Technical Report

Implementation Details of the Time Division Beacon Scheduling Approach for ZigBee Cluster-Tree Networks

André CUNHA

Mário ALVES

Anis KOUBAA

TR-070102

Version: 1.0

Date: 20-07-2007

Implementation Details of the Time Division Beacon Scheduling Approach for ZigBee Cluster-Tree Networks

André CUNHA, Mário ALVES, Anis KOUBAA

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: arec@isep.ipp.pt, mjf@isep.ipp.pt, akoubaa@dei.isep.ipp.pt

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

Abstract

This technical report describes the implementation details of the Time Division Beacon Scheduling Approach in IEEE 802.15.4/ZigBee Cluster-Tree Networks. In this technical report we describe the implementation details, focusing on some aspects of the ZigBee Network Layer and the Time Division Beacon Scheduling mechanism. This report demonstrates the feasibility of our approach based on the evaluation of the experimental results. We also present an overview of the ZigBee address and tree-routing scheme.

Contents

Figures

1 Introduction

This technical report describes the implementation details of the Time Division Beacon Scheduling Approach in IEEE 802.15.4/ZigBee Cluster-Tree Networks [1,2]. In this technical report we describe the implementation details, focusing on some aspects of the ZigBee Network Layer and the Time Division Beacon Scheduling. This report demonstrates the feasibility of our approach based on the evaluation of the experimental results.

This technical report gives a brief overview of the ZigBee Network Layer mechanism, namely the association and addressing schemes, the tree-routing and some features of the Network Layer Information Base.

We also provide a short description of our nesC/TinyOS [3,4] implementation (available at [5]) that supported the Time Division Beacon Scheduling Approach. This implementation was developed for the Crossbow MICAz mote[6].

2 Overview of the IEEE 802.15.4/ZigBee Address Assignment and Tree Routing mechanisms

2.1 Introduction

In ZigBee Networks there are 3 types of devices:

- ZigBee Coordinator – ZC
  - One for each ZigBee Network;
  - Initiates and configures the Network formation;
  - Acts as an IEEE 802.15.4 Personal Area Netork (PAN) coordinator;
  - Acts as router once the network is formed;
  - Is a Full Functional Device (FFD) – Implements the full protocol stack.
- ZigBee Router – ZR
  - Optional network component;
  - Associates with ZC or with previously associated ZR;
  - Acts as an IEEE 802.15.4 PAN coordinator;
  - Participates in multi-hop routing of messages.
  - Is a Full Functional Device (FFD) – Implements the full protocol stack.
- ZigBee End Device –ZED
  - Optional network component;
  - Shall not allow association;
  - Shall not participate in routing;
  - Can be a Reduced Function Device (RFD) – Implements a reduced subset of the protocol stack.

Throughout this document the names of the devices and the acronyms are used interchangeably.

The tree-routing relies on a distributed address assignment mechanism that provides to each potential parent (ZC and ZRs) a finite sub-block of unique network addresses based on the maximum number of children, depth and the number of routers in the PAN.

For setting up a Cluster-Tree Wireless Sensor Network (WSN), 3 Network parameters must be defined at the ZC. The addressing and tree routing mechanisms will operate according to these parameters, outlined next:

- the maximum number of children (*Cm*) of a ZR;
- the maximum number of child routers (*Rm*) of a ZR;
- the depth of the network (*Lm*).

2.2 Association Mechanism

The association procedure takes place when a node (ZR or ZED) device wants to join the network. To proceed with the association the device must scan all the channels for radio transmissions, so that it can select the most suitable PAN. The association is necessary if the device wants to transmit data in the PAN. As a result of the association mechanism, the device is assigned with a short address allowing it to transmit in the PAN.

The device associates with the PAN Coordinator defining its characteristics in the capability information field of the association request command. Figure 1 represents the association request frame command format.

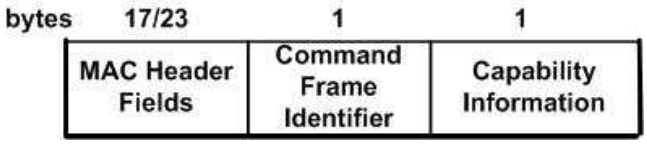


Figure 1 – Association request command format [1]

Besides the standard MAC frame fields, the frame has an addressing field, a command frame identifier field and the capability information of the device.

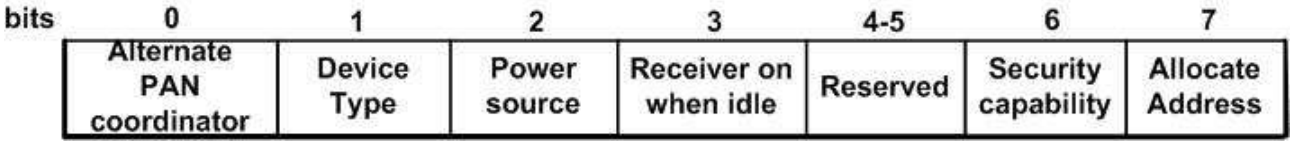


Figure 2 – Association request capability information field format [1]

The capability information field contains the following information:

- Alternate PAN coordinator – 1 if the device is capable of becoming a PAN coordinator (assuming that the device can be a router);
- Device type – 1 – FFD; 0 –RFD;
- Power source – 1 if the device is receiving power from the alternating current mains;
- Receiver on when idle – 1 if the receiver is on during the inactive period;
- Security – 1 if the device is capable of sending and receiving secured MAC frames with a security suite;
- Allocate address – 1 – if the device wants a short address; 0 if the device wants to communicate with the 64 bits extended address.

Upon the reception of the association request command frame, the ZC will process the command and signal the network layer using the *MLME_ASSOCIATE.indication* primitive. The network layer will process the request, by allowing/disallowing the association and assigning a short address (according to the address assignment functions - refer to section 2.2), and issues the MAC layer with the *MLME_ASSOCIATE.request* primitive stating the result of the request. The command is stored in the indirect transmission buffer being transmitted after a data request from the associating device.

Figure 3 represents the association response command format, received by the device requesting the association.

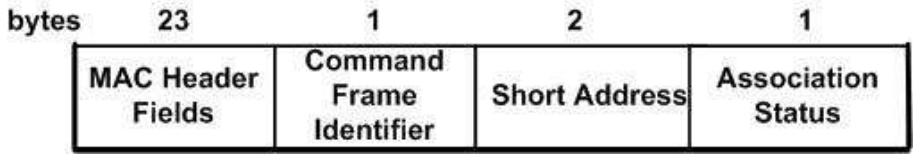


Figure 3 – Association response command format [1]

Besides the standard MAC frame fields, the association response command frame contains the assigned short address (the possible values are represented in figure 4) and the association status stating the reason of success or failure of the association procedure.

Value of *macShortAddress*	Description
0 x 0000–0 x fffd	The device shall use short addressing mode.
0 x fffe	The device shall use 64 bit extended addressing mode with an address consisting of *aExtendedAddress*.
0 x ffff	The device is not associated and shall not communicate on the PAN.

Figure 4 – Association request short address values [1]

Note that the ZigBee Coordinator short address is always 0x0000.

After a successful association there is an update of the MAC PAN Information Base (PIB), the MAC layer of the device stores the following association parameters:

- *a_LogicalChannel* – Logical channel of the PAN;
- *a_CoordAddrMode* – Coordination address mode;
- *a_CoordPANId* – Coordinator PAN id;
- *a_CoordAddress* – Coodinator address, depending on the address mode;
- *a_CapabilityInformation* – Capability information of the device when the association request was sent.
- *a_securityenable* – Security enable stating if the device is using security or not.

In the network layer, each device maintains a neighbour table with the information on every device within its transmission range. In our implementation each device neighbour table contains the following information:

- PAN_Id – PAN 16 bits short address
- Extended Address – Device 64 bits extended address (if possible);
- Network Address – Device 16 bits short address;
- Device Type – Device Type (Coordinator; Router; End Device);
- Relationship – Relation between the neighbour and the current device (Parent; Child; Sibling; Other);
- Depth – Optional Field – Depth of the device in the network;
- Permit Joining – Information about the device availability to accept associations;
- Logical Channel – Device Logical Channel;
- Potential Parent – Indication of whether the device has been ruled out as a potential parent due to a failed join attempt.

2.3 Short Address Assignment

A parent device uses the *Cm*, *Rm*, and *Lm* values to compute a Cskip function defining the size of the address sub-block that is distributed by each parent depending on its depth (d) in the network. For a given network depth *d*, Cskip(d) is calculated as follows:

$$Cskip(d) = \begin{cases} 1 + Cm \cdot (Lm - d - 1), & \text{if } Rm = 1 \\ \dfrac{1 + Cm - Rm - Cm \cdot Rm^{Lm-d-1}}{1 - Rm}, & \text{Otherwise} \end{cases}$$

A parent device that has a Cskip(d) value of zero will not be capable of accepting children and shall be treated as an end device.

A parent device that has a Cskip(d) value greater that zero shall accept devices and shall assign addresses if possible.

A parent device will assign an address that is one grater than its own to the first router that associated. The next associated router will receive an address that will be separated according to the return value of the Cskip(parent depth) function. The maximum number of associated routers is defined in the network parameter nwkMaxRouters (Rm).

