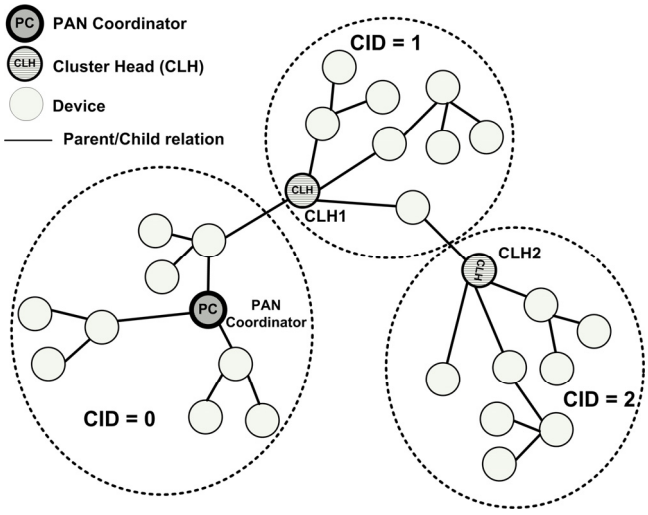

Fig. 4. Cluster-Tree Topology Model

Actually, the standard IEEE 802.15.4 (task group 4) [1] does not define how to build a cluster tree network. It only indicates that this is possible and may be initiated by higher layers. The cluster forming may be performed as follows:

- The PAN coordinator
  - forms the first cluster by establishing itself as *Cluster Head* (CLH) with a cluster identifier (CID) equals to zero
  - chooses an unused PAN identifier,
  - Broadcasts beacons to neighbouring devices
- A candidate device receiving a beacon frame may request to join the network to the CLH
  - If the PAN accepts the request to join the network, it adds the candidate device as a child device in its neighbour list. In turn, the new joined device adds the CLH as its parent in its neighbour list and starts transmitting periodic beacons. Other devices hearing these beacons may join the network at this device.
  - If for some reason the candidate device cannot join the network at the cluster head, it will search for another parent device.

For a large-scale network, it is possible to form a mesh of a multiple neighbouring clusters. In such a situation, the PAN coordinator can upgrade a device to become the CLH of a new cluster adjacent to the first one. Other devices gradually connect and form a multi-cluster network structure (see Fig. 4). The network layer defined in the ZigBee specification uses the primitives provided by the IEEE 802.15.4 MAC sub-layer and propose the cluster-tree protocol to for either a single cluster network or a potentially larger cluster tree network.

3. IEEE 802.15.4 PHYSICAL LAYER

The physical layer is responsible for data transmission and reception using a certain radio channel and according to a specific modulation and spreading technique.

The IEEE 802.15.4 offers three operational frequency bands: 2.4 GHz, 915 MHz and 868 MHz. There is a single channel between 868 and 868.6 MHz, 10 channels between 902 and 928 MHz, and 16 channels between 2.4 and 2.4835 GHz (see Fig. 5). The protocol also allows dynamic channel selection, a scan function that steps through a list of supported channels in search of a beacon, receiver energy detection, link quality indication and channel switching,.

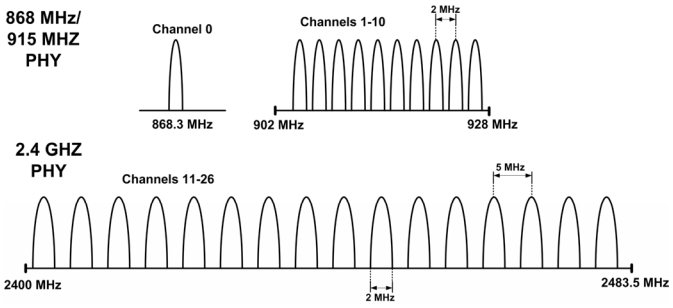

Fig. 5. Operating Frequency Bands

The data rate is 250 kbps at 2.4 GHz, 40 kbps at 915 MHZ and 20 kbps at 868 MHz. Lower frequencies are more suitable for longer transmission ranges due to lower propagation losses. Low rate transmissions provide better sensitivity and larger coverage area. Higher rate means higher throughput, lower latency or lower duty cycles. All of these frequency bands are based on the Direct Sequence Spread Spectrum (DSSS) spreading technique. The features of each frequency band (modulation, chip rate, bit rate …) are summarized in table 1.

Table 1. Frequency Bands and Data Rates

Frequency Band (MHz)	Spreading Parameters		Data Parameters		
	Chip rate (kchip/s)	Modulation	Bit rate (kbps)	Symbol rate (ksymbol/s)	Symbols
868	300	BPSK	20	20	Binary
915	600	BPSK	40	40	Binary
2400	2000	O-QPSK	250	250	16-ary

The physical layer of the IEEE 802.15.4 is in charge of the following tasks:

- *Activation and deactivation of the radio transceiver:* The radio transceiver may operate in one of three states: *transmitting*, *receiving* or *sleeping*. Upon the request of the MAC sub-layer, the radio is turned ON or OFF. The turnaround time from transmitting to receiving and vice versa should be no more than 12 symbol periods according to the standard (each symbol corresponds to 4 bits).
- *Energy Detection (ED) within the current channel*
  It is an estimation of the received signal power within the bandwidth of an IEEE 802.15.4 channel. This task does not make any signal identification or decoding on the channel. The energy detection time should be equal to 8 symbol periods. This measurement is typically used by the network layer as a part of channel selection algorithm or for the purpose of Clear Channel Assessment (CCA), to determine if the channel is busy or idle (see bellow).
- *Link Quality Indication* (*LQI*)
  The LQI measurement characterizes the Strength/Quality of a received packet. It measures the quality of a received signal on a link. This measurement may be implemented using receiver ED, a signal to noise estimation or a combination of both techniques. The LQI result may be used by the higher layers (Network and Application layers), but this procedure is not specified in the standard.
- *Clear Channel Assessment* (*CCA*)
  This operation is responsible for reporting the medium activity state: busy or idle. The CCA is performed in three operational modes:
  - *Energy Detection mode*: the CCA reports a busy medium if the detected energy is above the ED threshold.
  - *Carrier Sense mode*: the CCA reports a busy medium only is it detects a signal with the modulation and the spreading characteristics of IEEE 802.15.4 and which may be higher or lower than the ED threshold.

- *Carrier Sense with Energy Detection mode*: this is a combination of the aforementioned techniques. The CCA reports that the medium is busy only if it detects a signal with the modulation and the spreading characteristics of IEEE 802.15.4 and with energy above the ED threshold.

- *Channel Frequency Selection*
  The IEEE 802.15.4 defines 27 different wireless channels. A network can support only part of the channel set. Hence, the physical layer should be able to tune its transceiver into a specific channel request by a higher layer.

There are already commercially available sensor motes that are compliant with the IEEE 802.15.4. For instance, the MICAz mote from Crossbow Tech. provides a partial implementation of IEEE 802.15.4, operating at 2.4 GHz and 250 kbps. This mote uses 5 MHz for channel spacing conforming to the standard.

4. IEEE 802.15.4 MEDIUM ACCESS CONTROL

4.1 *General Description*

The MAC sub-layer of the IEEE 802.15.4 protocol provides an interface between the physical layer and the higher layer protocols of LR-WPANs.

The MAC sub-layer of the IEEE 802.15.4 protocol has many common features with the MAC sub-layer of the IEEE 802.11 protocol, such as the use of CSMA/CA (*Carrier Sense Multiple Access / Contention Avoidance*) as a channel access protocol, the support of contention-free and contention-based periods. However, the specification of the IEEE 802.15.4 MAC sub-layer is adapted to the requirements of LR-WPAN as, for instance, eliminating the RTS/CTS mechanism (used in IEEE 802.11) to reduce the probability of collisions, since collisions are more likely to occur in low rate networks.

The MAC protocol supports two operational modes that may be selected by the coordinator:

- *Beacon-enabled* mode: beacons are periodically generated by the coordinator to synchronize attached devices and to identify the PAN.A beacon frame is (the first) part of a superframe, which also embeds all data frames exchanged between the nodes and the PAN coordinator. Data transmissions between nodes are also allowed during the superframe duration.
- *Non Beacon-enabled* mode: in non beacon-enabled mode, the devices can simply send their data by using unslotted CSMA/CA. There is no use of a superframe structure in this mode.

Fig. 6 presents a structure of the IEEE 802.15.4 operational modes.

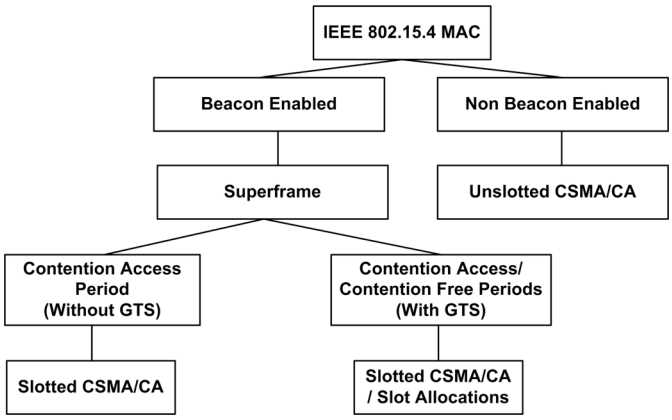

Fig. 6. IEEE 802.15.4 operational modes

In the following, we present the main characteristics of both beacon-enabled and non beacon-enabled modes.

