

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER'S THESIS

---

# Wireless Sensor Networks on-board Aircrafts: design and implementation of the Medium Access Control protocol

---

*Author:*

Cristian SÁNCHEZ

*Supervisor:*

Eriza FAZLI

*Co-supervisor:*

Josep PARADELLS

*A thesis submitted in fulfilment of the requirements  
for the degree of  
Master of Science in Information and Communication Technologies  
in the*

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

*through an internship in*

TriaGnoSys GmbH

December 2013

*”Estamos en este mundo para convivir en armonía.  
Quienes lo saben no luchan entre sí.”*

Sidharta Gautama

UNIVERSITAT POLITÈCNICA DE CATALUNYA

## *Abstract*

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Master of Science in Information and Communication Technologies

### **Wireless Sensor Networks on-board Aircrafts: design and implementation of the Medium Access Control protocol**

by Cristian SÁNCHEZ

The Wireless Sensor Networks is an upcoming technology with a huge potential. It has the ability of monitor an environment and provide the data collected to a central entity in a wireless way. The Sensor Wireless Application Network (SWAN) project, aims to develop a reliable on-board aircraft WSN prototype.

This thesis deals with the design, implementation and testing of a Medium Access Control (MAC) protocol, planned on the SWAN project and adapted to TriaGnoSys requirements. Also it has been designed, implemented and tested a TDMA scheduling algorithm conformed to the MAC sublayer requirements.

The system has been implemented and tested at TriaGnoSys GmbH, Wessling, Germany.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>viii</b>
<b>Abbreviations</b>	<b>ix</b>
<b>1 Introduction.</b>	<b>1</b>
1.1 Motivation.	1
1.2 Background and Scope.	2
1.3 Objectives.	3
1.4 Outline of the thesis.	3
<b>2 Sensor Wireless Application Network overview.</b>	<b>4</b>
2.1 Introduction.	4
2.2 System architecture.	4
2.2.1 Wireless network topology.	5
2.2.2 Redudancy.	6
2.2.3 SWAN working modes.	7
2.2.4 Node localization service.	8
2.3 Properties and benefits of WSNs on-board aircrafts.	9
2.4 WSN applications on-board aircrafts.	10
2.4.1 Flight test installation.	10
2.4.2 Aircraft structure health monitoring.	11
2.4.3 Fire and smoke detection systems.	11
2.4.4 Proximity switches on PAX seat.	12
2.4.5 Other applications.	12
<b>3 Medium Access Control protocol in WSNs.</b>	<b>14</b>
3.1 Introduction.	14
3.1.1 Main families of MAC protocols.	14
3.1.2 MAC operations.	15
3.1.3 Benefits and properties of designing correctly a MAC protocol.	15
3.2 Medium Access Control protocol in SWAN.	16
3.2.1 General aspects.	17

3.2.2	Addressing. . . . .	17
3.2.3	Superframe structure. . . . .	18
3.2.3.1	Maintenance mode superframe. . . . .	19
3.2.3.2	Operation mode superframe. . . . .	19
3.2.4	Frame formats. . . . .	20
3.2.4.1	MAC frame. . . . .	20
3.2.4.2	Beacon frame. . . . .	21
3.2.4.3	Maintenance command payload. . . . .	21
3.2.4.4	Operation payload. . . . .	22
3.2.5	MLME commands. . . . .	22
3.2.5.1	From coordinator to sensor/actuator. . . . .	23
3.2.5.2	From sensor/actuator to coordinator. . . . .	25
3.2.6	MAC procedures. . . . .	26
3.2.6.1	Association. . . . .	26
3.2.6.2	Disassociation. . . . .	27
3.2.6.3	Ranging measurements. . . . .	28
<b>4</b>	<b>MAC sublayer: Maintenance Mode implementation.</b>	<b>29</b>
4.1	Introduction. . . . .	29
4.2	Implementation platform. . . . .	30
4.3	Physical layer parameters. . . . .	32
4.4	MAC sublayer parameters. . . . .	34
4.4.1	FSM states. . . . .	34
4.4.1.1	MAC sublayer states. . . . .	34
4.4.1.2	Superframe states. . . . .	35
4.4.2	MAC constants. . . . .	35
4.4.3	MAC PIB attributes. . . . .	36
4.5	MAC sublayer logic. . . . .	39
4.5.1	Timer events. . . . .	40
4.5.1.1	FRAME_START . . . . .	40
4.5.1.2	BEACON_TIMEOUT . . . . .	41
4.5.1.3	DOWNLINKCFP_START . . . . .	41
4.5.1.4	DOWNLINK_SLOT_EXPIRATION . . . . .	42
4.5.1.5	START_SLEEPING . . . . .	43
4.5.1.6	UPLINKCAP_START . . . . .	43
4.5.1.7	START_SLEEPING_INACTIVE . . . . .	44
4.5.1.8	SEND_ASSOCIATION_REQ . . . . .	45
4.5.1.9	SEND_DISASSOCIATION_REQ . . . . .	45
4.5.1.10	SEND_ASSOCIATION_RES_ACK . . . . .	45
4.5.2	Reception of MAC frames from the physical layer. . . . .	46
4.5.2.1	TDMA_BEACON_PACKET . . . . .	46
4.5.2.2	DISASSOCIATION_REQUEST_CMD . . . . .	47
4.5.2.3	ASSOCIATION_REQUEST_CMD . . . . .	48
4.5.2.4	ASSOCIATION_RESPONSE_CMD . . . . .	49
4.5.2.5	ACK_ASSOCIATION_RESPONSE . . . . .	49
4.5.3	Transmission of MAC frames to the physical layer. . . . .	51
4.5.4	Reception of control commands from upper layers. . . . .	52