Considering a parent node with a depth d and an address of Aparent, the number of child devices n is between 1 and Cm-Rm.

$$1 \leq n \leq (Cm - Rm)$$

The Achild address of the n^{th} child router is calculated according to(n is the number of child routers):

$$Achild = Aparent + (n-1) \times Cskip(d) + 1 \quad , n = 1$$

$$Achild = Aparent + (n-1) \times Cskip(d) \quad , n > 1$$

The Achild address of the n^{th} cild end device is calculated according to(n is the number of child end devices):

$$Achild = Aparent + Rm \times Cskip(d) + n$$

The next figure represents an example of an address assignment scheme. Note that the network parameters are the following:

- Maximum depth: 3
- Maximum children: 6
- Maximum routers: 4

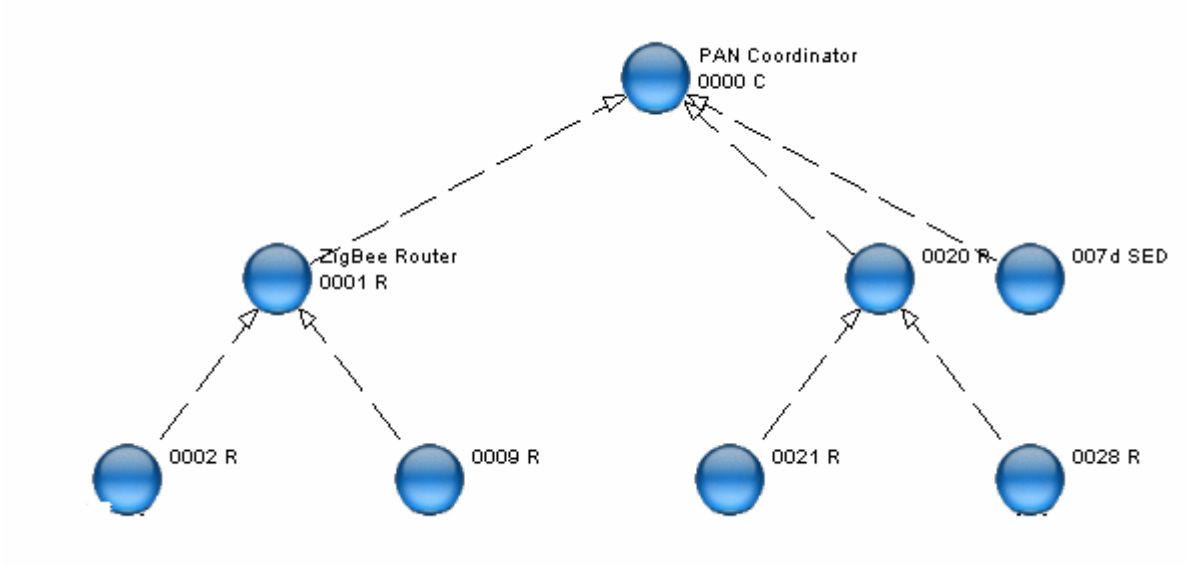


Figure 5 – Address assignment scheme example.

Figure 6 represents the PAN Coordinator available addressing scheme. With the above network parameters the coordinator is allowed to associate 4 routers and 2 end devices in its available address pool.

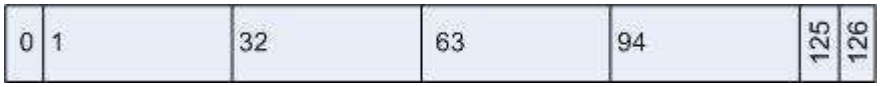


Figure 6 – PAN Coordinator addressing scheme (decimal values).

2.4 ZigBee Tree Routing Mechanisms

Our current implementation only supports the tree-routing (mesh routing is not supported yet). This routing mechanism is based on the addressing scheme of the network.

Each device, upon the reception of a data frame, reads the routing information fields (Figure 7) and checks the destination address.

Octets: 2	2	2	1	1	Variable
Frame Control	Destination Address	Source Address	Radius	Sequence Number	Frame Payload
	Routing Fields				
NWK Header					NWK Payload

Figure 7 – Network layer frame format. [2]

If the destination address is equal to its own address, the device will signal the upper layer with the NLDE_DATA.indication primitive along with the frame payload as argument. If the destination is a child of the device (neighbour table check), the device relays the packet to the appropriate child address. If the

destination address is not a child, the device must check if the address is a descendent using the following condition, being A the device network address, D the destination address and d the device depth in the network.

$$A < D < A + Cskip(d-1)$$

The device address of the next hop when route down is given by:

$$N = A + 1 + \left\lfloor \frac{D - (A+1)}{Cskip(d)} \right\rfloor \times Cskip(d)$$

If the destination address is not a descendant, the device will relay the packet to its parent.

Consider Figure 5 and a network with the following parameters: a maximum depth 3; children 6; routers 4. The Cskip values in the network are presented next:

Depth	Cskip(Depth)
0	31
1	7
2	1

If the ZR 0x0002 wants to transmit a message to ZR 0x0028 the tree-routing protocol will behave as follows:

1. ZR 0x0002 creates the data frame and sends it to it parent (0x0001). The most relevant fields of the data frame are outlined next:
    a. MAC destination address – 0x0001;
    b. MAC source address – 0x0002;
    c. Network Layer Routing Destination Address – 0x0028;
    d. Network Layer Routing Source Address – 0x0002;

2. ZR 0x0001 receives the data frame and realizes that the message in not for him and has to be relayed. The device checks its neighbour table for the routing destination address trying to find the destination is one of its child devices. Then, the device checks if the routing destination address is a descendant by verifying the condition A < D < A + Cskip(d-1) that will result in:

$$0x0001 < 0x0028 < 0x0001 + 7$$

Note that the ZR 0x0001 is a depth 1 device in the network. After verifying that the destination is not a descendant, the ZR 0x0001 will route the data frame to it parent, the ZC 0x0000.

The most relevant fields of the data frame are outlined next:

a. MAC destination address – 0x0000;
b. MAC source address – 0x0001;
c. Network Layer Routing Destination Address – 0x0028;
d. Network Layer Routing Source Address – 0x0002;

3. The ZC 0x0000 receives the data frame and will verify if the routing destination address exists in its neighbour table. After realizing that the destination device is not a neighbour the ZC, that is on top of the tree and cannot route up, the next hop address is calculated as follows:

$$N = 0x0000 + 1 + \left\lfloor \frac{0x0028 - (0x0000 + 1)}{31} \right\rfloor \times 31$$

The next hop address results in N = 32 (decimal) = 0x0020.

The most relevant fields of the data frame are outlined next:

a. MAC destination address – 0x0020;
b. MAC source address – 0x0000;
c. Network Layer Routing Destination Address – 0x0028;
d. Network Layer Routing Source Address – 0x0002;

4. The ZR 0x0020 receives the data frame and checks its neighbour table for the routing destination address. After verifying that the address is its neighbour, the message is routed to it. The next hop is assigned with the short address present in the selected neighbour table entry.

   The most relevant fields of the data frame are outlined next:

   a. MAC destination address – 0x0028;
   b. MAC source address – 0x0020;
   c. Network Layer Routing Destination Address – 0x0028;
   d. Network Layer Routing Source Address – 0x0002;

3 IEEE 802.15.4/ZigBee Implementation Architecture

3.1 Architecture and File Structure

The Time Division Beacon Scheduling was implemented in TinyOS/nesC[3,4] using our own implementation of the IEEE 802.15.4 protocol [5]. We had to develop some functionalities of the ZigBee standard [2], namely, the network layer with association/disassociation support, tree-routing mechanisms, neighbour tables and some of the Service Access Point (SAP) functionalities. The next figure represents the network layer reference model.

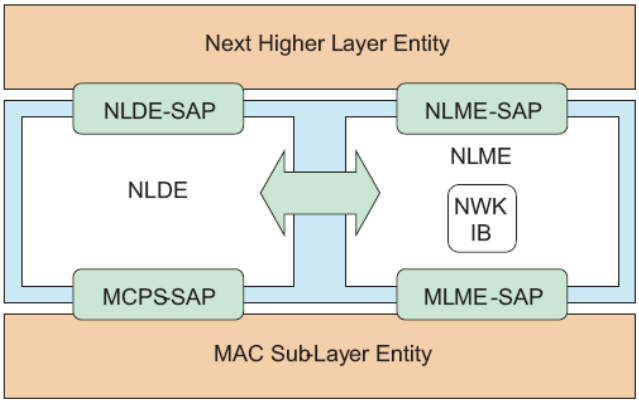


Figure 8 – Network layer reference model

The network layer provides two service entities. The Network Layer Data Entity (NLDE) provides a data service for allowing the transmission of data frames and topology specific routing. The Network Layer Management Entity (NLME) provides a management service allowing the application interface layer to interact with the network layer stack parameter. The management services provided are the following:

- Configuring a new device – Start the device operation as a ZigBee Coordinator/Router/End Device;
- Starting a Network – Establish a new network with the desired parameters;
- Joining and leaving a network – Association/disassociation procedures;
- Addressing – The ability for Coordinator or Routers to correctly assign addresses;
- Neighbour discovery – Maintenance of a neighbour table of all the devices one-hop away.
- Reception Control – Control the MAC layer operation mode for data reception.
- Route discovery – The ability to store a routing table. (Out of the focus of our implementation)

Figures 9 and 10 represent the architecture of our IEEE 802.15.4 implementation of the PHY and MAC layer and the ZigBee implementation of the NWK layer.

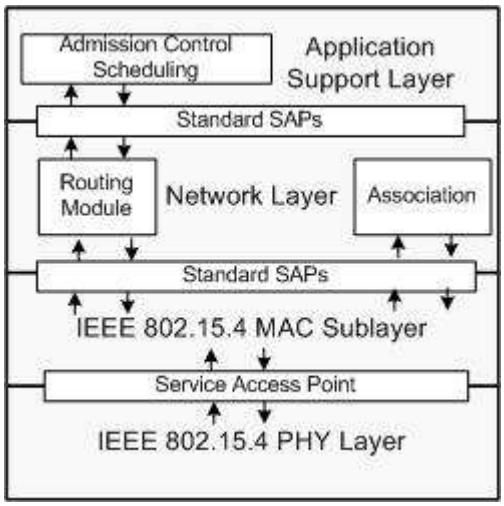


Figure 9 – Implementation Architecture

Figure 10 illustrates the TinyOS implementation diagram, respecting the layered structure presented in Figure 9. The Physical, Data Link and Network layers (gray modules in Figure 10) are implemented by us. The hardware drivers of the CC2420 radio transceiver are already provided by TinyOS.