4.2 *IEEE 802.15.4 Operational Modes*

a. The Beacon-enabled mode

When the coordinator selects the beacon-enabled mode, it forces the use of a superframe structure to manage communication between devices (that are associated to that PAN). The format of the superframe is defined by the PAN coordinator and transmitted to other devices inside every beacon frame, which is broadcasted periodically by the PAN coordinator. The superframe is divided into 16 equally sized slots and is followed by a predefined inactive period. The superframe structure is discussed in section 4.3.

As shown in Fig. 7 and Fig. 8, the superframe is contained in a Beacon Interval, which is bounded by two consecutive beacon frames, and includes one *Contention-Access Period* (CAP) and may include also a *Contention-Free Period* (CFP), as outlined next:

- If communications are restricted to the CAP (defined in the beacon, issued by the PAN Coordinator) a device wishing to communicate must compete with other devices using a slotted CSMA/CA mechanism. All transmissions must be finished before the end of the superframe, i.e., before the beginning of the inactive period (if exists). Refer to Fig. 7.

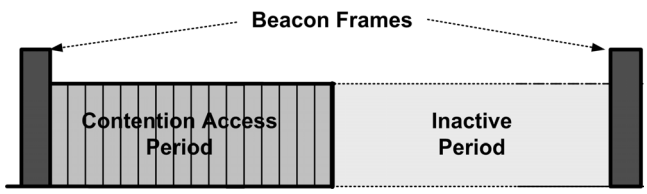

Fig. 7. The superframe structure without GTSs.

- If some guaranteed QoS is to be supported, then a *Contention-Free Period* (CFP) is defined. The CFP consists in *Guaranteed Time Slots* (GTSs) that may be allocated by the PAN coordinator to applications requiring low-latency or specific data bandwidth requirements. The CFP is a part of the superframe and starts at a slot boundary immediately following the

CAP, as shown is Fig. 8. The PAN coordinator may allocate up to seven GTSs and each GTS may occupy more than one time slot. With this superframe configuration, all contention-based communication must be finished before the start of the CFP, and a node transmitting a GTS must ensure that its transmission will be complete before the start of the next GTS (or the end of the CFP). According to the standard, the GTS is used only for communications between a PAN coordinator and a device. The GTS management are discussed in Section 5.7.

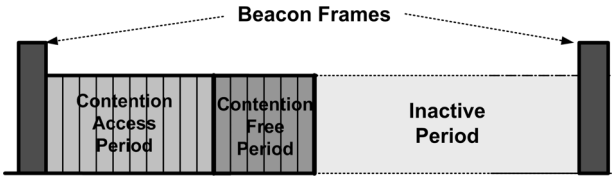

Fig. 8. The superframe structure with GTSs

In both configurations (CAP only or CAP/CFP), the superframe structure can have an *inactive period* during which the PAN coordinator does not interact with its PAN and may enter in a low power mode. Switching the network between activity/inactivity periods is very suitable for devices where reduced energy consumption is a main concern. In fact, the inactive periods enable the devices to save energy and thus extend network lifetime.

b. The Non Beacon-enabled Mode

When the PAN coordinator selects the non-beacon enabled mode, there are neither beacons nor superframes. Medium access control is provided by an unslotted CSMA/CA mechanism. All messages to be transmitted, with the exception of acknowledgment frames and any data frame that immediately follows the acknowledgment of a data request command (refer to Section 7.5.6.3 in [1]), must be dispatched according to this mechanism.

4.3 *The Superframe Structure*

The superframe is contained in a Beacon Interval bounded by two beacon frames, and has an active period and an inactive period (see Fig. 9). The beacon frame format is presented in Annex 1. The coordinator interacts with its PAN during the active period, and enters in a low power mode (sleep) during the inactive period.

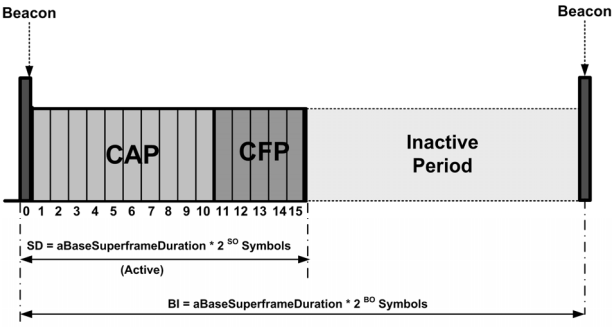

Fig. 9. Example of the structure of a Superframe

The structure of a superframe is defined by two parameters:

- *macBeaconOrder* (BO): this attribute describes the interval at which the coordinator must transmit beacon frames. The value of the *macBeaconOrder* and the *Beacon Interval* (BI) are related as follows:
  $for\ 0 \le BO \le 14,$
  $$BI = aBaseSuperframeDuration * 2^{BO}\ symbols$$

- *macSuperframeOrder* (SO): this attributes describes the length of the active portion of the superframe, which includes the beacon frame. The value of the *macSuperframeOrder* and the *Superframe Duration* (SD) are related as follows:
  $for\ 0 \le SO \le BO \le 14,$
  $$SD = aBaseSuperframeDuration * 2^{SO}\ symbols$$

If $SO = BO \Rightarrow SD = BI$ and then the superframe is always active. According to the standard, if $SO = 15$, the superframe will not be active following the beacon. Moreover, if $BO = 15$, then the superframe shall not exist and the network will operate in the non beacon-enabled mode. In this case, the value of SO is ignored.

As a result, a PAN that wishes to use the superframe structure must set *macBeaconOrder* to a value between 0 and 14 and *macSuperframeOrder* to a value between 0 and the value of *macBeaconOrder*. Otherwise, the PAN will operate in a non beacon-enabled mode with a value of *macBeaconOrder* and *macSuperframeOrder* equal to 15.

The active portion of each superframe is divided into *aNumSupeframeSlots*=16 equally spaced slots of duration $2^{SO} * aBaseSlotDuration$. The attribute $aBaseSlotDuration$ represents the number of symbols forming a superframe slot when the superframe order is equal to zero. The value of $aBaseSlotDuration$ is equal to 60 symbols.

The active portion of the superframe structure is composed of three parts:

- ***Beacon***: the beacon is transmitted without the use of CSMA at the start of slot 0. It contains the information on the addressing fields, the superframe specification, the GTS fields, the pending address fields, etc. For more details on the beacon frame, refer to Annex 1.
- ***CAP***: the CAP starts immediately after the beacon frame and ends before the beginning of the CFP (if it exists). Otherwise, the CAP ends at the end of the active part of the superframe. The minimum length of the CAP is fixed at $aMinCAPLength = 440\ Symbols$. This minimum length ensures that MAC commands can still be transferred to devices when GTSs are being used. A temporary violation of this minimum may be allowed if additional space is needed to temporarily accommodate the increase in the beacon frame length needed to perform GTS management. All the transmissions during the CAP are made using a slotted CSMA/CA mechanism to access the channel. However, the acknowledgement frames and any data that immediately follows the acknowledgement of a data request command are transmitted without contention. A device that cannot complete its transmission one *Inter Frame Spacing* (IFS: see Annex

3) period before the end of the CAP, must defer its transmission until the CAP of the next superframe.

- ***CFP***: The CFP starts immediately after the end of the CAP and must complete before the start of the next beacon frame. All the GTSs that may be allocated by the PAN coordinator are located in the CFP and must occupy contiguous slots. The CFP may therefore grow or shrink depending on the total length of all GTSs. The transmissions in the CFP are contention-free and therefore do not use a CSMA/CA mechanism to access the channel. Additionally, a frame may only be transmitted if the transmission ends one IFS before the end of the correspondent GTS.

4.4 *The CSMA/CA mechanisms*

The IEEE 802.15.4 defines two versions of the CSMA/CA mechanism:

- The *slotted CSMA/CA* version – used in the beacon-enabled mode.
- The *unslotted CSMA/CA* version – used in the non beacon-enabled mode.

In both cases, the CSMA/CA algorithm is based on backoff periods, where one backoff period is equal to $aUnitBackoffPeriod = 20\ Symbols$. This is the basic time unit of the MAC protocol and the access to the channel can only occur at the boundary of the backoff periods. In slotted CSMA/CA the backoff period boundaries must be aligned with the superframe slot boundaries where in unslotted CSMA/CA the backoff periods of one device are completely independent of the backoff periods of any other device in a PAN.

The CSMA/CA mechanism uses three variables to schedule the access to the medium:

- *NB* is the number of times the CSMA/CA algorithm was required to backoff while attempting the access to the current channel. This value is initialized to zero before each new transmission attempt.
- *CW* is the contention windows length, which defines the number of backoff periods that need to be clear of channel activity before starting transmission. CW is only used with the slotted CSMA/CA version. This value is initialized to 2 before each transmission attempt and reset to 2 each time the channel is assessed to be busy.
- *BE* is the backoff exponent, which is related to how many backoff period a device must wait before attempting to assess the channel activity.

Fig. 10 depicts a flowchart describing both versions of the CSMA/CA mechanism, which are described next.