4.6	MAC implementation extensions. . . . .	53
4.6.1	Communication with the upper layers. Data base consultation. . .	53
4.6.2	Several clusters within the same wireless sensor network. . . . .	55
4.6.3	Scan period. . . . .	56
4.7	Analysis, experiments and results. . . . .	61
4.7.1	Deployment 1. . . . .	63
4.7.2	Deployment 2. . . . .	64
4.7.3	Analysis and conclusions. . . . .	65
4.8	Conclusions and open problems. . . . .	66
<b>5</b>	<b>TDMA Scheduling algorithm. . . . .</b>	<b>67</b>
5.1	Introduction. . . . .	67
5.2	Related work. . . . .	69
5.3	Basics of the implemented scheduling algorithm. . . . .	70
5.3.1	Scheduling definitions and terminology. . . . .	70
5.3.2	TDMA structure. . . . .	72
5.3.3	TDMA frame sharing among clusters. . . . .	73
5.4	Scheduling algorithm implementation. . . . .	74
5.4.1	Input parameters. . . . .	75
5.4.1.1	Input Matrix. . . . .	75
5.4.1.2	Slot Parameters. . . . .	76
5.4.1.3	Occupation percentage of the InterNodelink segment. . .	77
5.4.2	Capacity calculation. . . . .	77
5.4.2.1	Slot occupation calculation. . . . .	78
5.4.2.2	Occupation percentages calculation. . . . .	79
5.4.3	Allocation algorithm. . . . .	80
5.4.3.1	Superframe length calculation. . . . .	81
5.4.3.2	The calculation of the slots where each segment start. . .	81
5.4.3.3	Determine the initial position of the pointers that indi- cate the next free slot of each segment. . . . .	81
5.4.3.4	Extract data from the Input Matrix to simulate the node request. . . . .	83
5.4.3.5	Main algorithm behavior. . . . .	83
5.4.4	Output parameters. . . . .	85
5.5	Analysis, experiments and results. . . . .	86
5.5.1	Experiment 1. . . . .	86
5.5.2	Experiment 2. . . . .	89
5.5.3	Experiment 3. . . . .	91
5.5.4	Experiment 4. . . . .	93
5.6	Conclusions and open problems. . . . .	94
<b>6</b>	<b>Conclusions. . . . .</b>	<b>97</b>
	<b>Bibliography . . . . .</b>	<b>99</b>

# List of Figures

2.1	SWAN architecture . . . . .	5
2.2	SWAN's wireless network topology . . . . .	6
2.3	Basic WSN redudancy scenario . . . . .	7
2.4	Multiple WSN clusters approach . . . . .	9
3.1	Maintenance mode superframe . . . . .	19
3.2	Operation mode superframe . . . . .	20
3.3	MAC frame . . . . .	21
3.4	Beacon payload . . . . .	21
3.5	Maintenance command payload . . . . .	21
3.6	Operation command payload . . . . .	22
3.7	MLME commands . . . . .	23
3.8	Association response payload . . . . .	24
3.9	Disassociation request payload . . . . .	24
3.10	Ranging request payload . . . . .	25
3.11	Set PAN ID payload . . . . .	25
3.12	Association request payload . . . . .	26
3.13	Ranging response payload . . . . .	26
3.14	Association procedure . . . . .	27
3.15	Disassociation procedure . . . . .	27
3.16	Ranging procedure . . . . .	28
3.17	Superframe of ranging procedure . . . . .	28
4.1	Event-driven programming paradigm schema . . . . .	32
4.2	Flow chart. FRAME_START event . . . . .	40
4.3	Flow chart. BEACON_TIMEOUT event . . . . .	41
4.4	Flow chart. DOWNLINKCFP_START event . . . . .	42
4.5	Flow chart. DOWNLINK_SLOT_EXPIRATION event . . . . .	43
4.6	Flow chart. START_SLEEPING event . . . . .	43
4.7	Flow chart. UPLINKCAP_START event . . . . .	44
4.8	Flow chart. START_SLEEPING_INACTIVE event . . . . .	45
4.9	Flow chart. SEND_ASSOCIATION_REQ event . . . . .	45
4.10	Flow chart. SEND_DISASSOCIATION_REQ event . . . . .	46
4.11	Flow chart. SEND_ASSOCIATION_RES_ACK event . . . . .	46
4.12	Flow chart. TDMA_BEACON_PACKET event . . . . .	47
4.13	Flow chart. DISASSOCIATION_REQUEST_CMD event . . . . .	48
4.14	Flow chart. ASSOCIATION_REQUEST_CMD event . . . . .	49
4.15	Flow chart. ASSOCIATION_RESPONSE_CMD event . . . . .	50

4.16	Flow chart. <b>ACK_ASSOCIATION_RESPONSE</b> event . . . . .	50
4.17	Flow chart. <b>Attempt Transmission</b> event . . . . .	51
4.18	Flow chart. <b>Initiate CSMA/CA algorithm</b> event . . . . .	52
4.19	Flow chart. <b>Control Command Association Response</b> event . . . . .	53
4.20	Message exchange between MAC sublayer and upper layers. . . . .	54
4.21	Several clusters within the same WSN. . . . .	56
4.22	Scan period. . . . .	57
4.23	Scan Beacon payload. . . . .	58
4.24	Flow chart. <b>SCAN_START</b> event . . . . .	59
4.25	Flow chart. <b>SCAN_BEACON_PACKET</b> event . . . . .	60
4.26	Boeing A321 passenger's cabin dimensions . . . . .	61
4.27	Position of the coordinators within the cabin. . . . .	62
5.1	TDMA structure 1 . . . . .	71
5.2	TDMA structure 2 . . . . .	73
5.3	Inter-cluster TDMA interferences . . . . .	74
5.4	Flow-chart. <b>Scheduling Algorithm</b> . . . . .	75
5.5	Frame pointers schema . . . . .	83
5.6	Message scheduling schema . . . . .	84
5.7	Experiment 1 plots . . . . .	88
5.8	Experiment 2 plots . . . . .	90
5.9	Experiment 3 plots . . . . .	92
5.10	Experiment 4 plots . . . . .	93