The admission control for the Time Division Beacon Scheduling is implemented in the Application Support Layer. This layer provides an interface to the Application Layer to access and manage the stack.

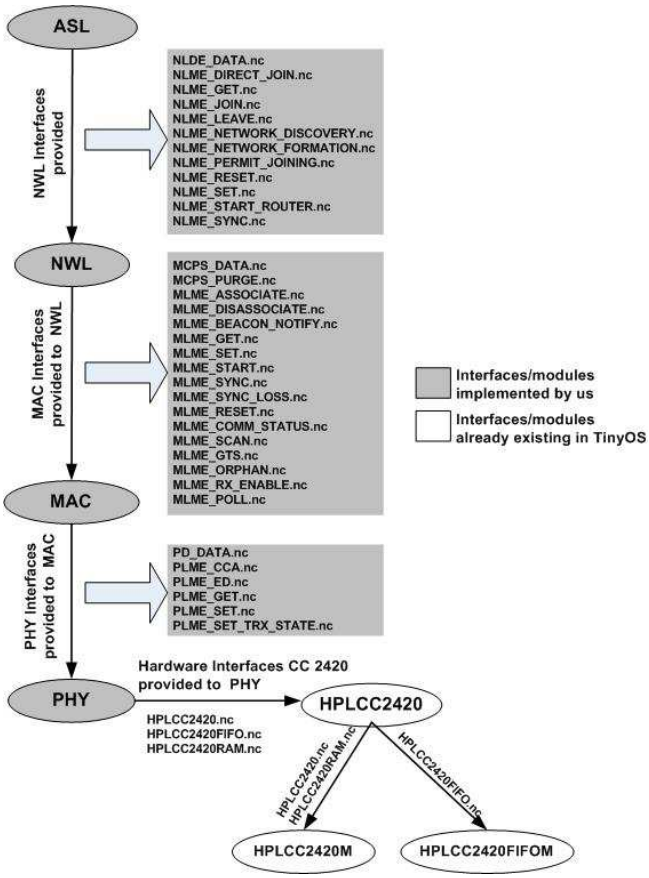


Figure 10 – TinyOS implementation file structure

3.2 Network Layer Implementation Overview

The network layer was developed as a TinyOS component (NWK) and each SAP as an interface. The MAC component provides the NWK the MAC SAP interfaces and the NWK provides the upper layer with the NWK SAP interfaces. Figure 11 represents the component diagram that connects the NWK module with the MAC module (this diagram was generated by the nesdoc tool provided by TinyOS [4]).

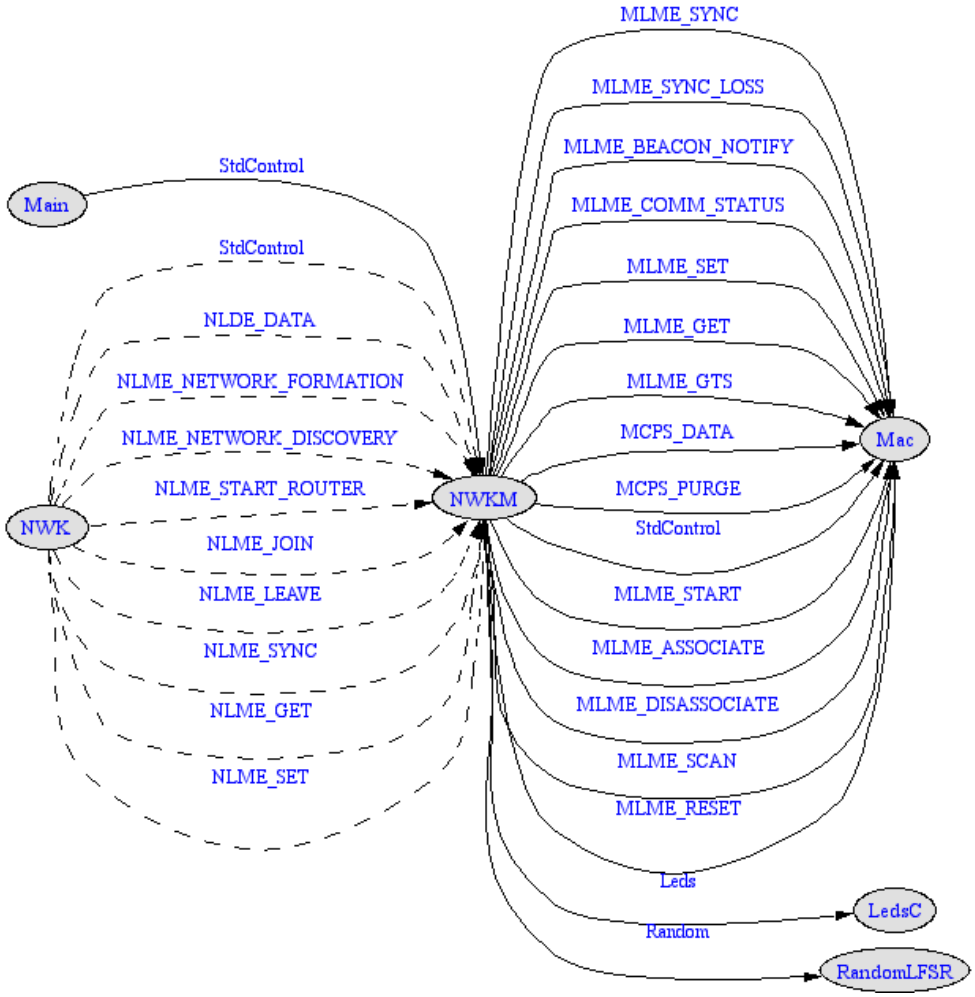


Figure 11 – TinyOS NWK component diagram

In order to implement the Time Division Beacon Scheduling, only a subset of the network layer was developed. The functionalities implemented are the following:

- Network association mechanisms – Tree association scheme;
- Neighbour table – Information about the parent node and the associated child devices only;
- NWK IB – Network layer information base;
- Tree routing.

The network discovery functions were implemented statically because, in our current implementation of the IEEE 802.15.4, the channel scan mechanism is not implemented. The network parameters were defined as constants.

3.3 Time Division Beacon Scheduling Implementation Details

The Time Division Beacon Scheduling Mechanism does not introduce relevant changes in the protocol specification. It relies on a negotiation mechanism based on command frames embedded in data frames.

When implementing this mechanism we assume the following:

1. The ZigBee network layer supports the tree-routing mechanism and the network addresses, of the devices, are assigned accordingly.
2. The ZigBee Coordinator is the first node broadcasting beacons in the network.
3. The ZigBee routers start to send beacons only after a successful negotiation.
4. The same BI is used by every router.

For the negotiation of the beacon transmission, each ZR must complete the following steps:

1. The ZR must successful associate with its parent and temporarily behaves as a ZigBee end device, without sending beacons.
2. The ZR initiates the negotiation protocol by sending a “START SENDING BEACON” request command.
3. The ZC receives the request and determines the schedule of the ZR.
4. After the schedule process, the ZC replies by sending a “START SENDING BEACON” response command with the status of the negotiation (SUCCESS or FAIL) and the transmission offset value to the requesting ZR
5. The ZR receives the command from the ZC and if the negotiation its successful it starts sending beacon in the defined offset related to its parent.

In order to implement this mechanism, some changes are needed in the Network and Mac layer Service Access Point (SAP) primitives. Is necessary to add a *StartTime* argument in the MLME-START.request primitive, as already proposed in the ZigBee standard [2, pag 245]. This primitive is used by the upper layer to request the MAC layer to start sending beacons or use a new superframe configuration. The new format of the primitive is as follows:

MLME-START.request (
PANID,
LogicalChannel,
BeaconOrder,
SuperframeOrder,
PANCoordinator,
BatteryLifeExtention,
CoordRealignment,
SecurityEnable,
StartTime)

The *StartTime* parameter will be used as a transmission offset referring to the ZigBee Router (ZR) parent. In the ZC the value of this parameter is 0.

The *StartTime* parameter size is 3 bytes and is specified in symbols.

In the NLME-START-ROUTER.request primitive there is also the need to add a *StartTime* parameter. The new format of the primitive is as follows:

NLME-START-ROUTER.request(
BeaconOrder,
SuperframeOrder,
BatteryLifeExtension,
StartTime)

The primitive is requested by the Network upper layer of the ZR to start the beacon transmission. The StartTime parameter is obtained after a successful negotiation with the ZC for beacon broadcasting. Note that in case of an unsuccessful negotiation the ZR will not be allowed to send beacons, therefore can only act as a ZED.

After a successful negotiation of the beacon transmission, the ZR will have two active periods, its own (the superframe duration) and the parent superframe duration. In its own active period the ZR is allowed to transmit frames to its associated devices or relay frames to the descendant devices in the tree. The frames destined to its parent are sent during the parents active period. To accomplish this behaviour there is a need to implement a different buffer mechanism for each message flow - the downstream to the device descendants and the upstream to the device ascendants.

The buffer mechanism is implemented in the MAC layer that uses the downstream buffer or the upstream buffer depending of the transmission options parameter of the MCPS_DATA.request primitive. The transmit options or *TxOptions* parameter, last argument of the primitive, define the transmission options for the data frame, allowing the frame to be send in the GTS or during the CAP period using the CSMA/CA or like an indirect transmission and with/without an acknowledgment request. This parameter will also define if the transmission will use the upstream (sent in the parent superframe) or the downstream buffer (sent in the devices superframe).