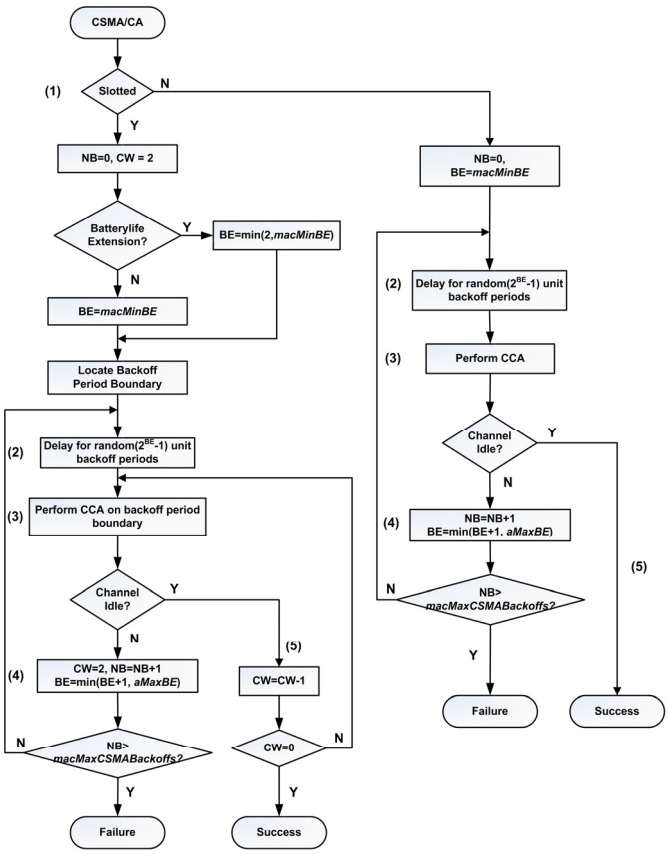

Fig. 10. The CSMA/CA Mechanism

i. The Slotted CSMA/CA Mechanism

The slotted CSMA/CA can be summarized in five steps:

- *Step 1- initialization of NB, CW and BE*: NB is initialized to 0 and the contention window *CW* is initialized to 2. Then the MAC protocol checks if the *macBattLifExt* attribute (see Annex 2) is set to *true*. In this case, the Backoff exponent *BE* is set to set to the minimum value of 2 or *macMinBE* attribute, otherwise BE is set to *macMinBE*. *macMinBE* attribute specifies the minimum of the backoff exponent, which is set to 3 by default. Note that when *macMinBE* is set to zero, collision avoidance is disabled during the first iteration of the algorithm, as it could be understood from step 2 in Fig. 10. After the initialization, the algorithm locates the boundary of the next backoff period.
- *Step 2- random waiting delay for collision avoidance*: the algorithm attempts to avoid collision by waiting during a given delay randomly generated in the range of $\left[0, 2^{BE} - 1\right]$ backoff periods. To disable the collision avoidance procedure at the first iteration, *BE* must be set to 0 and thus the waiting delay is null and the algorithm directly goes to step 3.

- *Step 3- Clear Channel Assessment* (CCA): the CCA must be started at a boundary of a backoff period just after the expiration of the waiting delay timer and repeatedly performs *CW* times a clear channel assessment before the access to the channel. If the channel is detected in a *busy* state, the algorithm goes to step 4, otherwise, i.e. the channel is idle, the algorithm goes to step 5.
- *Step 4 - busy channel*: if the channel is assessed to be *busy*, CW value is reset to 2 and the values of *NB* and BE are increased by one. However, *BE* cannot exceed *aMaxBE*, which is a constant defined in the standard, and with a default value equal to 5. If the number of retries exceeds *macMaxCSMABackoffs*, whose the default value is 5, the algorithm terminates with a channel access failure status, otherwise, i.e. the number of retries is below or equal to *macMaxCSMABackoffs*, the algorithm returns to step 2.
- *Step 5 - idle channel*: if the channel is assessed to be *idle*, the value of the contention window CW is decreased by one. If the contention window has expired ($CW = 0$), the MAC protocol **may** start successfully its transmission, otherwise, i.e. $CW \neq 0$, the algorithm returns to step 3. It is important to note that the transmission of the current frame is started only if the remaining number of backoff periods in the current superframe is sufficient to handle both the frame and the subsequent acknowledgement transmissions. Otherwise, the transmission of the frame is deferred until the next superframe.

ii. The Unslotted CSMA/CA Mechanism

The unslotted CSMA/CA is similar to the slotted version with some few exceptions.

- *Step 1*- A first exception, the *CW* variable is not used in the unslotted CSMA/CA. This is because the unslotted CSMA/CA has no need to iterate the CCA procedure after detecting an idle channel. Hence, in step 3, if the channel is assessed to be idle, the MAC protocol immediately starts the transmission of the current frame. Second, the unslotted CSMA/CA does not support *macBattLifeExt* mode and, hence, BE is always initialized to the *macMinBE* value.
- *Step 2 and Step 3* are exactly the same as those in the slotted CSMA/CA version. The only difference is that the CCA starts immediately after the expiration of the random backoff delay generated in step 2.
- *Step 4* is the same than that in the slotted CSMA/CA with the exception that the algorithm does not increase the value of CW. If ever *NB* exceeds the value of *macMaxCSMABackoffs*, the algorithm terminates in a failure state, otherwise, it returns to step 3.
- In *Step 5*, the MAC sub-layer starts immediately transmitting its current frame just after a channel is assessed to be *idle* by the CCA procedure.

5. STARTING AND MAINTAINING PANS

5.1 *How does a device start its own PAN?*

A PAN can be created by an FFD only after performing an active channel or an ED channel scan and choosing an appropriate PAN identifier. Channel scan procedures are explained in Annex 4. The standard does not define any algorithm for selecting a suitable PAN identifier from the list of PAN descriptors returned from the active channel scan. However, the standard proposes a *PAN identifier conflict resolution* mechanism when two PAN coordinators choose the same PAN identifier. This procedure is described in section 7.5.2.2 in [1].

The message sequence chart of a PAN Start procedure is presented in Fig. 11.

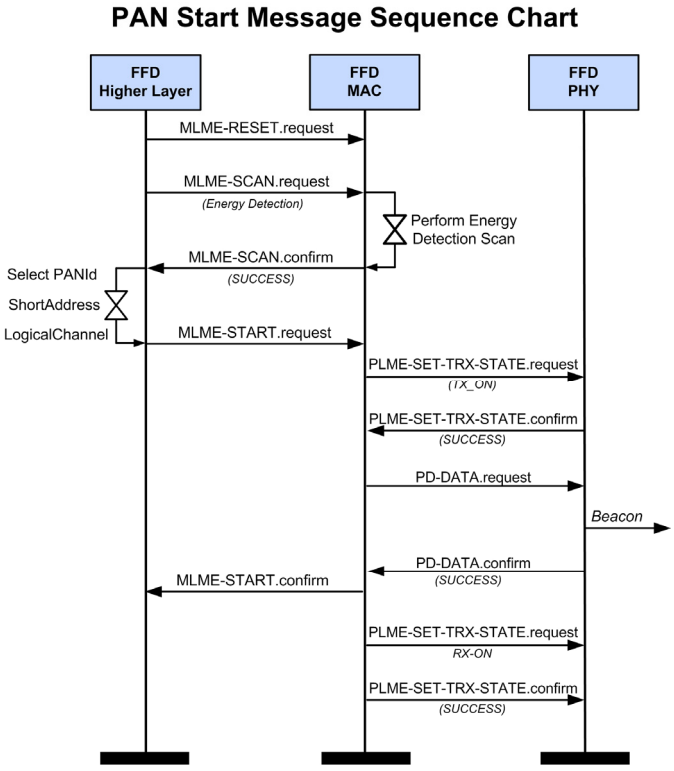

Fig. 11. PAN Start Message Sequence Chart

Note that the MLME stands for the MAC Sub-layer Management Entity, PLME stands for Physical Layer Management Entity, and PD stands for the Physical Data service. More information on these entities is detailed in [1].

In summary, the FFD higher layer generates the `MLME-RESET.request` primitive to reset the FFD to its initial conditions. Then, a scan request is sent to the FFD MAC sub-layer to perform an active channel scan or an energy detection scan with the *PANCoordinator* parameter set to TRUE and the *CoordRealignment* parameter set to FALSE. After receiving this request, the MAC sub-layer searches

for adequate channel and PAN identifier. After Completion of the scan, the MAC sub-layer responds with the `MLME-RESET.confirm` primitive and provides the PANId, the short address and the selected logical channel. The higher layer then requests the FFD MAC to start operating as a PAN coordinator. The MAC sub-layer requests the physical layer to enable its transceiver in *transmission mode* (TX_ON). After receiving the confirmation, the MAC sub-layer generates a data transmission request using the `PD-DATA.request` primitive and then, begins sending its beacon frame. In case of success, the physical layer sends a confirmation of the successful data transmission to the MAC sub-layer, which in its turn sends back a confirmation to the next higher layer. Finally, the transceiver is enabled in *receiving monde* (RX_ON) upon a request sent from the MAC sub-layer to receive data.

Once a PAN is started, it must be maintained by its PAN coordinator by generating and sending beacons frames, managing association and dissociation of other devices to the PAN, providing synchronization services, allowing GTS allocation and management, etc.