# List of Tables

3.1	Beacon payload fields . . . . .	22
3.2	MLME commands from coordinator to sensor/actuator . . . . .	23
3.3	MLME commands from sensor/actuator to coordinator . . . . .	25
4.1	MAC sublayer states . . . . .	35
4.2	Superframe states . . . . .	35
4.3	MAC constants . . . . .	36
4.4	MAC PIB attributes. . . . .	39
4.5	Superframe structure . . . . .	61
4.6	Deployment 1. Nodes associated to each coordinator . . . . .	63
4.7	Deployment 1. Association time (ms) . . . . .	63
4.8	Deployment 1. Q factor . . . . .	63
4.9	Deployment 2. Nodes associated to each coordinator . . . . .	64
4.10	Deployment 2. Association time (ms) . . . . .	64
4.11	Deployment 2. Q factor . . . . .	64
5.1	Input Matrix . . . . .	76
5.2	Experiment 1 InputMatrix . . . . .	86
5.3	Experiment 2 InputMatrix . . . . .	89
5.4	Experiment 3 InputMatrix . . . . .	91
5.5	Experiment 4 InputMatrix . . . . .	93

# Abbreviations

<b>BAN</b>	<b>B</b> ody <b>A</b> rea <b>N</b> etworks
<b>CDMA</b>	<b>C</b> ode <b>D</b> ivision <b>M</b> ultiple <b>A</b> ccess
<b>CAP</b>	<b>C</b> ontention <b>A</b> ccess <b>P</b> eriod
<b>CCA</b>	<b>C</b> lear <b>C</b> hannel <b>A</b> ssessment
<b>CFP</b>	<b>C</b> ontention <b>F</b> ree <b>P</b> eriod
<b>CS</b>	<b>C</b> arrier <b>S</b> ense
<b>CSMA/CA</b>	<b>C</b> arrier <b>S</b> ense <b>M</b> ultiple <b>A</b> ccess / <b>C</b> ollision <b>A</b> voidance
<b>DSSS</b>	<b>D</b> irect <b>S</b> equence <b>S</b> pread <b>S</b> pectrum
<b>ED</b>	<b>E</b> nergy <b>D</b> etection
<b>FCS</b>	<b>F</b> rame <b>C</b> heck <b>S</b> equence
<b>FDMA</b>	<b>F</b> requency <b>D</b> ivision <b>M</b> ultiple <b>A</b> ccess
<b>FFD</b>	<b>F</b> ull <b>F</b> unction <b>D</b> evice
<b>FSM</b>	<b>F</b> inite <b>S</b> tate <b>M</b> achine
<b>FTI</b>	<b>F</b> light <b>T</b> est <b>I</b> nstallation
<b>GTS</b>	<b>G</b> uaranted <b>T</b> ime <b>S</b> lot
<b>HMI</b>	<b>H</b> uman <b>M</b> achine <b>I</b> nterface
<b>ID</b>	<b>I</b> Dentifier
<b>IEEE</b>	<b>I</b> nsitute of <b>E</b> lectrical and <b>E</b> lectronics <b>E</b> ngineers
<b>IFS</b>	<b>I</b> nter <b>F</b> rame <b>S</b> pacing
<b>LASA</b>	<b>L</b> ow-energy <b>A</b> daptative <b>S</b> lot <b>A</b> llocation
<b>LLC</b>	<b>L</b> ogical <b>L</b> ink <b>C</b> ontrol
<b>LQI</b>	<b>L</b> ink <b>Q</b> uality <b>I</b> ndication
<b>LR-WPAN</b>	<b>L</b> ow- <b>R</b> ate <b>W</b> ireless <b>P</b> ersonal <b>A</b> rea <b>N</b> etwork
<b>MAC</b>	<b>M</b> edium <b>A</b> ccess <b>C</b> ontrol
<b>MCPS</b>	<b>M</b> AC <b>C</b> ommon <b>P</b> art <b>S</b> ublayer

---

<b>MFR</b>	<b>MAC FooteR</b>
<b>MHR</b>	<b>MAC HeadeR</b>
<b>MLME</b>	<b>MAC subLayer Management Entity</b>
<b>MPDU</b>	<b>MAC Protocol Data Unit</b>
<b>MSDU</b>	<b>MAC Service Data Unit</b>
<b>OSI</b>	<b>Open Systems Interconnection</b>
<b>PAN</b>	<b>Personal Area Network</b>
<b>PHY</b>	<b>PHYsical layer</b>
<b>QoS</b>	<b>Quality of Service</b>
<b>RFD</b>	<b>Reduced Function Device</b>
<b>RTS/CTS</b>	<b>Request-To-Send / Clear-To-Send</b>
<b>RSSI</b>	<b>Received Signal Strength Indicator</b>
<b>SAP</b>	<b>Service Access Point</b>
<b>SHM</b>	<b>Structural Health Monitoring</b>
<b>SDMA</b>	<b>Space Division Multiple Access</b>
<b>SPAP</b>	<b>Slot Pre-Allocation Protocol</b>
<b>SRSA</b>	<b>Self-oRganizing Slot Allocation protocol</b>
<b>SWAN</b>	<b>Sensor Wireless Application Networks</b>
<b>TDMA</b>	<b>Time Division Multiple Access</b>
<b>WDC</b>	<b>Wireless Data Concentrator</b>
<b>WPAN</b>	<b>Wireless Personal Area Network</b>
<b>WSN</b>	<b>Wireless Sensor Networks</b>

*I would like to thank Dr. Axel Jahn, Dr. Markus Werner and Dr. Matthias Holzbock for giving me the opportunity to develop this project and gain a wide range of precious experiences.*

*Thanks also to my supervisor Eriza Hafid Fazli at TriaGnoSys and to Josep Paradells at UPC for their support and patience.*

*Thanks to all the students with whom I have shared the working room at TriaGnosys. Also thanks to the people that I met at Munich, and specially to Naam.*

*Finally, thanks to my parents, my brother, and my friends, who supported and still support me.*

# Chapter 1

## Introduction.

### 1.1 Motivation.

In the last years the Wireless Sensor Network (WSN) technology has been widely studied and tested. It is an upcoming technology with a huge potential regarding the monitoring of physical variables of a specific environment and providing the data collected to a central entity. In addition, in some cases the same devices may act over other systems and carry out a task in response of a certain event.