As defined in the IEEE 802.15.4 standard [1] the transmission options parameters has the following format:

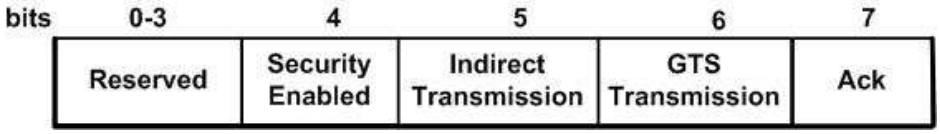


Figure 12 – MCPS_DATA.request TxOptions format [1]

In order to inform the MAC layer of which buffer to use, we have changed the transmission format including an upstream parameter. The Time Division Beacon Scheduling transmission option parameter has the following format:

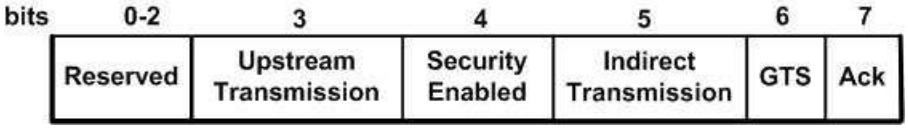


Figure 13 – Time Division Beacon Scheduling MCPS_DATA.request TxOptions format

During the ZR superframe all the frames that need to be transmitted to the parent will be stored in the upstream buffer. When the device enters the parent superframe, it tries to transmit the messages using the CSMA/CA.

To enable the use of the 2 message buffers the device must wake up in the parents superframe. This is accomplished by adding new two timer events to the MAC layer. One that is triggered in the beginning of the parent superframe and turns on the transceiver in receive mode and another in the end turning the transceiver off.

The maintenance of the devices synchronization is the major challenge in this implementation. The devices must be always synchronized with their respective parents. A first difficulty in the implementation of the beacon-enabled mode was related to the TinyOS management of hardware timer provided by the MICAz motes, which does not allow to have the exact values of the beacon interval, superframe, time slot and backoff durations as specified by the IEEE 802.15.4 standard. To accomplish a precise synchronization, a timer component was developed, with an asynchronous behaviour regarding the code execution, based on the hardware clock.

There are two different types of timers in this implementation. The synchronous timers are used in the implementation for events that don't need accuracy and the asynchronous timers that are more precise due to their asynchronous behaviour.

The clock tick granularity of the MICAz mote that best fits our requirements is equal to 69.54 microseconds, which approximately corresponds to four symbols. In fact, the four symbols duration has a theoretical value of 64 microseconds which leads to a cumulative effect on the discrepancy with theoretically values of beacon interval, superframe durations and time slot durations in millisecond for high superframe and beacon orders. For instance, the beacon interval (BI) of 8 is equal to 245760 symbols, which theoretically corresponds to 3932.160 ms, but experimentally corresponds to 4266.588 ms, based on the MICAz clock granularity. This discrepancy, however, does not impact the correct behaviour of the implemented protocol. Annex A contains information comparing the theoretical and experimental values of the beacon interval. Since we are using the same mote platform for each node we experience a coherent network behaviour.

In our current implementation we did not include the implementation of the scheduling algorithm yet. Instead, the application running on the ZC has the offset values pre-established for each node address that request a negotiation for a time window slot.

The Time Division Beacon Scheduling was implemented using our own implementation of the IEEE 802.15.4/ZigBee protocol stack using nesC/TinyOS for the MICAz motes. The implementation is available in [5].

3.4 Negotiation Protocol

The negotiation of the beacon transmission is a simple protocol that uses the data frames payload with a predefined format payload. The format of the negotiation fields is the following:

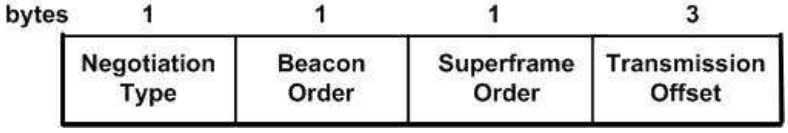


Figure 14 – Negotiation Fields.

- Negotiation type – Indicates the type of the negotiation command. This field can have the following values; 1 for a negotiation request, 2 for a negotiation accept and 3 for a negotiation deny;
- Beacon Order – Indicates the beacon order of the ZR device requesting the negotiation;
- Superframe Order – Indicates the superframe order of the ZR device requesting the negotiation;
- Transmission Offset – Indicates the transmission offset schedule by the ZC in a negotiation accept command.

The next diagram represents the sequence of Network layer events from the association of the ZR (A) until the beacon transmission after a successful negotiation (B).

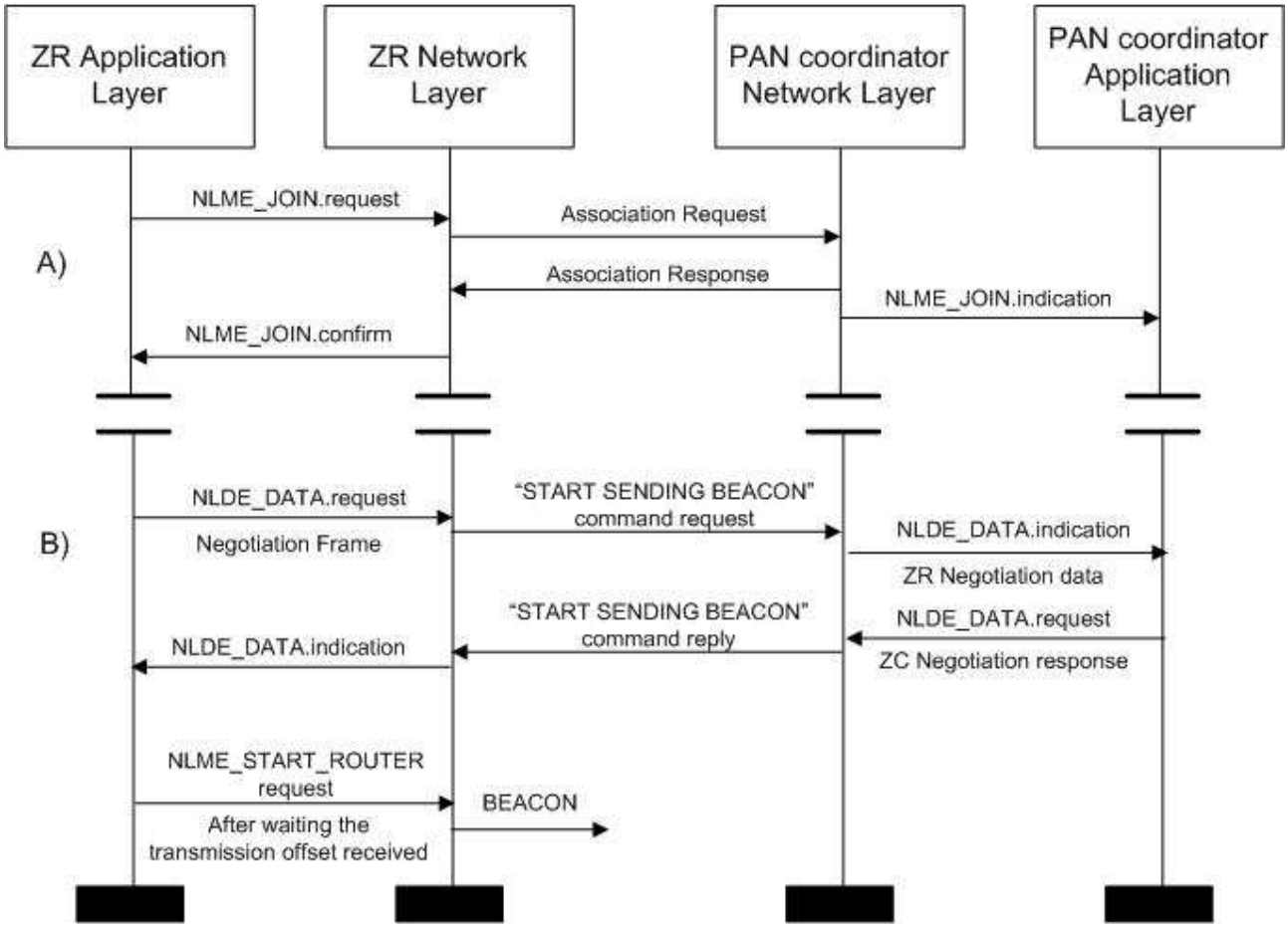


Figure 15 – Negotiation diagram

4 Experimental Results

4.1 Experimental Platform

This implementation was developed for TinyOS version 1.1.15 under the Cygwin for Windows XP environment.

The application used to write code was the Programmers Notepad 2 that provides code highlighting.

The hardware used was the Crossbow MICAz motes. These "IEEE 802.15.4-compliant" motes operate in the 2.4 GHz ISM band and have a 16 Mhz Atmel ATMega128L microcontroller [7] (with 128 kB of program Flash) and a Chipcon CC2420 802.15.4 radio transceiver [8] (allowing a 250 kbps data rate).

The MIB510 programming board was used to program the motes. This programmer can upload the applications to motes through the COM port and allow a debug mechanism by sending data through the COM port and reading it in a COM port software listener, like the ListenRaw (found in the TinyOS distribution) or the Windows HiperTerminal. This debug mechanism raises a problem concerning the hardware operation because the relaying of data through the COM port blocks all the other mote operations, while this data is being sent. This can cause synchronization problems. We also use the MIB600 programmer to program the motes through the IP network.

In order to overcome the COM debug problems we use a packet sniffer to show the packets transmitted providing a good debug mechanism by transmitting debug data in the packets payloads.

We have two different packet sniffer applications. The first is an IEEE 802.15.4/ZigBee packet sniffer provided by Chipcon the CC2420 Packet Sniffer for IEEE 802.15.4 v1.0 [9] that provides a raw list of the packets transmitted. This application works in conjunction with a CC2400EB board and a CC2420 radio transceiver. We also use the Daintree IEEE 802.15.4/ZigBee Network/Protocol Analyser [10] that provides more functionalities like the network topology and some network analysis parameters.