In the following, we present the basic operations that may occur in a PAN after being operational.

5.2 *Beacon Generation*

An FFD is permitted to generate and send beacon frames only if it satisfied at least one of the following conditions:

- The FFD is the PAN coordinator of a new PAN,
- The FFD is a device on a previously established PAN,
- The *macShortAddress* attribute of the FFD is not equal to 0xffff.

Beacon generation is performed using the `MLME-START.request` primitive, which contains the information on the *PANId*, *PANCoordinator* boolean value, in addition to *macBeaconOrder* and *macSuperFrameOrder* parameters. The latter parameters determine the duration of the beacon interval and the duration of the active and inactive portions.

The time at which the most recent beacon is transmitted, is recorded in *macBeaconTxTime* and must be computed so that its value is taken at the same symbol boundary in each beacon frame.

All beacon frames are transmitted at the beginning of each superframe at an interval equal to *aBaseSuperframeDuration* * 2^n symbols, where *n* is the *macBeaconOrder* (refer to Section 4.3).

Beacon transmission must be given priority over all other transmit and receive operations.

5.3 *Device Discovery*

According to the standard [1], once an FFD is successfully associated with a PAN, it may indicate its presence by sending beacon frames to allow other devices to perform device discovery.

However, this point is particularly confusing in the standard. In fact, if an FFD device, which is not the PAN Coordinator, starts sending beacon frames after being associated to a beacon-enabled PAN, then collisions between beacons from different coordinators will inevitably occur. As of July 2005, many proposals are being discussed within the framework of the IEEE 802.15.4b [2], which aims to carry specific enhancements and clarifications to the IEEE 802.15.4-2003 standard published in [1], to resolve such conflicts in the MAC sub-layer. This has been rectified in 15.4b by the addition of a start time parameter in the MLME-START.request primitive. This is the only proposal for beacon scheduling which has been adopted in by TG4b.

If it would be possible, the transmission of beacon frames by an FFD device other than a PAN coordinator is initiated through the use of the `MLME-START.request` primitive with the *PANCoordinator* parameter set to *FALSE*. Upon receipt of this primitive, the MLME begins transmitting beacons using the identifier of the PAN with which the device has associated *macPANId*, and its short address, *macShortAddress*.

5.4 *Association and Disassociation*

a. Association

When a device wants to join an existing network without creating a new PAN, it must be associated with an existing PAN.

A flowchart representing the association mechanism is shown in Fig. 12.

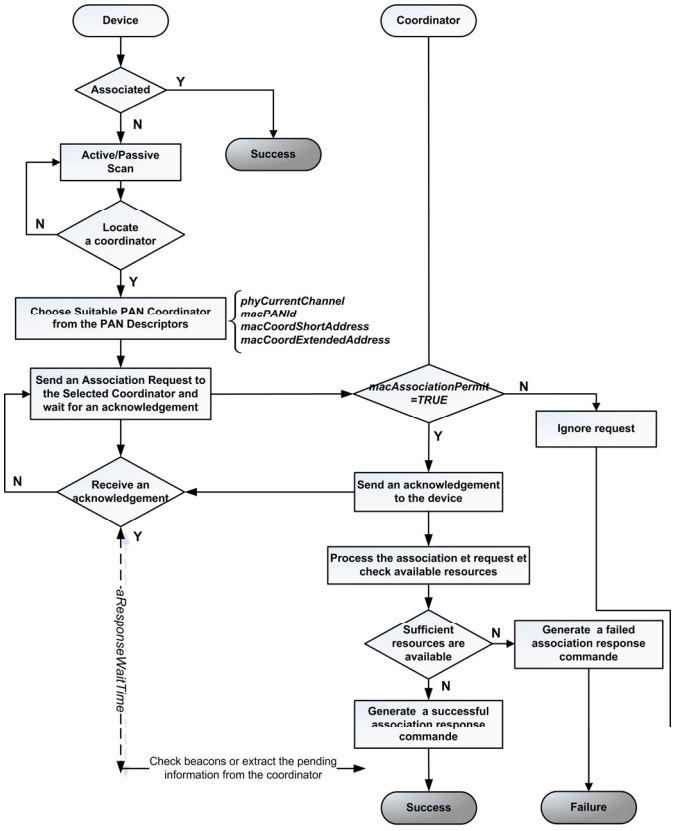

Fig. 12. The Association mechanism

The association process starts with an active or a passive scan. The passive scan, like the active scan, allows a device to locate any coordinator transmitting beacon

frames within a Personal Operating Space (POS), whereas the beacon request command is not required for passive scan. The results of the scan are then used to choose a suitable PAN characterized by its physical channel *phyCurrentChannel*, its identifier *macPANId*, its extended and short addresses *macCoordExtendedAddress* and *macCoordShortAddress*. All those attributes are requested by the next higher layer after an association was completed.

The association to a given PAN coordinator is not permitted only if the PAN coordinator allows it by the setting the *macAssociationPermit* to TRUE. Similarly, a device should attempt to associate with a PAN that is currently allowing associations. If the *macAssociationPermit* is set to FALSE by the PAN coordinator, then all association request commands will be ignored.

In a beacon-enabled mode, a device may begin tracking the beacon of the coordinator with which it wishes to associate, *a priori*. This is achieved by using the `MLME-SYNC.request` procedure with the *TrackBeacon* parameter set to TRUE.

When an unassociated device completes a passive or an active scan and selects a PAN identifier from a list of PAN descriptors, the higher layer of the unassociated device sends the `MLME-ASSOCIATE.request` primitive to the MAC sub-layer, which will try to associate with the selected PAN. The MAC sub-layer sends a data packet to the corresponding coordinator requesting the association. If the association request is received correctly, the coordinator must send an acknowledgement frame, thus confirming receipt.

However, the acknowledgement to an association request does not mean that the device was associated. In fact, the coordinator needs time to process the request and determine if the current resources available on the PAN are sufficient to allow another device to associate. This processing time must be made within *aResponseWaitTime* symbols.

If the coordinator finds that this device was previously associated, it must remove all previous device-specific information. If sufficient resources are available, the coordinator successfully associates the device by allocating it a new short address and generates an association response command containing the new address and a status indicating a successful association. If the resources are not sufficient to allow a new association, the coordinator generates to the device an association response command with a status indicating a failure. Note that the association response command is sent to the device using *indirect transmission*, i.e. the association command frame is added to the list of *pending transactions* stored on the coordinator and extracted at the discretion of the device.

On the other side, the device, after getting an acknowledgement frame, waits for a time interval of *aResponseWaitTime* symbols. It either checks the beacons in the beacon-enabled network or extracts the association response command from the coordinator after *aResponseWaitTime* symbols.

Upon reception of the association response command, the device sends an acknowledgement to the coordinator. If the association is successful, the device stores the addresses of the coordinator with which it has associated. The short address of the coordinator is stored in *macCoordShortAddress* and the extended address is stored in *macCoordExtendedAddress*.

If the association was not successful, the device sets the *macPANId* to the default value (0xffff).

b. Disassociation

The disassociation process may be initiated by either the coordinator or the device itself. The disassociation command is created by the next higher layer by issuing the `MLME-DISASSOCIATION.request` primitive to the MAC sub-layer.

- *Coordinator-initiated disassociation*: If the coordinator wants to disassociate one of its associated devices, it sends the *disassociation notification* command to the device using indirect transmission, i.e. the disassociation notification command frame is added to the list of *pending transactions* stored on the coordinator and extracted at the discretion of the device. When the device receives the disassociation notification command frame it should send an acknowledgement, thus confirming receipt. Even if the acknowledgement is not received by the coordinator, the coordinator considers that the device is disassociated. All the references to the device are removed by the PAN coordinator.
- *Device-initiated disassociation*: On the other side, if an associated device wants to leave a PAN, it sends a disassociation notification command to the coordinator. Upon reception, the coordinator sends back an acknowledgement, thus confirming receipt. Even if the acknowledgement is not received the device will consider itself as disassociated. All the references to the PAN must be removed by the device.

5.5 *Synchronization*

The standard defines mechanisms to synchronize the coordinators with their associated devices. This procedure is particularly important in beacon-enabled mode where each associated device must synchronize its transmission with the beacon transmissions from its coordinator. Hence, for PANs supporting beacons, synchronization is performed by receiving and decoding the beacon frames. For beaconless PANs, synchronization is performed by polling the coordinator for data.

A synchronization problem arises if a device is not able to receive the beacon from its coordinator, or if *aMaxFrameRetries* attempts of transmissions come to failure. In this case, the device may consider itself as orphan and must retry to either perform the orphaned device realignment procedure or reset the MAC sub-layer and perform association procedure.

In the following, we present the synchronization mechanisms in both PAN modes.