The WSN are wireless networks composed by several devices named nodes with processing and transmission properties. The nodes are connected to one or more sensors, and eventually to actuators. The sensors are electronic devices that convert a physical measure magnitude to a electric signal, with the aim to be processed by the node. Some examples of physical magnitudes that can be measured by such sensors are the temperature, the sound, the pressure, the humidity, etc. The actuators are electronic devices with the capacity to perform a task in response of a certain event. Some examples of actuator tasks may be to switch on/off a light, or actuate over a servomotor to move a mechanical piece of a machine.

This technology can be applied in many scenarios such as environmental, buildings, transport, industry, cities, health services, agriculture, public infrastructures, etc. Some concrete application examples are the monitoring of the air quality, measuring the structure health of a bridge or a building, the actuation over a lighting system, or the localization and tracking of mobile devices. The WSN may be isolated networks or may be connected to other networks through a gateway to perform remote access and management tasks. In addition the connection to other networks like Internet allows to share the data collected by the sensors of the WSN.

This thesis is focused on a sub-project named Sensor Wireless Application Network (SWAN) within the German Research and Development project *Luftfahrtforschungsprogramm (LuFo)* that has the objective to study, implement and test WSNs on-board aircrafts.

## 1.2 Background and Scope.

SWAN is a sub-project of the research project Aircraft Total Maintenance Operations, Solutions and Technologies Program-Germany (@MOST-G) funded by the German Ministry of Research, *Bundesministerium für Bildung und Forschung*.

The main objective of all @MOST-G projects is to reduce the time an aircraft has to stand on the land for maintenance and repairs. In particular, the @MOST-G SWAN project investigates the capability and feasibility of integrating Wireless Sensor Networking technology as key enabler to assist and provide a flexible and efficient wireless transport and access network migrating from existing avionics communication data networks and extending today's aircraft systems located in different aircraft domains, namely aerodynamics, engine, cabin and structure. A basic improvement comes with the ability of the sensors to report its status to the whole infrastructure. In addition, WSN ease the design and installation of aircraft systems as well as providing enhanced functionality and reliability for those systems.

The key objectives are to investigate typical systems engineering issues like requirements analysis, feasibility assessment and market analysis, aircraft regulation and certification as well as security, safety and reliability issues required to design an aircraft systems architecture integrating WSN, increasing the efficiency in maintenance and operations.

The main role of TriaGnoSys GmbH in the SWAN project was the design, develop and test of networking software for sensors, data collectors and central server. TriaGnoSys was leading the software development for the network, server development, software testing, and integration of the complete system. Although the Medium Access Control (MAC) protocol for the SWAN project was developed by another partner, TriaGnoSys decided to design and implement a parallel WSN MAC protocol for its own internal experiments with the purpose to assemble an own WSN testbed.

### 1.3 Objectives.

The main objective of this thesis is to study the MAC protocol and implement the MAC sublayer of the SWAN system. In addition, has been considered the design and implementation of a Time Division Multiple Access (TDMA) scheduling algorithm conformed to the SWAN requeriments.

### 1.4 Outline of the thesis.

The thesis is structured in further five chapters. Chapter 2 introduces the SWAN system. In it is explained the system architecture focusing on the wireless network, some proprieties and benefits of WSNs on-board aircraft and finally some application examples.

The Chapter 3 is divided in two blocks. The first block explains some general aspects of the MAC protocol in WSNs, such as the main families of MAC protocols, the MAC protocol operations and the reason to good design a MAC protocol. While in the second block is explained extensively the SWAN's MAC protocol, regarding some general aspects, the addressing, the superframe structure, the frame formats, the MLME commands and the MAC procedures.

In Chapter 4 is explained the implementation of the MAC sublayer within a simulation environment. Basically are explained the technical aspects regarding this implementation, such as some physical layer parameters, the MAC sublayer parameters, the MAC sublayer logic and the implementation of some extensions not considered in the previous specifications.

In Chapter 5 is explained the design and implementation of a TDMA scheduling algorithm conformed to the SWAN system requeriments. This scheduling algorithm has the main characteristics to be contention-free and perfect-periodic, based on power of 2 harmonic periods. In the chapter are explained the basics of the scheduling algorithm, the scheduling algorithm implementation, some analysis, experiments and results, and finally some conclusions and open problems.

Finally, in Chapter 6 are explained the conclusions regarding the overall work done in this thesis.

## Chapter 2

# Sensor Wireless Application Network overview.

### 2.1 Introduction.

The SWAN project has the objective to study and implement WSNs on-board aircrafts. As is usual, all the networks have a specific architecture in which each element of the network has a specific purpose. Although the WSN is focused on the wireless part and its main objective is to perform measures and exchange the data among nodes in a wireless way, these networks may require a backend network, usually a wired network, to support its operations and extend the overall network capabilities. In this chapter is presented a general overview of the whole system architecture, emphasizing in the wireless part.

Moreover, the WSNs have intrinsic properties and benefits; if these systems are applied to a specific scenario, like the aircraft domain scenario, these properties and benefits can be increased and conformed to the application's system requirements. Consequently, in the second subsection of this chapter are presented the main properties and benefits of the WSNs on-board aircrafts.

Finally, are explained some application examples of WSN on-board aircrafts and listed some others to have an overview of the potential of this technology.

### 2.2 System architecture.

The SWAN infrastructure is formed by sensor nodes, wireless data concentrators (WDCs), SWAN server, client applications and human machine interface (HMI). As is shown in



Figure 2.1 the only wireless part of the system is the communication among nodes, the other parts are connected using wires.