The outputs shown in the next section were provided by these packet sniffers.

4.2 Network Scenario

We have conducted 2 different experiments in order to evaluate the Time Division Beacon Scheduling feasibility with different network depths.

In the first experiment we want to evaluate the network behaviour taking into account the following parameters.

- 13 ZigBee routers (Including the PAN Coordinator);
- Beacon Interval (BI): 8 (245760 symbols; 4266885 us;);
  - 16 Window Time Slots with the duration of 15360 symbols (266680 us);
  - Maximum superframe duration of 4 (15360 symbols);
- Superframe Duration (SO): 4 (15360 symbols; 266680 us;);
- Addressing Information:
  - Maximum depth: 3
  - Maximum children: 6
  - Maximum routers: 4

The network topology is the following:

- PAN Coordinator at depth 0
- 4 ZigBee Routers at depth 1
- 2 ZigBee Routers at depth 2

In the second experiment we want to evaluate the network behaviour taking into account the following parameters.

- 15 ZigBee routers (Including the PAN Coordinator);
- Beacon Interval (BI): 8 (245760 symbols; 4266885 us;);
  - 16 Window Time Slots with the duration of 15360 symbols (266680 us);
  - Maximum superframe duration of 4 (15360 symbols);
- Superframe Duration (SO): 3 (7680 symbols; 133340 us;);
- Addressing Information:
  - Maximum depth: 3
  - Maximum children: 6
  - Maximum routers: 4

The network topology is the following:

- PAN Coordinator at depth 0
- 2 ZigBee Routers at depth 1
- 4 ZigBee Routers at depth 2
- 8 ZigBee Routers at depth 3

4.3 Experiment 1

In this experiment we want to evaluate the network behaviour with depth 2 ZR. Figures 17 and 18 represent the network topology after the association and negotiation mechanism.

The devices operate with a beacon order equal to 8. There will be 16 time windows available for the ZR. Figure 16 show the schedule scheme for the beacon interval. Each associated ZR will be assigned the respective time window according to their position in the network tree. For viewing proposes only the 2 last bytes are shown in the device short address.

Time Window	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Device Short Address	00	01	02	09	20	21	28	-	3F	40	47	-	5E	5F	66	-

Figure 16 - Time Window schedule scheme

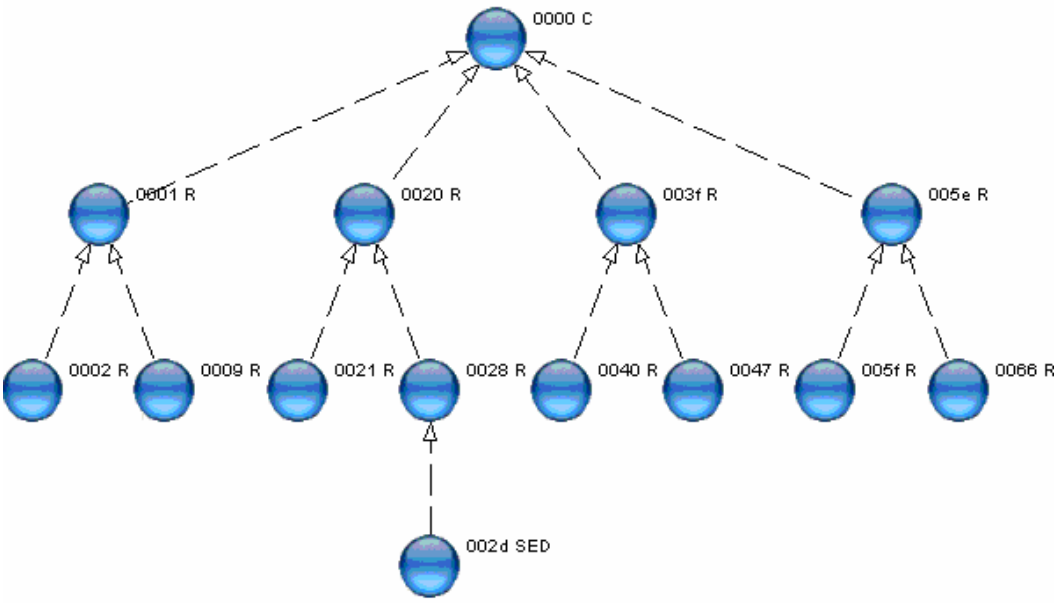


Figure 17 - Experiment 1 network topology – Tree Layout [10]

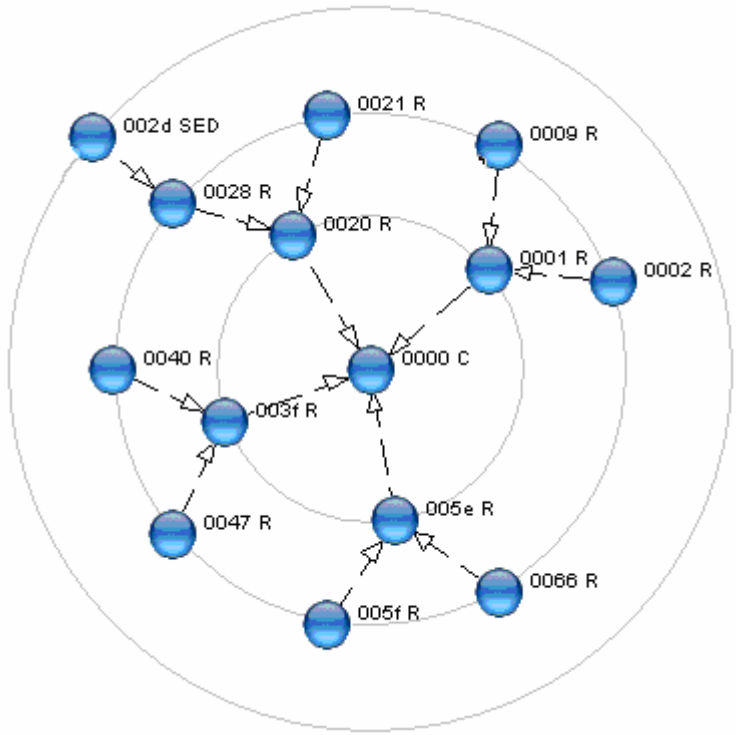


Figure 18 - Experiment 1 network topology – Radial Layout [10]

Figure 19 show the association and negotiation to broadcast beacons of a ZR. The ZR (ext. addr. 0x0000000200000002) associates with the PAN Coordinator, negotiates the beacon offset and starts broadcasting its beacon.

	Time	Time Delta	Source	Destination	Protocol	Packet Type
1	16:10:55.797	+00:00:04.266	0x0000	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1
	16:11:00.062	+00:00:04.266	0x0000	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1
2	16:11:00.064	+00:00:00.002	0x0000000200000002	0x0000	IEEE 802.15.4	Command: Association Request
	16:11:00.066	+00:00:00.002			IEEE 802.15.4	Acknowledgment
	16:11:00.070	+00:00:00.004	0x0000000200000002		IEEE 802.15.4	Command: Data Request
	16:11:00.071	+00:00:00.001			IEEE 802.15.4	Acknowledgment
	16:11:00.073	+00:00:00.002	0x0000000100000001	0x0000000200000002	IEEE 802.15.4	Command: Association Response
	16:11:00.075	+00:00:00.002			IEEE 802.15.4	Acknowledgment
	16:11:04.330	+00:00:04.255	0x0000	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1
	16:11:08.595	+00:00:04.266	0x0000	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1
	16:11:12.861	+00:00:04.266	0x0000	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1
3	16:11:12.863	+00:00:00.002	0x0001	0x0000	IEEE 802.15.4	Data
	16:11:12.865	+00:00:00.002			IEEE 802.15.4	Acknowledgment
	16:11:12.867	+00:00:00.002	0x0000	0x0001	IEEE 802.15.4	Data
	16:11:12.869	+00:00:00.002			IEEE 802.15.4	Acknowledgment
	16:11:13.138	+00:00:00.269			IEEE 802.15.4	Acknowledgment
4	16:11:17.128	+00:00:03.990	0x0000	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1
	16:11:17.404	+00:00:00.276	0x0001	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1
	16:11:21.394	+00:00:03.990	0x0000	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1
	16:11:21.669	+00:00:00.276	0x0001	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1
	16:11:25.660	+00:00:03.990	0x0000	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1
	16:11:25.935	+00:00:00.275	0x0001	0xffff	IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1

Figure 19 - Association and negotiation example [10]

In Figure 19, marked as 1, is the beacon broadcast of the ZC containing the network configuration BO and SO, as seen in the Packet Type field. Note that the Time Delta, 4266 ms, between beacons represent the beacon interval. Refer to Annex A for information comparing the theoretical and experimental values of the beacon interval.

The sequence of messages, marked as 2 in Figure 19, is the association procedure. The ZR with the extended address of 0x0000000200000002 sends an association request to the ZC (0x0000). The ZC acknowledge the reception of the request and informs the ZR that there is pending data (using the pending data field in the acknowledge frame). Then, the ZR sends a data request command frame requesting the pending data. The ZC replies with the association response command frame containing the status of the association, that in this case, is successful and the ZR is assigned with the short address 0x0001.

Now, the ZR is associated and can transmit in the network, but it still need to request the ZC for a beacon broadcast transmission permit and a time window slot (or transmission offset). The negotiation procedure is marked as 3 and Figure 20 shows the negotiation packets decoded. Until now, and after the network association, the ZR behaves as a normal ZED. When the negotiation for beacon transmission finishes, the ZR starts to broadcast beacons in its assigned time window, as seen in Figure 19 marked as 4.

Note that both the association and negotiation for beacon transmission took place during the ZC superframe.