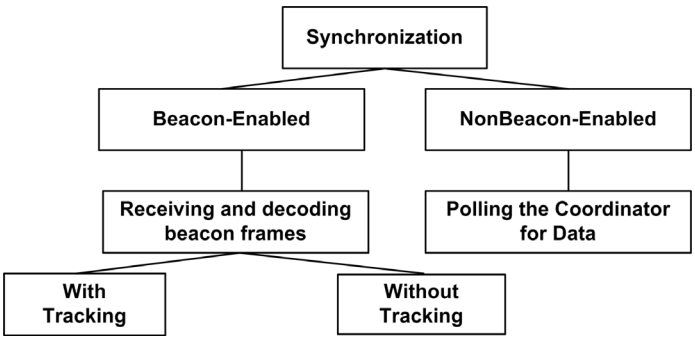

Fig. 13. Synchronization in IEEE 802.15.4

a. Synchronization in beacon-enabled PANs

In beacon-enabled PANs, all associated devices must be able to receive and decode beacon frames from their coordinator and synchronize their transmission. Such synchronization allows the associated device to detect any pending messages or to track the beacon. The device must be permitted to acquire beacon synchronization only with beacons containing the PAN identifier specified in *macPANId*. If the *macPANId* attribute of a given device is set to 0xffff, then the device is not associated and shall not attempt to acquire beacon synchronization.

The beacon synchronization may be of two types:

- *Beacon synchronization with tracking*: beacon synchronization is performed by using the `MLME-SYNC.request` primitive. If the *TrackBeacon* attribute of this primitive is set to TRUE, then the device synchronizes itself with the next beacon and attempts to track all future beacons. If the beacon tracking is activated, the MAC sub-layer must enable the receiver at a time prior to the next expected beacon frame transmission, i.e. just before the start of the superframe.
- *Beacon synchronization without tracking*: in this case, the latter primitive should be send with the *TrackBeacon* attribute set to FALSE. In this case, the device will acquire the beacon only once or terminate the tracking after the next beacon if tracking was enabled through a previous request.

To acquire beacon synchronization, a device must enable its receiver and search for at most [*aBaseSuperframeDuration* * (2^n +1)] symbols, where *n* is the *macBeaconOrder*. This process is repeated until a beacon frame containing the current PAN identifier or the number of missed beacons reaches *aMaxLostBeacons*. In the latter case, the MAC sub-layer notifies the next higher layer by sending an `MLME-SYNC-LOSS.indication` primitive with a loss reason of `BEACON-LOSS`.

When a device receives a beacon frame, it verifies if the beacon frame was sent by the coordinator with which it is associated. Hence, if the source address and the PAN identifier embedded in the beacon frame are not the same as those stored in the device, the beacon frame is discarded.

b. Synchronization in non beacon-enabled PANs

All devices operating in a non beacon-enabled PAN must be able to poll the coordinator for data at the discretion of the next higher layer.

A device is instructed to poll the coordinator when the MAC sub-layer receives the `MLME-POLL.request` primitive and then follow the procedure to extract the pending data from the coordinator.

c. Orphaned device realignment

A device may conclude that it becomes an orphan device (out of the range of its last PAN) if a predetermined number of transmission attempts have failed. A single communication failure occurs when a device transaction fails to reach the coordinator, i.e. an acknowledgement is not received after *aMaxFrameRetries* attempts at sending the data. If the device comes to the conclusion that is has been orphaned, it triggers the *orphaned device realignment* procedure or resets the MAC sub-layer and then perform the association procedure.

The orphaned device realignment consists on doing an orphan channel scan (see Annex 4). If the orphan scan is successful, i.e. the PAN has been located; the device must update its MAC attributes with the PAN information contained in the coordinator realignment command. If the orphan scan comes to failure, the next higher layer must decide what further actions need to be taken (retry the orphan scan or attempts to re-associate).

5.6 *Transmission and Reception of Data*

a. Transmission of data

The transmission of data depends on the operational mode of the PAN. In a beacon-enabled PAN, a device that wants to transmit data must locate the beacon frame of its coordinator and sends its data according to the superframe structure, using slotted CSMA/CA in the CAP or within its allocated GTS. In a non beacon-enabled mode, each device simply uses the unslotted CSMA/CA mechanism to transmit data.

b. Reception

As for the reception, any device may instruct its sub-layer to enable (or not) its receiver during idle periods. Any device may receive transmission from all devices complying with the standard specification and using the same channel, provided that these devices are in its POS. The MAC sub-layer must be able to filter unwanted frames. The filter depends on whether the MAC sub-layer is operating in promiscuous mode or not. In promiscuous mode (*macPromiscuousMode*=TRUE), the MAC sub-layer discards all received corrupted frames, i.e. frames that do not contain a correct value in their FCS field. In a non promiscuous mode, the MAC sub-layer must only accept frames that satisfy all following requirements:

- The frame type subfield of the frame control field must not contain an illegal frame type.

- If the frame type indicates that the frame is a beacon frame, the source PAN identifier must match *macPANId* unless *macPANId* is equal to 0xffff, in which case the beacon frame is accepted regardless of the source PAN identifier.
- If a destination PAN identifier is included in the frame, it must match *macPANId* or must be the broadcast PAN identifier (0xffff).
- If a short destination address is included in the frame, it must match either *macShortAddress* or the broadcast address (0xffff). Otherwise, if an extended destination address is included in the frame, it must match *aExtendedAddress*.
- If only source addressing fields are included in a data or MAC command frame, the frame must be accepted only if the device is a PAN coordinator and the source PAN identifier matches *macPANId*.

If any of the requirements above is not satisfied, the frame must be discarded by the MAC sub-layer, otherwise, it parses it to the next higher layer, if successfully processed.

c. Extracting pending data from a coordinator

This communication mechanism is called *indirect transmission*, where a given device polls pending data from its coordinator.

In a beacon-enabled mode, a device is aware whether it has any frame pending by examining the contents of the received beacon frames. If its address is contained in the *Pending Address* field of the beacon frame, then the device sends a data request command to the coordinator in the CAP. If this request is correctly received by the coordinator, it sends back an acknowledgement to the device, thus confirming receipt. When the device receives the acknowledgement, it enables its receiver for at most *aMaxFrameResponseTime* symbols in the CAP of a beacon-enabled PAN, or *aMaxFrameResponseTime* symbols in a non beacon-enabled PAN to receive the corresponding frame from the coordinator. If any data is pending, the coordinator must send the pending frame, otherwise, it must send a frame containing a zero- length payload, indicating that no data is present. The sending of the pending data is based on CSMA/CA, unless the MAC sub-layer can start transmission of the data frame between *aTurnaroundTime* and *aTurnaroundTime* + *aUnitBackoffPerids* symbols and there is time remaining in the CAP for the transmission of the message, an appropriate IFS and acknowledgement.

If the requesting device does not receive any data from the coordinator within *aMaxFrameResponseTime* CAP symbols in a beacon-enabled PAN or symbols in a non beacon-enabled PAN, it concludes that there are no pending data at the coordinator. If the requesting device does not receive a data frame from the coordinator, it sends an acknowledgement frame, if requested, thus confirming receipt.

5.7 *GTS Allocation and Management*

a. Definition of the GTS and related rules

A GTS (Guaranteed Time Slot) is a portion of the superframe that is dedicated (on the PAN) exclusively to a given device. The GTS allows the corresponding device to access the medium without contention in the CFP. It is a kind of resource reservation in WPANs.

A GTS can only be allocated by the PAN coordinator, and it must be used only for communications between the PAN coordinator and a device. A single GTS may extend over one or more superframe slots. The PAN coordinator may allocate up to seven GTSs at the same time, provided that there is sufficient capacity in the superframe.

The GTS must be allocated by a device before use and can be deallocated at any time at the discretion of the PAN coordinator or the device that originally requested the GTS. A device to whom a GTS has been allocated can also transmit during the CAP.

The PAN coordinator is the responsible for performing GTS management. The PAN must have enough resources to store all required information to manage the seven potential GTSs. For each GTS, the PAN coordinator stores its *starting slot*, *length*, *direction*, and *associated device address*. All these parameters must be embedded in the GTS request command. The GTS direction specifies if the direction of data flow (in that GTS) is from the device to the coordinator (*transmit*) or from the coordinator to the device (*receive*). Each device may request one transmit GTS and/or one receive GTS.

A device is able to allocate and use a GTS only if it is currently tracking the beacons. If synchronization with the PAN coordinator is lost, all its GTS allocations will be lost.

In extreme situations, the PAN coordinator may deallocate one or more GTSs to preserve the minimum CAP length of *aMinCAPLength*.

b. GTS Allocation

A device that wants to allocate a GTS must send a GTS request command to its PAN coordinator indicating the GTS characteristics according to the requirement of the intended application.

The GTS request command frame is presented in Fig 14.

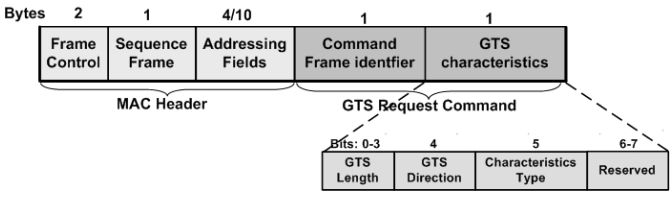

Fig. 14. GTS Request Command Frame

The *GTS Length* contains the number of superframe slots being requested for the GTS. The *GTS Direction* specifies the relative direction of data frame transmissions by the device. If the value is '1', the GTS is allocated for receiving data; else if the value is '0', the GTS is allocated for transmitting data. The *Characteristic Type* represents a

GTS allocation if the value is '1' and a GTS deallocation if the value is '0'.