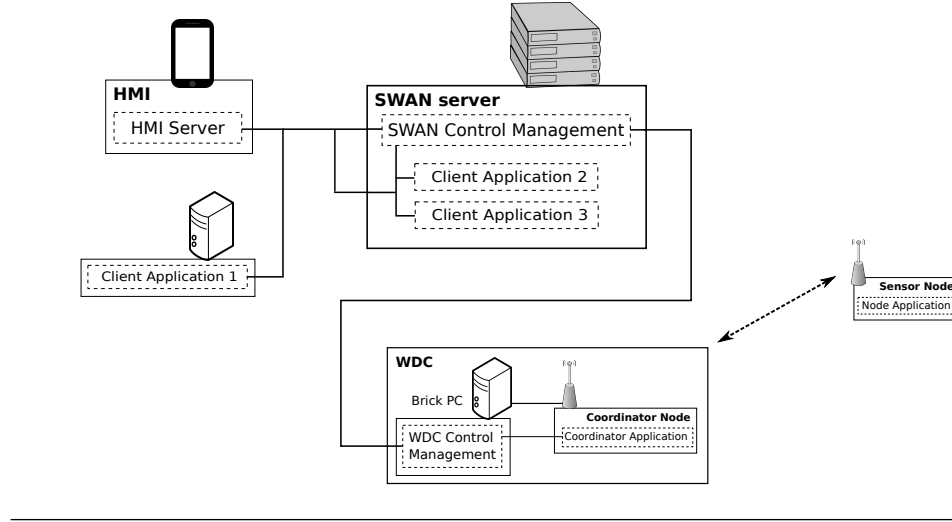


FIGURE 2.1: SWAN architecture.

All the sensor nodes have a unique ID for identification purposes. Sensor nodes communicate with a certain coordinator node which is attached to a WDC by a serial port interface. The WDC collects the data from the sensor nodes through the coordinator node and connects the WSN to the wired backbone network of the aircraft. WDC and coordinator node can be considered as one entity.

All the WDCs are connected to a SWAN server. The purpose of the SWAN server is to control and manage all the components of the WSN. The software implemented in the SWAN server acts as a gate to the data collected by the wired network through the WDCs. The SWAN server also provides an interface to the client applications and the HMI.

The client applications operate with the data collected by the sensor nodes. Also they can send commands to change the state of the actuator nodes. The HMI permits the interaction to display and manage the data in a comprehensible way.

### 2.2.1 Wireless network topology.

The SWAN's wireless network is organized according to a star topology as is shown in Figure 2.2, based on the IEEE 802.15.4 LR-WPAN standard [1]. There are two types of devices, full-function devices (FFD) and reduced-function devices (RFD). All the sensor nodes or RFDs are connected to the central node which is the coordinator node or FFD. Inter-node communication, i.e. the communication among sensor/actuator nodes, is allowed in certain scenarios.

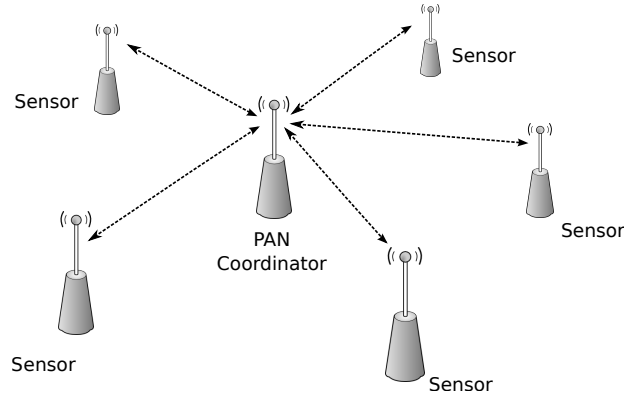


FIGURE 2.2: SWAN wireless network topology.

The advantages of the star topology are the simplicity, no need of routing protocols, the failure of one link only affects one node, it is easy to install and modify, the network density is low and network security can be implemented on the WDC. On the contrary as drawbacks, if the coordinator fails all the cluster is down and it is required a WDC among star-nets.

### 2.2.2 Redudancy.

The redudancy is the duplication of critical components or functions of a specific system which aim is to increase the reliability of that system. The main advantage to integrate a redundant system in any WSN is to ensure the data transmission reliability.

As the wireless topology in SWAN system is the star topology and the main disadvantage of this topology is the failure of the central node, i.e. the coordinator node, the reliability of the network may be increased considerably implementing redudancy in the WDC side. On the other hand, the major drawbacks of implementing redudancy are that the inter-cluster interferences may be increased and the resources may be wasted intensively due the duplication of the transmission and processing of the same data.

There are four possible redudancy scenarios explained in the following list:

- **No redudancy:** In no redundant scenario, no extra coordinator nodes or WDCs are deployed. All the sensor/actuator nodes must associate to the same WDC. In this case the reliability is not improved, maintaining the clasical problem of the failure of the central node in the star network topology.
- **Downlink redudancy:** Downlink redudant scenario means to implement several WDCs into the same cluster as presented in Figure 2.3. It will require a longer

superframe and the share of the downlink part of the superframe among the several WDCs; but if the superframe must be keep in a predefined length, it will require more intelligence on the nodes to differentiate the redundant data, thus extra processing and therefore an increased energy waste.

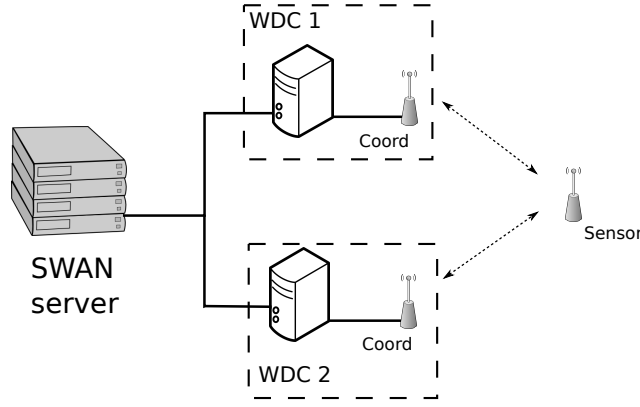


FIGURE 2.3: Basic redundancy scenario in a star topology.

Another option is to attach multiple antennas to the same sensor/actuator node, improving the reliability that the data transmitted from the coordinator node reaches the sensor/actuator node. The signal combination received by the different antennas will require an extra processing and therefore more energy waste.

If downlink redundancy is not desired, the system can keep only one WDC per cluster active with the drawback that the uplink redundancy will be disabled.

- **Uplink redundancy:** The uplink redundant scenario consist of implementing several WDCs within the same cluster as is presented in Figure 2.3, since sensor/actuator nodes can transmit the data to multiple coordinators. The probability of data loss is decreased if the sensor nodes transmit their data to multiple coordinators. These multiple data streams should be combined in the SWAN server. The uplink redundancy is the simpler way to increase the reliability and therefore is preferred.
- **Downlink and Uplink redundancy:** Downlink and uplink redundancy means that the two previous scenarios take place at the same time, hence a data combiner is required in both sides of the wireless network.