The next table shows a packet decode list of the negotiation mechanism. The left list is the negotiation request (from ZR 0x0002 to ZC 0x0000) and the right is the negotiation accept (from ZC 0x0000 to ZR 0x0002). Highlighted is the data frame payload, in green (the first byte) is the negotiation type of message, blue (the second and third bytes) the information of the beacon order and superframe order and yellow (the fourth to sixth bytes) the beacon transmission offset value in symbols.

Frame 15 (Length = 27 bytes)	Frame 17 (Length = 27 bytes)
Time Stamp: 14:53:34.863	Time Stamp: 14:53:34.867
Frame Length: 27 bytes	Frame Length: 27 bytes
Capture Length: 27 bytes	Capture Length: 27 bytes
Link Quality Indication: 136	Link Quality Indication: 164
Receive Power: -49 dBm	Receive Power: -42 dBm
IEEE 802.15.4	IEEE 802.15.4
Frame Control: 0x8821	Frame Control: 0x8821
Sequence Number: 165	Sequence Number: 34
Destination PAN Identifier: 0x1234	Destination PAN Identifier: 0x1234
Destination Address: 0x0000	Destination Address: 0x0001
Source PAN Identifier: 0x1234	Source PAN Identifier: 0x1234

Source Address: 0x0001	Source Address: 0x0000
Frame Check Sequence: Correct	Frame Check Sequence: Correct
ZigBee NWK	ZigBee NWK
Frame Control: 0x0004	Frame Control: 0x0004
Destination Address: 0x0000	Destination Address: 0x0001
Source Address: 0x0001	Source Address: 0x0000
Radius = 1	Radius = 1
Sequence Number = 97	Sequence Number = 79
NWK Payload: 01:08:04:00:00:00	NWK Payload: 02:08:04:00:3c:00

Figure 20 - Negotiation mechanism packet decode example [10]

1							2	3				
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields	Beacon payload	LQI	FCS	
+473217		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit	00 00 84			
=1181408862	21	BCN 0 0 0 1	0x83	0x1234	0xFFFF	0x0000	08 04 15 0 1 0	0 1	56 34 12	112	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+269211		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1181678073	15	BCN 0 0 0 1	0xFB	0x1234	0xFFFF	0x0001	08 04 15 0 1 1	0 1		160	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+270582		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1181948655	15	BCN 0 0 0 1	0x36	0x1234	0xFFFF	0x0002	08 04 15 0 1 1	0 1		156	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+266768		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1182215423	15	BCN 0 0 0 1	0x01	0x1234	0xFFFF	0x0009	08 04 15 0 1 1	0 1		128	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+262612		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1182478035	15	BCN 0 0 0 1	0x37	0x1234	0xFFFF	0x0020	08 04 15 0 1 1	0 1		84	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+270980		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1182749015	15	BCN 0 0 0 1	0xD4	0x1234	0xFFFF	0x0021	08 04 15 0 1 1	0 1		68	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+266351		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1183015366	15	BCN 0 0 0 1	0xF9	0x1234	0xFFFF	0x0028	08 04 15 0 1 1	0 1		84	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+529508		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1183544874	15	BCN 0 0 0 1	0xF5	0x1234	0xFFFF	0x003F	08 04 15 0 1 1	0 1		52	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+270887		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1183815761	15	BCN 0 0 0 1	0x23	0x1234	0xFFFF	0x0040	08 04 15 0 1 1	0 1		76	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+266217		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1184081978	15	BCN 0 0 0 1	0x11	0x1234	0xFFFF	0x0047	08 04 15 0 1 1	0 1		52	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+582672		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1184664650	15	BCN 0 0 0 1	0xAC	0x1234	0xFFFF	0x005E	08 04 15 0 1 1	0 1		136	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+270802		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1184935452	15	BCN 0 0 0 1	0x6D	0x1234	0xFFFF	0x005F	08 04 15 0 1 1	0 1		152	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+266496		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1185201948	15	BCN 0 0 0 1	0x11	0x1234	0xFFFF	0x0066	08 04 15 0 1 1	0 1		0	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields	Beacon payload	LQI	FCS	
+475134		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit	00 00 84			
=1185677082	21	BCN 0 0 0 1	0x84	0x1234	0xFFFF	0x0000	08 04 15 0 1 0	0 1	56 34 12	112	OK	
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS	
+271436		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit				
=1185948518	15	BCN 0 0 0 1	0xFC	0x1234	0xFFFF	0x0001	08 04 15 0 1 1	0 1		160	OK	

Figure 21 - Experiment 1 beacon frames [9]

In Figure 21 is possible to identify the assigned window time slots by the time between beacon receptions (marked as 1), the beacon source address (marked as 2) and the beacon superframe specification (marked as 3).

In order to test the functionality of the network a message stream is sent from the end device 0x002d associated with the ZR 0x0028 to the ZR 0x0066. Figure 22 shows a log capture of the message flow.

Time	Time Delta	Source	Destination	NWK Source	NWK Destination	Protocol	Packet Type	
16:59:41.501	+00:00:00.267	0x0028	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:41.502	+00:00:00.001	0x002d	0x0028	0x002d	0x0066	Zigbee NWK	NWK Data	1
16:59:41.504	+00:00:00.002					IEEE 802.15.4	Acknowledgment	
16:59:42.030	+00:00:00.526	0x003f	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:42.301	+00:00:00.271	0x0040	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:42.567	+00:00:00.267	0x0047	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:43.151	+00:00:00.584	0x005e	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:43.422	+00:00:00.271	0x005f	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:43.688	+00:00:00.267	0x0066	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:44.163	+00:00:00.474	0x0000	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 0, N	
16:59:44.434	+00:00:00.271	0x0001	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:44.705	+00:00:00.271	0x0002	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:44.971	+00:00:00.267	0x0009	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:45.234	+00:00:00.263	0x0020	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:45.236	+00:00:00.002	0x0028	0x0020	0x002d	0x0066	Zigbee NWK	NWK Data	2
16:59:45.240	+00:00:00.004					IEEE 802.15.4	Acknowledgment	
16:59:45.505	+00:00:00.265	0x0021	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:45.768	+00:00:00.263	0x0028	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:46.301	+00:00:00.532	0x003f	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:46.571	+00:00:00.271	0x0040	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:46.838	+00:00:00.267	0x0047	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:47.421	+00:00:00.582	0x005e	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:47.691	+00:00:00.271	0x005f	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:47.958	+00:00:00.267	0x0066	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:48.431	+00:00:00.473	0x0000	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 0, N	
16:59:48.436	+00:00:00.005	0x0020	0x0000	0x002d	0x0066	Zigbee NWK	NWK Data	
16:59:48.438	+00:00:00.002					IEEE 802.15.4	Acknowledgment	
16:59:48.440	+00:00:00.002	0x0000	0x005e	0x002d	0x0066	Zigbee NWK	NWK Data	3
16:59:48.442	+00:00:00.002					IEEE 802.15.4	Acknowledgment	
16:59:48.704	+00:00:00.262	0x0001	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:48.975	+00:00:00.271	0x0002	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:49.241	+00:00:00.267	0x0009	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:49.502	+00:00:00.261	0x0020	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:49.772	+00:00:00.269	0x0021	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:50.039	+00:00:00.268	0x0028	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:50.571	+00:00:00.531	0x003f	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:50.841	+00:00:00.271	0x0040	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:51.108	+00:00:00.267	0x0047	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:51.691	+00:00:00.583	0x005e	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:51.693	+00:00:00.002	0x005e	0x0066	0x002d	0x0066	Zigbee NWK	NWK Data	
16:59:51.695	+00:00:00.002	0x005e	0x0066	0x002d	0x0066	Zigbee NWK	NWK Data	
16:59:51.697	+00:00:00.002	0x005e	0x0066	0x002d	0x0066	Zigbee NWK	NWK Data	4
16:59:51.699	+00:00:00.002					IEEE 802.15.4	Acknowledgment	
16:59:51.963	+00:00:00.264	0x005f	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:52.229	+00:00:00.267	0x0066	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 1	
16:59:52.703	+00:00:00.473	0x0000	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 4, PC: 1, AP: 0, N	

Figure 22 - Experiment 1 message flow - capture log [10]

In Figure 22, marked a 1, shows the first transmission of the packet from the ZED 0x002d to its parent (ZR 0x0028). Note that this transmission is carried out during the ZR (0x0028) superframe. The routing of the data frame from the ZR (0x0028) to its parent in the cluster-tree (ZR 0x0020) is marked as 2. The multi-hop continues (marked as 3) with the routing of the frame from the ZR 0x0020 to the ZC (0x0000) and to the ZR 0x005e. This transmission sequence (marked as 3) is carried out during the ZC superframe. Then, ZR 0x005e routes the frame to its final destination, the ZR 0x0066 (marked as 4). The retransmission of the data frame, marked as 4 in Figure 22, is due to the failure of the acknowledge transmission of ZR 0x0066.

4.4 Experiment 2

In this experiment we want to evaluate the network behaviour with depth 3 ZR. The next figure represents the network topology after the association and negotiation mechanism.