On receipt of a GTS allocation request, the PAN coordinator sends an acknowledgement frame to confirm the receipt and checks if there are available resources **in the current superframe** based on the remaining length of the CAP and the desired length of the requested GTS.

The resources are considered to be available if the maximum number of GTS has not been reached, and the allocation of the GTS does not reduce the length of the CAP to less than *aMinCAPLength* = 440 Symbols.

The allocation of GTSs is made in a FIFO order by the PAN coordinator provided that sufficient resources are available. The PAN coordinator makes this decision within *aGTSDescPersistenceTime* (= 4) superframes. Hence, the requesting device, after receiving the acknowledgement of the GTS request command, keeps tracks of beacons frames for at most *aGTSDescPersistenceTime* superframes. If no GTS descriptor is associated to the device within the time, the GTS request is considered to have failed.

The result of the GTS request is reported by the coordinator in the beacon frames using a GTS descriptor for each requesting device. If the GTS was successfully allocated the PAN coordinator sets the *start slot* in the GTS descriptor to the superframe slot at which the GTS begins, and the *GTS Length* in the GTS descriptor to the length of the GTS. If the available resources are insufficient for the new requested allocation, the PAN coordinator sets the *start slot* to 0 and the GTS length to the largest GTS length that can currently be supported.

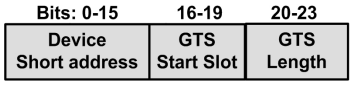

Fig. 15. GTS Descriptor

The PAN coordinator then includes this GTS descriptor in its beacon and update the GTS specification filed of the beacon frame accordingly. The PAN coordinator must also update the final *CAP slot* subfield of the *superframe specification* filed of the beacon frame (see Fig. 18), which indicates the final superframe slot used by the decreased CAP. The GTS descriptor remains in the beacon frame for *aGTSDescPersistenceTime* superframes, after which it should be removed automatically. The PAN coordinator may be allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length due to the inclusion of the GTS descriptor.

c. GTS usage

When the higher layer instruct the MAC sub-layer to send data using the GTS, the MAC sublayer determines if it has a valid GTS that is allocated:

- The PAN coordinator checks if it has *receive* GTS corresponding to the device with the requested destination address.
- If the device is not a PAN coordinator, it determines whether a transmit GTS has been allocated.

If a valid GTS is found, the MAC sublayer must transmit the data during the GTS, i.e., between its starting slot and its starting slot plus its length.

If the device has any receive GTSs, the MAC sublayer of the device must ensure that the receiver is enabled at a time prior to the start of the GTS and for the duration of the GTS, as indicated by its starting slot and its length. The PAN coordinator shall send all frames within a receive GTS with the acknowledgment request subfield of the frame control field set to 1.

Before commencing transmission in a GTS, each device must ensure that the data transmission, the acknowledgment, if requested, and the IFS, suitable to the size of the data frame, can be completed before the end of the GTS.

If a device misses the beacon at the beginning of a superframe, it must not use its GTSs until it receives a subsequent beacon correctly. If a loss of synchronization occurs due to the loss of the beacon, the device considers all of its GTSs deallocated.

d. GTS deallocation

A device is instructed to request the deallocation of an existing GTS through the `MLME-GTS.request` primitive, using the characteristics of the GTS it wishes to deallocate. From this point onward, the GTS to be de-allocated is no longer used by the device, and its stored characteristics are reset.

To request a deallocation of an existing GTS, a device sends a deallocation request to the PAN coordinator. The characteristics type subfield of the GTS characteristics field of the request must be set to 0 (i.e., GTS deallocation), and the length and direction subfields shall be set according to the characteristics of the GTS to deallocate. If the GTS request command is received correctly, the PAN coordinator sends an acknowledgment frame, thus confirming receipt. Upon On receipt of a GTS request command with the characteristics type subfield of the GTS characteristics field set to 0 (GTS deallocation), the PAN coordinator shall attempt to deallocate the GTS. If the GTS

characteristics contained in the GTS request command do not match the characteristics of a known GTS, the PAN coordinator shall ignore the request. The PAN coordinator must also update the final CAP slot subfield of the superframe specification field of the beacon frame, indicating the final superframe slot utilized by the increased CAP. It does not add a descriptor to the beacon frame to describe the deallocation.

When a GTS deallocation is initiated by the PAN coordinator, the MLME shall notify the next higher layer of the change. This notification is achieved when the MLME issues the `MLME-GTS.indication` primitive with a *GTSCharacteristics* parameter containing the characteristics of the deallocated GTS and a characteristics type subfield set to 0. The PAN coordinator must then deallocate the GTS and **add a GTS descriptor into its beacon frame corresponding to the deallocated GTS,** but with its starting slot set to 0. The descriptor remains in

the beacon frame for *aGTSDescPersistenceTime* superframes. The PAN coordinator is allowed to reduce its CAP below *aMinCAPLength* to accommodate the temporary increase in the beacon frame length due to the inclusion of the GTS descriptor. On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress* and a *start slot* equal to 0, the device shall immediately stop using the GTS.

e. GTS reallocation

The deallocation of a GTS may result in the superframe becoming fragmented. Fig. 16 shows an example of possible fragmentation after the deallocation of GTS2.

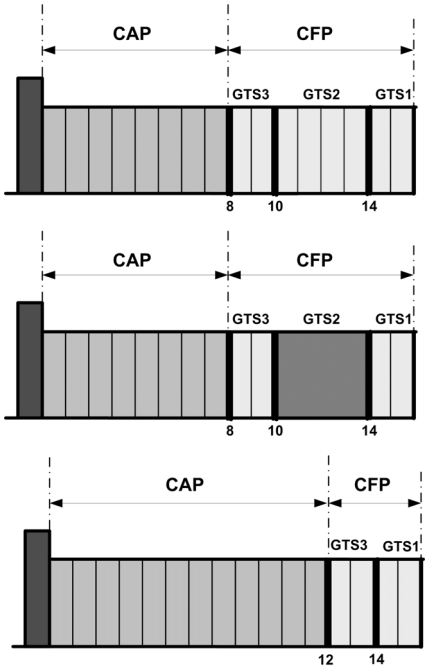

Fig. 16. GTS reallocation - Defragmentation

The PAN coordinator must ensure that any gaps occurring in the CFP, appearing due to the deallocation of a GTS, are removed to maximize the length of the CAP.

When a GTS is deallocated by the PAN coordinator, it adds a GTS descriptor into its beacon frame indicating that the GTS has been deallocated. If the deallocation is initiated by a device, the PAN coordinator does not add a GTS descriptor into its beacon frame to indicate the deallocation. For each device with an allocated GTS having a starting slot lower than the GTS being deallocated, the PAN coordinator must update the GTS with the new starting slot and add a GTS descriptor to its beacon corresponding to this adjusted GTS. The new starting slot is computed so that no space is left between this GTS and either the end of the CFP, if the GTS appears at the end of the CFP, or the start of the next GTS in the CFP.

On receipt of a beacon frame containing a GTS descriptor corresponding to *macShortAddress* and a direction and length corresponding to one of its GTSs, the device adjusts the starting slot of the GTS corresponding to the GTS descriptor and start using it immediately.

f. GTS expiration

The PAN coordinator must attempt to detect when a device has stopped using a GTS using the following rules:

- For a transmit GTS, the PAN coordinator assumes that a device is no longer using its GTS if a data frame is not received from the device in the GTS at least every 2**n* superframes, where *n* is defined below.
- For receive GTSs, the PAN coordinator assumes that a device is no longer using its GTS if an acknowledgment frame is not received from the device at least every 2*n superframes, where n is defined below.

The value of n is defined as follows:

$$n = 2^{(8-macBeaconOrder)} \quad for\ 0 \le macBeaconOrder \le 8$$
$$n = 1 \quad for\ 9 \le macBeaconOrder \le 14$$

6. IEEE 802.15.4/ZIGBEE FOR (LARGE-SCALE) WIRELESS SENSOR NETWORKS

ZigBee/IEEE 802.15.4 is aimed at providing a wireless communication infrastructure for applications requiring minimum power consumption and cost per node and which can cope with limited transmission bandwidth (low data rate). Application domains can range from home/building and industrial automation, remote meter reading, environmental and medical monitoring, PC peripherals, consumer electronics, etc.

With such a wide range of potential application patterns in mind, the ZigBee Alliance has committed to provide a set of application profiles. These profiles are an agreement on messages, message formats and processing actions that enable applications residing on separate devices to send commands, request data and process commands/requests to create an interoperable, distributed application [3]. Interested ZigBee vendors/manufacturers can develop profiles and apply for adoption to the ZigBee Alliance. Currently (June 2005), a profile for Home Controls already exists and profiles for Commercial Building Automation and Industrial Plant Monitoring are under development [4]. Residential Network, Low-Power Sensor Network and Synchronous Star Network profiles are also envisaged for the future.

The physical layer of the IEEE 802.15.4 protocol pretty matches with the requirements of wireless sensor networks. In fact, wireless sensor networks are intended to operate in severe condition and noisy environment, hence, the use of the DSSS spreading technique supported by the standard will mitigate the channel perturbation effects. Also, thanks to the low data rate specification in the standard, the IEEE 802.15.4 physical layer is quite efficient in terms of energy consumption, which is very suitable for wireless sensor networks.