### 2.2.3 SWAN working modes.

There are two working modes defined in the SWAN system: *Maintenance mode* and *Operation mode*. The Maintenance mode is runned when the aircraft is on the ground and has the functions of associate and disassociate the sensor/actuator nodes from the

PAN and to configure them; it is based on an hybrid MAC protocol structure, utilizing TDMA during the Downlink period and CSMA/CA during the Uplink period. Operation mode is runned during the flight and it performs the operational services of the network, such as localizing the nodes and the sensing or actuating tasks; it is fully based on TDMA.

*Uplink communication* is the communication from the sensor/actuator nodes to the coordinator node, whereas *Downlink communication* is the communication from the coordinator node to the sensor/actuator nodes.

*Sensing tasks* are the monitoring tasks done by the sensor nodes of physical parameters, such as the temperature monitoring or the smoke detection system. *Actuating tasks* are the operations done by the actuator nodes over other systems of the aircraft, such as the electrical system switching on/off the lights, or the mechanical systems such as the motors controlling the flaps.

#### 2.2.4 Node localization service.

The localization of the sensor/actuator nodes within the aircraft is a key functionality that the system integrates. To know the precise location of each sensor/actuator node has the advantage to be more reliable and fast their maintenance in the cases that some device is not working properly or it is running out of battery.

The localization service is based on Received Signal Strength Indication (RSSI) measurements and it is performed during the Operation mode. A requeriment of the localization service is that all the clusters of the WSN, which cover all the area as is presented in Figure 2.4, have to run using the same frequency band to allow to the localization algorithm calculate the positions of the nodes accurately. The localization algorithm is executed in the SWAN server.

The main disadvantage of this requeriment, since all the clusters of the WSN have to share the frequency channel, is that the overlapping clusters can not use the channel at the same time due inter-cluster interferences. A solution is to multiplex in time the communications of the overlapping clusters. This solution is studied in the scheduling algorithm implemented in Chapter 5.

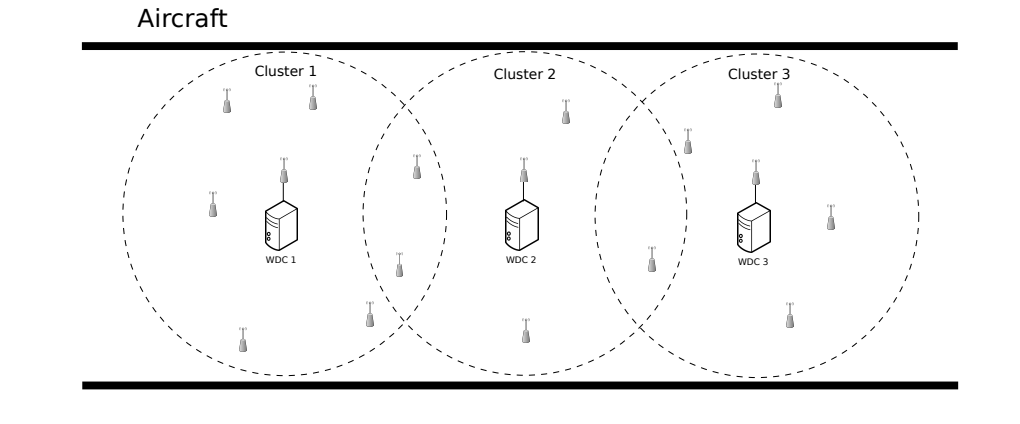


FIGURE 2.4: Multiple clusters covering the whole area.

## 2.3 Properties and benefits of WSNs on-board aircrafts.

In the following points are discussed some features of WSNs that are crucial for increasing the maintenance and operation efficiency in aircrafts.

- **Flexible installation:** wireless technology allows a very flexible and fast network systems installation without the need of wires. It permits for instance to install temporary systems with a little effort. In addition, the temporary installations will become permanent installations in the future years and they will be considered in the aircraft design.
- **Fault detection:** wireless sensors permit the detection of abnormal behaviors prior to any further analysis or fault conditions, delivering the data collected to a central entity which performs further diagnosys and prognosys.
- **Diagnosys and prognosys:** they permit enhanced diagnostic functions for systems by monitoring parameters and conditions which are not captured in current diagnosys due to complexity constraints. Similarly, they permit enhanced prognosys functions involving a multitude of measured parameters having impact on the predicted lifetime of a systems component; in particular, when parameters required for an accurate assessment of the health status of a component need to be collected from many remote locations within the aircraft, e.g. structure health monitoring. This additional information could help in trouble shooting and more quickly finding the root-cause of a problem without generating additional workloads for e.g. unnecessary removal of linings, etc.
- **Ease of servicing:** wireless sensor technology can help to increase the efficiency in many servicing tasks, making them more simple avoiding complex processes

like checking the filling levels of hydraulic or lubrication fluids, or monitoring mechanical systems in hidden areas of the aircraft.

- **Data loading:** permit to extend the access to the aircraft's data network from electronic devices that at the present do not have direct access to it.
- **Localization:** inherently supports localization due to the localized nature of a sensor nodes radio coverage area. If high precision localization is necessary sensor network capabilities can be combined with dedicated localization mechanisms.
- **Identification and configuration:** automatic identification of an aircraft entity is very advantageous for an efficient maintenance process; WSN can provide the necessary identification means e.g. via smart sensor tags or micro sensor nodes. WSNs can also be used to transport configuration-related data (e.g. part-number, functional identification number, equipment serial number, software/firmware version number, etc.).
- **Relation to RFID technology:** WSN technology does not only provide the means to wirelessly identify items, but also to transfer information of any type over larger distances than those covered by RFID technology. WSN and RFID technology can be combined and allow for smart applications. This in particular holds true whenever the status of removable items needs to be tracked in an automated way. For these examples, smart, lightweight and low cost RFID tags can be placed on the removable item. These RFID tags are interrogated by means of a WSN node equipped with RFID reading capabilities. Hence, RFID and WSN technologies are complementing each other.