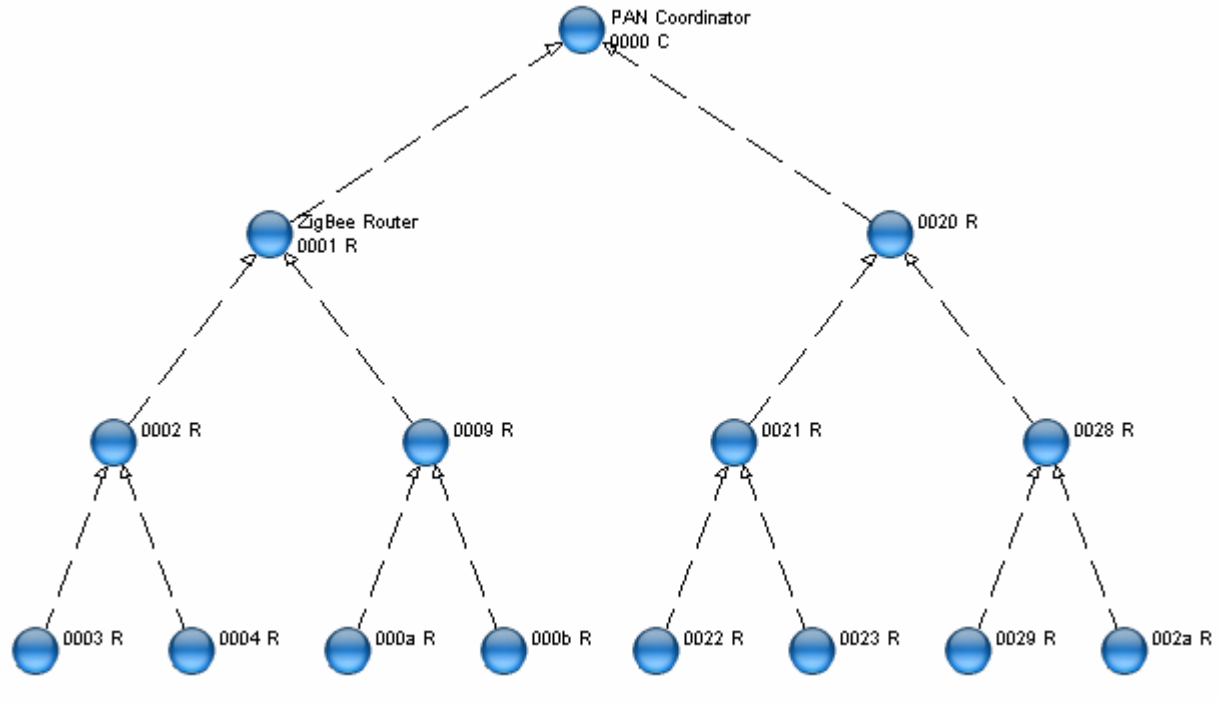


Figure 23 - Experiment 2 network topology - Tree Layout [10]

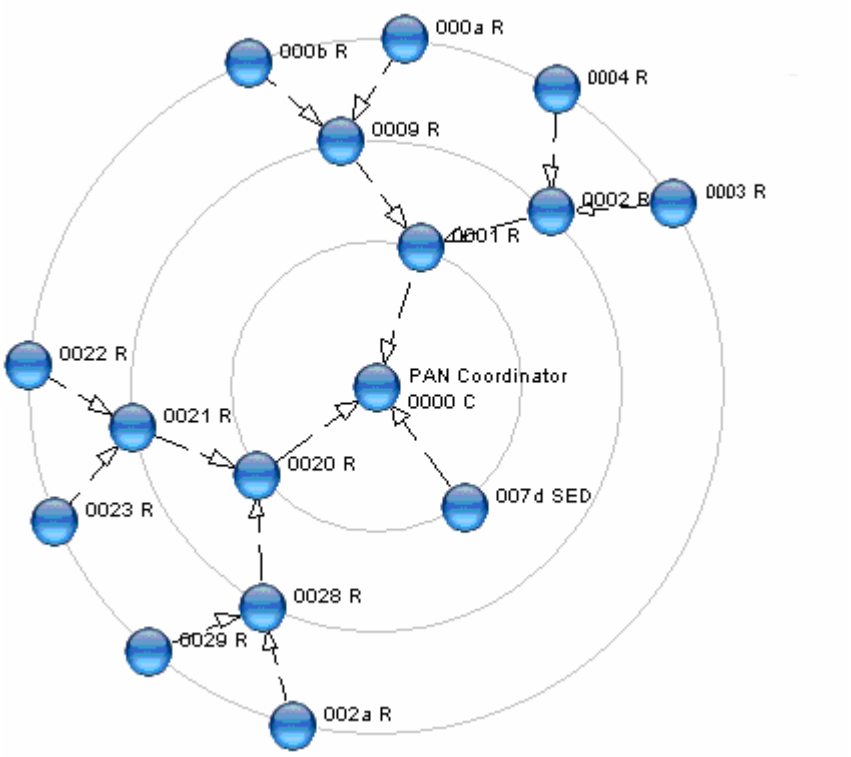
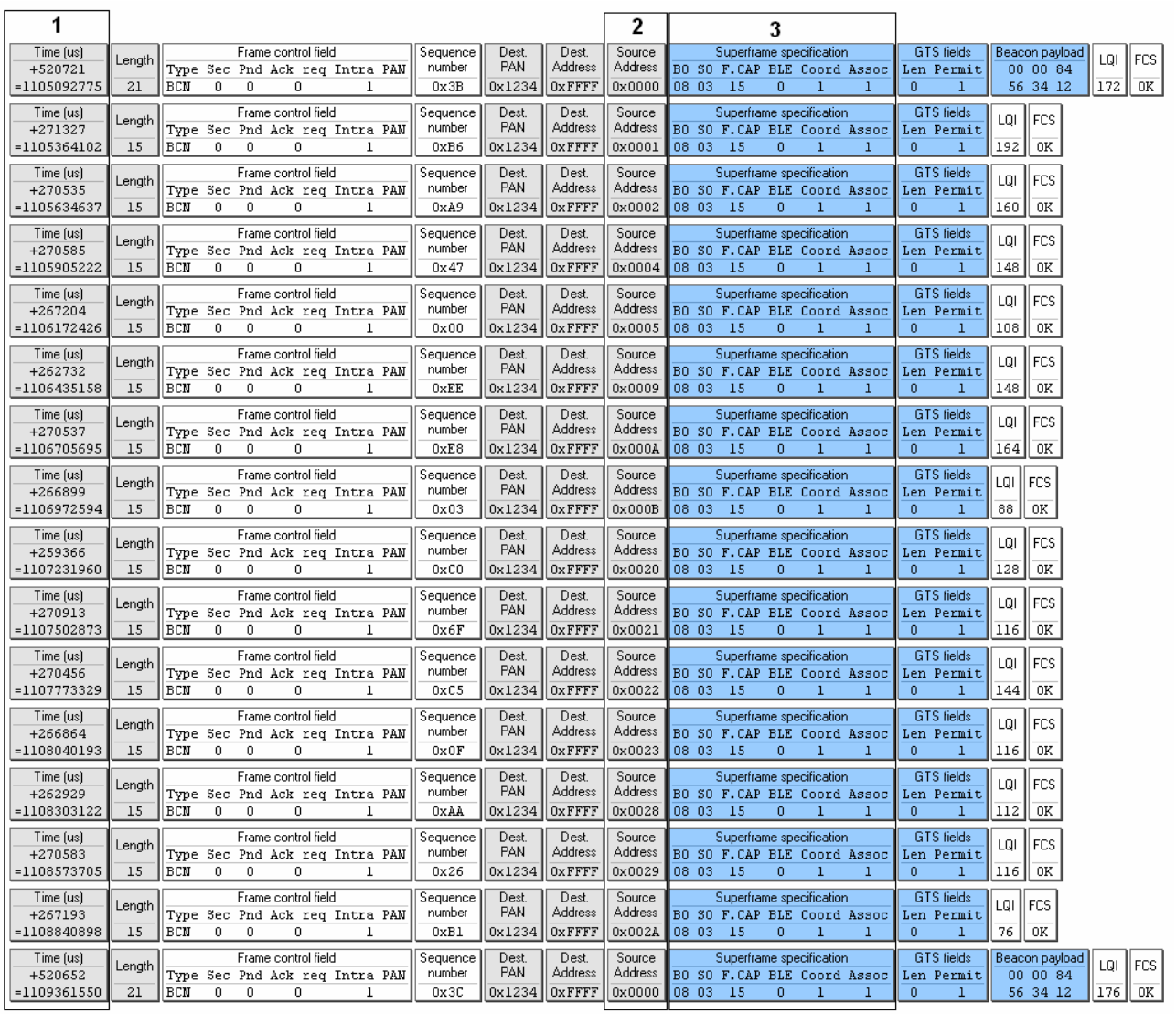


Figure 24 - Experiment 2 network topology - Radial Layout [10]



1						2	3				
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields	Beacon payload	LQI	FCS
+520721		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit	00 00 84		
=1105092775	21	BCN 0 0 0 1	0x3B	0x1234	0xFFFF	0x0000	08 03 15 0 1 1	0 1	56 34 12	172	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+271327		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1105364102	15	BCN 0 0 0 1	0xB6	0x1234	0xFFFF	0x0001	08 03 15 0 1 1	0 1		192	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+270535		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1105634637	15	BCN 0 0 0 1	0xA9	0x1234	0xFFFF	0x0002	08 03 15 0 1 1	0 1		160	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+270585		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1105905222	15	BCN 0 0 0 1	0x47	0x1234	0xFFFF	0x0004	08 03 15 0 1 1	0 1		148	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+267204		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1106172426	15	BCN 0 0 0 1	0x00	0x1234	0xFFFF	0x0005	08 03 15 0 1 1	0 1		108	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+262732		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1106435158	15	BCN 0 0 0 1	0xEE	0x1234	0xFFFF	0x0009	08 03 15 0 1 1	0 1		148	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+270537		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1106705695	15	BCN 0 0 0 1	0xE8	0x1234	0xFFFF	0x000A	08 03 15 0 1 1	0 1		164	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+266899		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1106972594	15	BCN 0 0 0 1	0x03	0x1234	0xFFFF	0x000B	08 03 15 0 1 1	0 1		88	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+259366		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1107231960	15	BCN 0 0 0 1	0xC0	0x1234	0xFFFF	0x0020	08 03 15 0 1 1	0 1		128	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+270913		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1107502873	15	BCN 0 0 0 1	0x6F	0x1234	0xFFFF	0x0021	08 03 15 0 1 1	0 1		116	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+270456		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1107773329	15	BCN 0 0 0 1	0xC5	0x1234	0xFFFF	0x0022	08 03 15 0 1 1	0 1		144	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+266864		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1108040193	15	BCN 0 0 0 1	0x0F	0x1234	0xFFFF	0x0023	08 03 15 0 1 1	0 1		116	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+262929		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1108303122	15	BCN 0 0 0 1	0xAA	0x1234	0xFFFF	0x0028	08 03 15 0 1 1	0 1		112	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+270583		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1108573705	15	BCN 0 0 0 1	0x26	0x1234	0xFFFF	0x0029	08 03 15 0 1 1	0 1		116	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields		LQI	FCS
+267193		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit			
=1108840898	15	BCN 0 0 0 1	0xB1	0x1234	0xFFFF	0x002A	08 03 15 0 1 1	0 1		76	OK
Time (us)	Length	Frame control field	Sequence number	Dest. PAN	Dest. Address	Source Address	Superframe specification	GTS fields	Beacon payload	LQI	FCS
+520652		Type Sec Pnd Ack req Intra PAN					BO SO F.CAP BLE Coord Assoc	Len Permit	00 00 84		
=1109361550	21	BCN 0 0 0 1	0x3C	0x1234	0xFFFF	0x0000	08 03 15 0 1 1	0 1	56 34 12	176	OK

Figure 25 - Experiment 2 beacon list [9]

In Figure 25 is possible to identify the assigned window time slots by the time between beacon receptions (marked as 1), the beacon source address (marked as 2) and the beacon superframe specification (marked as 3).