Also the Data Link Layer, namely the Medium Access Control mechanism, seems potentially interesting for wireless sensor network applications. IEEE 802.15.4 robust frame formats, dynamically adaptable synchronization mechanisms and sleeping ratios, adaptable medium/large address space (16/64 bit addresses), best effort/guaranteed message transmission are just some of the features fostering its adequateness for this type of networks.

Nevertheless, the IEEE 802.15.4 protocol, by itself, cannot be considered as a plug-and-play solution for (large-scale) wireless sensor networks applications, since it lacks the layers above the Data Link, namely a Network Layer. It is here that ZigBee plays a major role. In fact, the cluster-tree topology publicized in the IEEE 802.15.4 standard seems potentially interesting for large-scale sensor networks (a large number of nodes scattered over a wide area) since it is scalable. However, a wireless sensor network organized in a cluster-tree topology requires a routing mechanism, due to the need of multi-hop communications (most likely, source and destination nodes are not close enough to communicate directly, due to limited radio coverage). The ZigBee Network Layer fulfils this requirement by providing such a routing mechanism (based on the Ad-hoc On Demand Distance Vector algorithm), as well as network management and security services.

ANNEX 1: FRAME FORMATS

The standard defines four frame formats for MAC frames, beacon frames, the data frames and acknowledgment frames.

MAC FRAME FORMAT

The general frame format is given in Fig. 17.

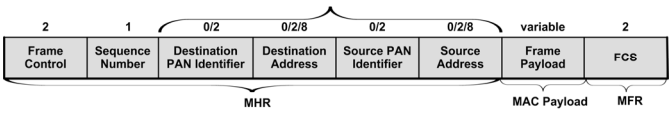

Fig. 17. MAC Frame Format

Each MAC frame consists of the following basic three parts:

- The *MAC Header* (*MHR*), which includes
  - *Frame Control*: it is a 16-bits field and contains information defining the frame type and other control flags (Security Enabled, Frame Pending, Acknowledgment Request, Intra-PAN …).
  - *Sequence Number*: it is an 8-bit field and specifies a unique sequence identifier for the frame.
  - *Destination PAN Identifier*: it is a 16-bit field that specifies the unique PAN identifier of the intended recipient of the frame.
  - *Destination Address*: it is either a 16-bit or 64-bit field (depending on the value of the *destination addressing* subfield of the *Frame Control* field) that specifies the address of the intended recipient of the frame.
  - *Source PAN Identifier*: it is a 16-bit field that specifies the unique PAN identifier of the originator of the frame.
  - *Source Address*: it is either a 16-bit or 64-bit field (depending on the value of the *destination addressing* subfield of the *Frame Control* field) that specifies the address of the originator of the frame.
- The *MAC Payload* of variable length, which contains information specific to individual frame types.
- The *MAC Footer* (*MFR*), which contains the Frame Check Sequence (*FCS*) field. The FCS field is 16 bits in length and contains a 16 bit Cyclic Redundancy Check (*CRC*).

BEACON FRAME FORMAT

The beacon frame format is given in Fig. 18.

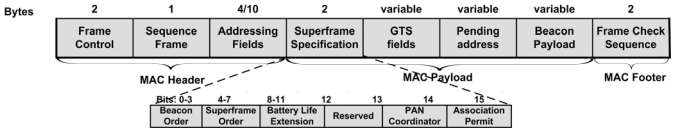

Fig. 18. Beacon Frame Format

The MAC Header and the MAC Footer are the same presented in the general frame format. In the *Frame Control* field, the *Frame Type* subfield must contain the value that indicates a beacon frame, i.e. *0x0*. The *Sequence Number* field contains the current value of *macBSN*, which is a random value within the range $[0x\ 00, 0x\ ff]$.

The MAC payload contains the following information:

- *Superframe Specification* is a 16-bit field that specifies different parameters related to the superframe such as the *Beacon Order*, the *Superframe Order*, the *Final CAP Slot*, the *Battery Life Extension*, the *PAN Coordinator*, the *Association Permit* subfields.
- The *GTS* field has a variable size and contains information on GTSs being allocated such as the *GTS list* and other control flags.
- The *Pending Address* field has a variable length and contains information of the devices that currently have messages pending with the coordinator.
- The *Beacon Payload* field is an optional sequence of up to *aMaxBeaconPayloadLength* bytes specified to be transmitted in the beacon frame by the next higher layer. If *macBeaconPayloadLength* is nonzero, the set of octets contained in *macBeaconPayload* must be copied in this field.

DATA FRAME FORMAT

The data frame format is given in Fig. 19.

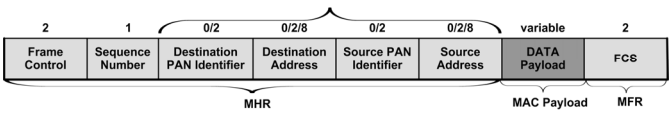

Fig. 19. Data Frame Format

In the *Frame Control* field, the *Frame Type* subfield must contains the value that indicates a data frame, i.e. *0x1*. The *Sequence Number* field contains the current value of *macBSN*.

The *DATA Payload* Field contains the sequence of octets that the next higher layer has requested the MAC sub-layer to transmit.

ACKNOWLEDGMENT FRAME FORMAT

The acknowledgment frame format is given in Fig. 20.

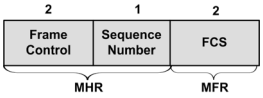

Fig. 20. Acknowledgment Frame Format

The MHR for an acknowledgment frame only contains the *Frame Control* field and the *Sequence Number* field. In the *Frame Control* field, the *Frame Type* subfield must contains the value that indicates an acknowledgment frame, i.e. *0x2*. The *Sequence Number* field contains the value of the sequence number received in the frame for which the acknowledgement is sent.

ANNEX 2: MAC ATTRIBUTES AND CONSTANTS

MAC SUB-LAYER CONSTANTS

This list of MAC sub-layer constants is non exhaustive. We only present the constants used in this technical report. The complete list of constant is available in [1] page 134.

Constant Name	Description	Default value
aExtendedAddress	The 64 bit (IEEE) address assigned to the device.	Device specific
aMaxBE	The maximum value of the backoff exponent in the CSMA/CA algorithm.	5
aMaxBeaconPayloadLength	The maximum size, in octets, of a beacon payload.	52
aMaxFrameResponseTime	The Maximum number of CAP slots in a beacon enabled PAN, or symbols in a non beacon enabled PAN, to wait for a frame intended as a response to a data request frame.	1220
aMaxFrameRetries	The maximum number of retries allowed after a transmission failure.	3
aMaxLostBeacons	The number of consecutive lost beacons that will cause a MAC sublayer of a receiving device to declare a loss of synchronization.	
aMinCAPLength	The minimum number of symbols forming the CAP. This ensures that MAC commands can still be transferred to devices when GTSs are being used. An exception to this minimum shall be allowed for the accommodation of the temporary increase in the beacon frame length needed to perform GTS maintenance	440
aResponseWaitTime	The maximum number of symbols a device shall wait for a response command to be available following a request command.	30720
aTurnaroundTime	RX-to-TX or TX-to-RX maximum turnaround time. This a physical layer constant	12
aUnitBackoffPerids	The number of symbols forming the basic time period used by the CSMA-CA algorithm.	20
aGTSDescPersistenceTime	The number of superframes in which a GTS descriptor exists in the beacon frame of a PAN coordinator.	4

MAC SUB-LAYER ATTRIBUTES

This list of MAC sub-layer attributes is non exhaustive. We only present the attributes used in this technical report. The complete list of constant is available in [1] page 134.

Attribute Name	Description	Default value
macAssociationPermit	Indication of whether a coordinator is currently allowing association. A value of TRUE indicates that association is permitted.	FALSE
macBattLifeExt	Indication of whether battery life extension, by reduction of coordinator receiver operation time during the CAP, is enabled. A value of TRUE indicates that it is enabled.	FALSE
macBeaconOrder	à Specification of how often the coordinator transmits a beacon. The macBeaconOrder, BO, and the beacon interval, BI, are related as follows: for $0 \leq BO \leq 14$, BI = *aBaseSuperframeDuration * 2 BO* symbols. If BO = 15, the coordinator will not transmit a beacon.	15
macBeaconPayload	The contents of the beacon payload.	NULL
macBeaconPayloadLength	The length, in octets, of the beacon payload.	0
macBSN	The sequence number added to the transmitted beacon frame.	Random value from within the range.
macCoordExtendedAddress.	The 64 bit address of the coordinator with which the device is associated.	-
macCoordShortAddress	The 16 bit short address assigned to the coordinator with which the device is associated. A value of 0 x fffe	0xffff

	indicates that the coordinator is only using its 64 bit extended address. A value of 0 x ffff indicates that this value is unknown.	
macDSN	The sequence number added to the transmitted data or MAC command frame.	random
macMaxCSMABackoffs	The maximum number of backoffs the CSMA-CA algorithm will attempt before declaring a channel access failure.	4
macMinBE	The minimum value of the backoff exponent in the CSMA-CA algorithm. Note that if this value is set to 0, collision avoidance is disabled during the first iteration of the algorithm. Also note that for the slotted version of the CSMA/CA algorithm with the battery life extension enabled, the minimum value of the backoff exponent will be the lesser of 2 and the value of *macMinBE*.	3
macPANId	The 16 bit identifier of the PAN on which the device is operating. If this value is 0 x ffff, the device is not associated.	0xffff
macPromiscuousMode	This indicates whether the MAC sublayer is in a promiscuous (receive all) mode. A value of TRUE indicates that the MAC sublayer accepts all frames received from the PHY.	FALSE
macShortAddress	The 16 bit address that the device uses to communicate in the PAN. If the device is a PAN coordinator, this value shall be chosen before a PAN is started. Otherwise, the address is allocated by a coordinator during association. A value of 0xfffe indicates that the device has associated but has not been allocated an address. A value of 0xffff indicates that the device does not have a short address.	0xffff
macSuperframeOrder	This specifies the length of the active portion of the superframe, including the beacon frame. The macSuperframeOrder, SO, and the superframe duration, SD, are related as follows: for $0 \leq SO \leq BO \leq 14$, SD = *aBaseSuperframeDuration * 2 SO* symbols. If SO = 15, the superframe will not be active following the beacon.	15

ANNEX 3: THE INTER FRAME SPACING

The IFS period defines the amount of time that separates the transmission of two consecutive frames. In fact, the MAC sub-layer needs a finite amount of time to process data received by the physical layer. In an acknowledged transmission the IFS follows the acknowledgement frame, otherwise the IFS follows the frame itself.