## 2.4 WSN applications on-board aircrafts.

A description of various maintenance and system-related wireless sensors applications examples are explained in the following subsections. Some of these applications are an adaptation from actual implementations to wireless sensors ones, other ones are completely new applications.

### 2.4.1 Flight test installation.

Flight Test Installation (FTI) is a major component of the development and certification process of an aircraft program. The today's aircraft measurement campaigns requires a huge amount of analog/discrete wired sensors in order to measure physical magnitudes such as temperature, humidity or stress. Wired sensors installation sometimes is very

complicated or even impossible, like the installation of wired sensors outside the aircraft's fuselage. Moreover, any cabling required for sensor measurements has a direct impact on the aircraft domain, i.e. aerodynamics, structure, cabin, engines, etc.

Using WSN permits to install sensors in all the compartments of the aircraft without need of wires, therefore with a reduced installation effort. The inspections will be easier due the sensors can be setup by mobile wireless devices such as PDAs or laptops. In conclusion, the time-consuming planning of measurement campaigns can be severely reduced.

### **2.4.2 Aircraft structure health monitoring.**

Structural Health Monitoring (SHM) describes the continuous and autonomous monitoring of defects, stress/strain, environmental and flight parameters by means of permanently attached or embedded sensor systems to ensure the structural integrity of the aircraft.

Nowadays, no SHMs systems are deployed on wings, fuselage or other structures of the aircraft. To ensure the today's structural integrity, interval-based checks are performed on ground by maintenance personnel.

Using wireless sensor networks for SHM Systems promises a flexible and efficient approach because no wiring harness, for example on wings, is required to connect the sensors. Furthermore, it allows deploying sensors with less installation restrictions and simplifies the integration into the structure. In addition, it derives an efficient-predictive maintenance scheduling of the structural integrity to prolong the operation.

### **2.4.3 Fire and smoke detection systems.**

Fire and Smoke Detection systems are installed in the pressurized and in the unpressurized areas of the aircraft. These systems have two main functions: fire/smoke detection and fire extinguishing.

The today's approach is a redundant complex wired system composed by smoke/fire sensor detectors and the control system. The use of WSNs permits the communication between these different parts to be wireless and substitute the actual smoke/fire detectors by new ones.

The main benefits are a reduced requirement of wired installations, the reduction of weight and size of smoke detectors and the localization of these devices.

#### **2.4.4 Proximity switches on PAX seat.**

Passenger seats should be safe and comfortable to satisfy the passengers physical health and serve the passengers needs while travelling over long distances. So called proximity switches can be used to meet these demands. Proximity switches can be foreseen to detect the health status of arm rests or seat belts, PAX presence, or table positions.

Nowadays, visual inspections are performed manually by maintenance personal. From the maintenance perspective, it might be a very time-consuming procedure to accomplish inspections on larger aircrafts like the A380 with nearly 500 passenger seats on two decks.

Integrating wireless sensor nodes as proximity switches into the seats reduces the maintenance inspection times. Proximity switches detect the seat status and wirelessly transmit this information to a main controller or to notify the crew/maintenance personal.

#### **2.4.5 Other applications.**

In here are listed some other applications of WSNs on-board aircrafts.

- High Lift System.
- Temperature Monitoring.
- Floor Panel Heating.
- Water Lines Ice Protection.
- Lighting Control System.
- Emergency Lighting System.
- Standalone Identification System.
- Doors Surveillance System.
- Discrete Key Lines.
- Equipment Identification and Localization.
- Passenger Localization and Tracking.
- Landing Gear System.
- Landing Gear Health Monitoring System.
- Galley Network Power Management.



- Engine Pylon Structural Monitoring.
- Passenger Oxygen Mask Activation.
- Tuneable Vibration Actuation System.
- Electrical Cargo Loading System.

## Chapter 3

# Medium Access Control protocol in WSNs.

### 3.1 Introduction.

The Medium Access Control protocol is a sublayer of the data link layer, the second layer of the seven-layer OSI model. It provides addressing and channel access control mechanisms allowing several devices that use a shared medium communicate among them within a multiple access network. The MAC sublayer acts as an interface between the logical link control (LLC) sublayer and the network's physical layer (PHY).

The IEEE 802.15.4 standard specifies the MAC sublayer and the PHY layer for Low-Rate Wireless Private Area Networks (LR-WPAN), addressed to low-data rate, low-power consumption and low-cost wireless networking, like the WSN devices [1]. This specification is the base from which the SWAN's MAC sublayer is designed and implemented.

#### 3.1.1 Main families of MAC protocols.

Essentially there are two main families for regulating access to the wireless medium: contention-based and reservation-based approaches, each of them with their own advantages and drawbacks. Any MAC protocol implementation is based on one of those two approaches or a combination of both [2].

The reservation-based approach requires the knowledge of the network topology to establish a schedule in the communication among nodes. The objectives of this schedule, which make use of dedicated resources in node transmissions, may be ensure fairness,

reduce collisions or assure a minimum bandwidth, by avoiding the transmission of different nodes at the same time. This approach assumes that nodes are synchronized in time, normally achieved by using beacon frames. The more representative example of this family is the TDMA protocol.

The contention-based approach is less complex than reservation-based protocols. In this approach is not needed the scheduling nor the network knowledge, so the nodes have no allocated communication resources. On the contrary, the wireless nodes compete for the channel access in a probabilistic way without the need of a centralized control mechanism. The more representative example of this family is the CSMA mechanism.

### **3.1.2 MAC operations.**

In WSN, the MAC sublayer handles all access to the physical radio channel and is responsible for the following tasks:

- Generate network beacons if the device is a coordinator.
- Synchronizing to the network beacons.
- Support PAN associations and disassociation.
- Supporting device security.
- Employ a defined mechanism for channel access, i.e. CSMA, TDMA, FDMA, etc.
- Provide a reliable link between two peer MAC entities.