In order to test the functionality of the network a message stream is sent from the end device 0x007d associated with the ZC 0x0000 to the ZR 0x0066. Figure 26 show a radial diagram of the message flow.

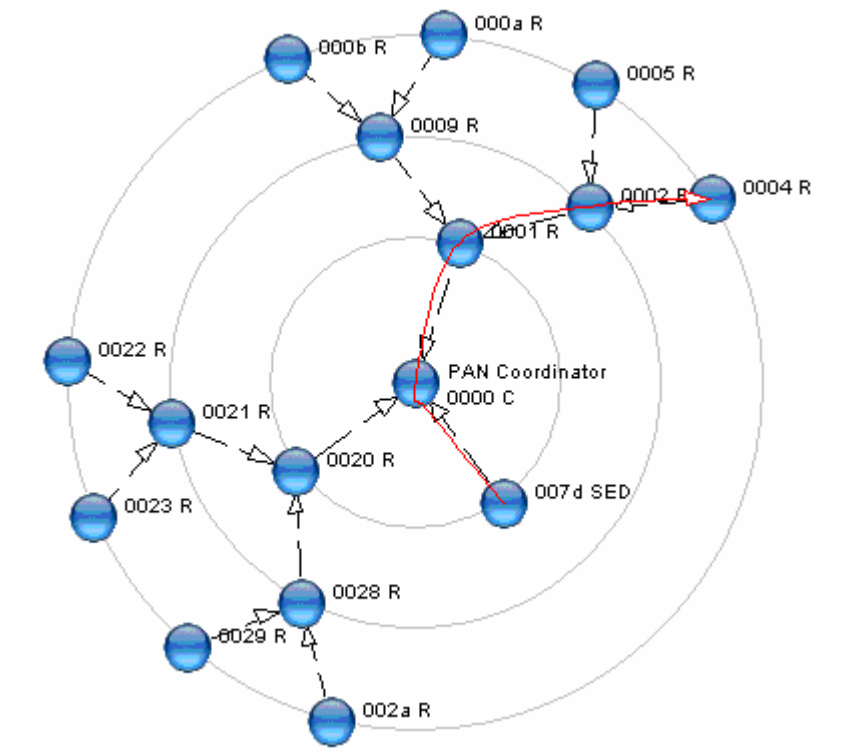


Figure 26 - Experiment 2 message flow - radial layout [10]

	Time	Time Delta	Source	Destination	NWK Source	NWK Destination	Protocol	Packet Type
	19:05:40.682	+00:00:00.259	0x000a	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
	19:05:40.692	+00:00:00.010	0x0005	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
	19:05:40.949	+00:00:00.257	0x000b	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
1	19:05:41.171	+00:00:00.222	0x007d	0x0000	0x007d	0x0004	Zigbee NWK	NWK Data
1	19:05:41.173	+00:00:00.002					IEEE 802.15.4	Acknowledgment
1	19:05:41.176	+00:00:00.004	0x007d	0x0000	0x007d	0x0004	Zigbee NWK	NWK Data
1	19:05:41.182	+00:00:00.005	0x007d	0x0000	0x007d	0x0004	Zigbee NWK	NWK Data
	19:05:41.200	+00:00:00.018	0x0020	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
	19:05:41.470	+00:00:00.270	0x0021	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
	19:05:42.270	+00:00:00.800	0x0028	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
	19:05:42.541	+00:00:00.271	0x0029	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
	19:05:42.807	+00:00:00.267	0x002a	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
	19:05:43.329	+00:00:00.521	0x0000	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1, N
2	19:05:43.331	+00:00:00.002	0x0000	0x0001	0x007d	0x0004	Zigbee NWK	NWK Data
2	19:05:43.333	+00:00:00.002	0x0000	0x0001	0x007d	0x0004	Zigbee NWK	NWK Data
2	19:05:43.337	+00:00:00.003	0x0000	0x0001	0x007d	0x0004	Zigbee NWK	NWK Data
2	19:05:43.338	+00:00:00.002					IEEE 802.15.4	Acknowledgment
	19:05:43.607	+00:00:00.268	0x0001	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
3	19:05:43.609	+00:00:00.002	0x0001	0x0002	0x007d	0x0004	Zigbee NWK	NWK Data
3	19:05:43.614	+00:00:00.006	0x0001	0x0002	0x007d	0x0004	Zigbee NWK	NWK Data
3	19:05:43.620	+00:00:00.005	0x0001	0x0002	0x007d	0x0004	Zigbee NWK	NWK Data
	19:05:43.878	+00:00:00.259	0x0002	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
4	19:05:43.880	+00:00:00.002					IEEE 802.15.4	Acknowledgment
4	19:05:43.882	+00:00:00.002	0x0002	0x0004	0x007d	0x0004	Zigbee NWK	NWK Data
4	19:05:43.884	+00:00:00.003					IEEE 802.15.4	Acknowledgment
4	19:05:43.886	+00:00:00.002	0x0002	0x0004	0x007d	0x0004	Zigbee NWK	NWK Data
4	19:05:43.888	+00:00:00.002					IEEE 802.15.4	Acknowledgment
4	19:05:43.892	+00:00:00.003	0x0002	0x0004	0x007d	0x0004	Zigbee NWK	NWK Data
4	19:05:43.894	+00:00:00.002	0x0002	0x0004	0x007d	0x0004	Zigbee NWK	NWK Data
4	19:05:43.956	+00:00:00.063	0x0002	0x0004	0x007d	0x0004	Zigbee NWK	NWK Data
4	19:05:43.959	+00:00:00.002					IEEE 802.15.4	Acknowledgment
	19:05:44.152	+00:00:00.193	0x0004	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
	19:05:44.419	+00:00:00.267	0x0005	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1
	19:05:44.681	+00:00:00.262	0x0009	0xffff			IEEE 802.15.4	Beacon: BO: 8, SO: 3, PC: 1, AP: 1

Figure 27 - Experiment 2 message flow - capture log [10]

In Figure 27, marked a 1, shows the first transmission of the packet from the ZED 0x007d to its parent (ZC 0x0000). The routing of the data frame from the ZC to its child router in the cluster-tree (ZR 0x0001) is marked as 2. The multi-hop continues (marked as 3), with the routing of the frame from the ZR 0x0001 to the ZR 0x0002. Then, ZR 0x0002 routes the frame to its final destination, the ZR 0x0004 (marked as 4). The retransmission of the data frame is due to the failure of the acknowledge frames transmission or reception.

5 References

[1] IEEE 802.15.4 Standard-2003, "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, 2003.

[2] ZigBee-Alliance, "ZigBee specification," http://www.ZigBee.org/, 2006.

[3] David Gay, Phil Levis, Rob von Behren, Matt Welsh, Eric Brewer, and David Culler, "The nesC language: A holistic approach to network embedded systems", in PLDI'03.

[4] www.tinyos.net

[5] The OPEN ZigBee implementation - www.open-zb.net

[6] Crossbow Technologies INC. http://www.xbow.com

[7] ATmega128L 8-bit AVR Microntroller Datasheet, Atmel ref: 2467MAVR-11/04, http://www.atmel.com

[8] Chipcon, SmartRF CC2420 Datasheet (rev 1.3), 2005.http://www.chipcon.com

[9] Chipcon, "Chipcon Packet Sniffer for IEEE 802.15.4", 2006

[10]Daintree Networks, "Sensor Network Analyser, www.daintree.net," 2006

Annex A – Theoretical and effective values of the beacon interval

MICAz Beacon Interval Durations					
BO	Symbols	Backoff Periods	Duration (us) Effective Values	Duration (us) Theoretical Values	Clock Ticks
0	960	48	16667,52	15360	240
1	1920	96	33335,04	30720	479
2	3840	192	66670,08	61440	959
3	7680	384	133340,16	122880	1917
4	15360	768	266680,32	245760	3835
5	30720	1536	533360,64	491520	7670
6	61440	3072	1066721,28	983040	15340
7	122880	6144	2133442,56	1966080	30679
8	245760	12288	4266885,12	3932160	61359
9	491520	24576	8533770,24	7864320	122717
10	983040	49152	17067540,48	15728640	245435
11	1966080	98304	34135080,96	31457280	490870
12	3932160	196608	68270161,92	62914560	981739
13	7864320	393216	136540323,8	125829120	1963479
14	15728640	786432	273080647,7	251658240	3926958

MICAz beacon interval durations - Effective values/ Theoretical values.