The length of an IFS frame depends on the frame size. The transmission of short frames, whose sizes are lower than $aMaxSIFSFrameSize = 18\ Bytes$, is followed by a *SIFS* period of a duration of at least $aMinSIFSPeriod = 12\ symbols$. On the other hand, the transmissions of long frame, whose lengths are greater than $aMaxSIFSFrameSize$ is followed by a LIFS of duration of at least $aMinLIFSPeriod = 40\ symbols$. Fig. 21 illustrates these concepts.

The CSMA/CA must take this requirement into account for transmissions in the CAP.

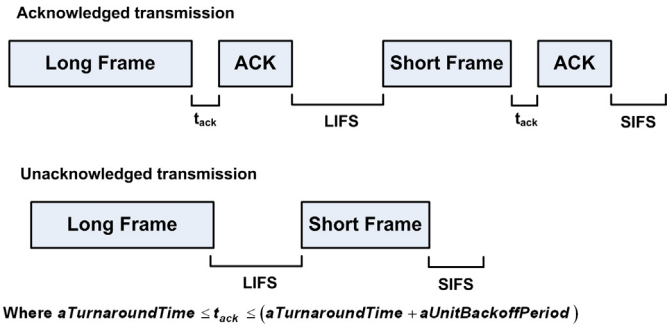

Fig. 21. The Inter Frame Spacing

ANNEX 4: CHANNEL SCANNING OPERATIONS

Channel scanning operations are required to identify existing PANs before association, to start a PAN and to resolve PAN identifier conflicts.

The standard defines four types of channel scans:

- ***ED Channel Scan*:** an ED channel scan allows an FFD to obtain a measure of the peak energy in each requested channel. This could be used by a prospective PAN coordinator to select *an appropriate channel* in which to operate before starting a new PAN. During an ED scan the MAC sub-layer must discard all frames received over the PHY data service.
- ***Active Channel Scan***: an FFD performs an active scan to locate any coordinator transmitting beacon frames within its personal operating range. The active scan is initiated by the device itself by sending a *beacon request* command. This could be used by:
  - A prospective PAN coordinator to select an appropriate PAN identifier prior to starting a new PAN,
  - A device that wants to be associated to PAN.

  During an active scan the MAC sub-layer must discard all frames received over the PHY data service that are not beacon frames.
- ***Passive Channel Scan***: a passive scan, like an active scan, allows a device to locate any

coordinator transmitting beacon frames within its personal operating range. The *beacon request* command, however, is not transmitted. This type of scan could be used by a device prior to association. During an active scan the MAC sub-layer must discard all frames received over the PHY data service that are not beacon frames.

- ***Orphan Channel Scan***: An orphan scan allows a device to attempt to relocate its coordinator following a loss of synchronization. During an orphan scan, the MAC sub-layer shall discard all frames received over the PHY data service that are not coordinator realignment MAC command frames.

The flowchart of the last three channel scan are presented in Fig. 22 , 23, 24.

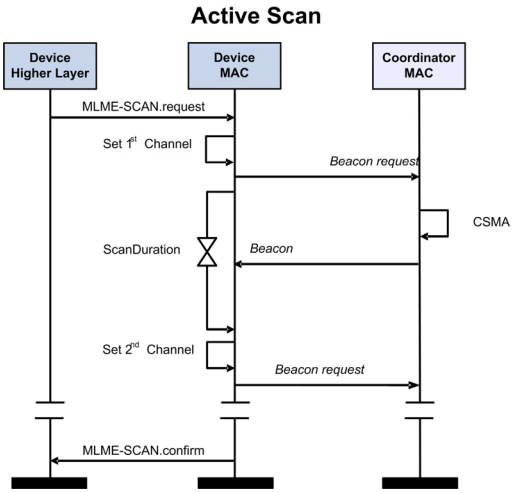

Fig. 22. Active Scan Flowchart

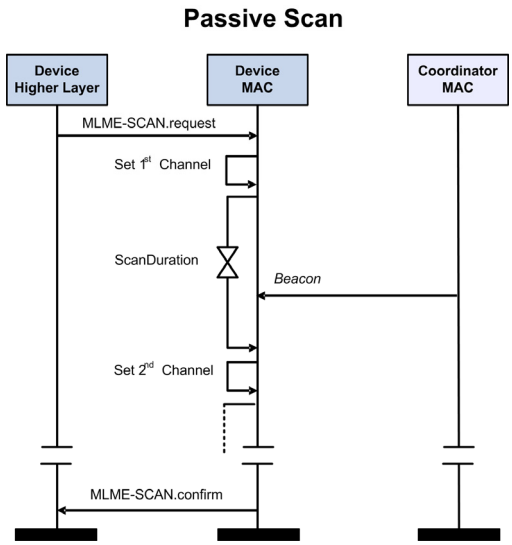

Fig. 23. Passive Scan Flowchart

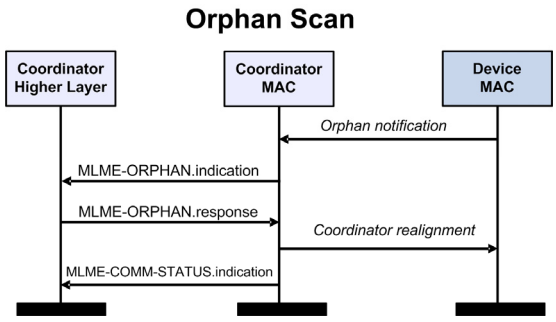

Fig. 24. Orphan Scan Flowchart

ANNEX 5: MAC COMMANDS

The MAC sub-layer of the IEEE 802.15.4 standard defines 9 commands summarized in the table below.

Command Frame Identifier	Command Name	RFD	
		TX	RX
0 x 01	Association Request	X	
0 x 02	Association Response		X
0 x 03	Disassociation Notification	X	X
0 x 04	Data Request	X	
0 x 05	PAN ID Conflict Notification	X	
0 x 06	Orphan Notification	X	
0 x 07	Beacon Request		
0 x 08	Coordinator Realignment		X
0 x 09	GTS Request		
0 x 0a - 0 x ff	Reserved		

The MAC command frame format is presented in Fig. 25.

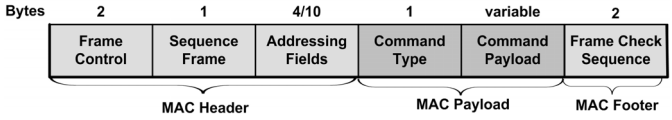

Fig. 25. MAC Command Frame Format

The *Command Type* field must contain the value that indicates the appropriate command type.

Note that these commands are handled by the *MAC sub-layer Management Entity* (MLME) or the *MAC Data Services accessed through the MAC Common Part Sub-layer* (MCPS). The MLME and the MCPS may generate commands by means of its primitives. For instance, the MLME is responsible of the association procedure and a device may request an association to PAN using the `MLME-ASSOCIATE.request` command generated by a device higher layer to the MAC sub-layer. A confirmation of an association when it is successful is sent by the MAC layer back to the higher layer using the `MLME-ASSOCIATE.confirm` command. The DATA Request command is handled by the *MAC Data Services* (`MCPS-DATA.request and MCPS-DATA.confirm`).

7. REFERENCES

[1] "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)", IEEE-SA Standards Board

[2] IEEE 802.15 WPAN™ Task Group 4b (TG4b), http://grouper.ieee.org/groups/802/15/pub/TG4b.html

[3] ZigBee Alliance, ZigBee Specification, December 2004. Available for download at: http://www.zigbee.org

[4] Z. Smith (Ember Corporation), De-mystifying the Building of ZigBee Applications, Sensors Expo & Conference Presentations, June 2005. Available for download at: http://www.zigbee.org/en/events/sensors_expo_conference_2005_06_06.asp

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocols for wireless microsensor networks,” in Proc. of the Hawaii International Conference on Systems Sciences, Jan. 2000.

TABLE OF CONTENTS