### **3.1.3 Benefits and properties of designing correctly a MAC protocol.**

The main role of MAC protocol is to coordinate the access to the shared medium which in WSN is the wireless medium. As this medium is inherently broadcast in nature and therefore prone to interferences, a well-designed MAC protocol is essential in this kind of networks. In this way the interferences and hence the packet collisions will be avoided, improving the reliability and preventing retransmissions.

Another reason to have a well-designed MAC protocol is the constrained hardware on which the WSN nodes are implemented and particularly the limited batteries. These nodes are powered by small batteries that shall operate for a long time, sometimes several years, without human intervention. The main source of energy consumption is the radio device. Therefore with a good MAC protocol, controlling the active and

sleeping periods of the transceiver, the energy consumption could be reduced drastically and hence increase the lifetime.

A particularity is that each application require a different MAC implementation due different requirements of that applications, the type of network control (centralized or decentralized), the type of data exchanged, the type of links, the number of nodes, the mobility of the nodes, the expected lifetime, etc. The MAC protocol needs a trade-off among energy consumption (longevity), reliability, fairness, scalability, latency, mobility, security and throughput.

The following effects impact in energy consumption of each device and they must be avoided as far as possible by the correct MAC protocol design.

- **Collisions:** they happen when two or more nodes within the same range transmit simultaneously using the same frequency band. The energy used to do a useful transmission is simply wasted and also the nodes have to spend more energy to do the packet retransmission. The avoidance of collisions is achieved in reservation-based MACs, while contention-based MACs require a deeper study.
- **Overhearing:** it happens when nodes receive (listen) irrelevant packets, for example unicast packets destined to other nodes, or signals like preambles. It can be avoided through header filtering of received packets.
- **Overhead:** protocol overhead is a energy wasting source. An example is RTS/CTS control packets used in some contention-based protocols. It can be avoided using protocols which do not use RTS/CTS control packets.
- **Idle listening:** usually a sensor node does not know when it will receive a packet, so it should maintain the radio in reception mode waiting for that packet and therefore wasting energy. A way to avoid this is to put nodes to sleep as long as possible while avoiding deafness.
- **Redundant data:** it happens when a node receives twice the same data from the same point or from two different points of the network.

### 3.2 Medium Access Control protocol in SWAN.

The MAC protocol designed for the SWAN project is an hybrid protocol, i.e. it uses scheduling-based access method in some periods of the superframe and contention-based in other periods. It is fully based on Beacon-enabled operational mode [1].

### 3.2.1 General aspects.

The MAC in SWAN is mainly responsible of the following tasks:

- Beacon generation if the device is a PAN coordinator.
- Synchronization of the transmissions.
- Provide a reliable data link.
- Support associations and disassociations.
- Provide RSSI measurements.

The requirements of the MAC layer are the following:

- Guaranteed and collision free data transmission, achieved by TDMA.
- Possibility to make the network more scalable, provided by CSMA/CA.
- The amount of sensor may be varying, attained by a dynamic time slot assignment.

There are two MAC sublayer services that are accessed through two Service Access Points (SAP):

- MAC Common Part Sublayer (MCPS): supports the transport of operation commands between peer application entities.
- MAC Sublayer Management Entity (MLME): provides the service interfaces through which layer maintenance functions may be invoked.

### 3.2.2 Addressing.

The system supports broadcast, multicast and unicast addressing modes. Broadcast addressing is a transmission from a PAN coordinator to all sensor/actuator nodes in the network. Multicast addressing means transmission for a certain group of sensor/actuator nodes. Unicast addressing is a transmission to only a single sensor/actuator node.

Each sensor/actuator node has a short and a long address, i.e. 16-bit and 64-bit addresses respectively. The long addresses, i.e. 64-bit addresses, are used during Maintenance mode; on the other hand, the short addresses, i.e. 16-bit addresses, are used during the Operation mode and they are provided when the node associates.

Moreover, each device in the network, i.e. PAN coordinator and sensor/actuator nodes, have a unique 16-bit PAN identifier. It allows the communication between sensor/actuator nodes within a group and is chosen depending on the sensor application. The PAN ID of the coordinators is 0x0000.

### 3.2.3 Superframe structure.

There are three well defined periods within a superframe, namely Downlink period, Uplink period and Inter-node link period. During Downlink period are performed the communications from the PAN coordinator to the sensor/actuator nodes. Within Uplink period are performed the communications from the sensor/actuator nodes to the PAN coordinator. In Inter-node link period, are carried out the communications among sensor/actuator nodes.

The previous periods can be split in two divisions, contention-free period (CFP) and contention-access period (CAP). During CFP, TDMA is carried out by a scheduling algorithm determined by the SWAN server which provides deterministic communications. The frame slots are assigned to the nodes with the objective to transmit their packets achieving a low latency, delivering guarantee and fairness among nodes, but as drawback the low flexibility of the TDMA schedules. During CAP, CSMA/CA algorithm is executed accessing to the channel in a non-deterministic way, therefore the packets can not accomplish delivering rates, low latency and fairness as in TDMA, being more energy consuming for the node due the non optimal utilization of the transceiver, with the advantage of scalability due its flexibility.

In addition, there are the Beacon frames due it is a MAC protocol based on beacon-enabled operational mode. The main functions of the beacon frames are to provide the synchronization of the network, to provide the superframe structure to sensor nodes, coordinate inter-node communications, provide the network timing by timestamps and they are addressed to the broadcast PAN ID 0xFFFF.

The acknowledgement of the data transmitted is an optional feature. Acknowledgement transmission provides reliable communications. As the superframe is designed, each acknowledgement has to be transmitted within a different slot and period from the ones that the packet that should be acknowledged has been transmitted; this is different from the IEEE 802.15.4 standard which says that the acknowledgement transmission should

start a little period after the frame reception<sup>1</sup>. Therefore, the requested acknowledgements of the frame received will use capacity of the superframe.

Finally, there are two superframe structures due the two working modes, i.e. Operation mode superframe and Maintenance mode superframe.

### 3.2.3.1 Maintenance mode superframe.

In maintenance mode the association and disassociation of sensor/actuator nodes is allowed, which define the number of sensor and actuator nodes that will be active in the operation mode and therefore configure the number of timeslots needed in this last period. In maintenance mode also is carried out the configuration of the sensor/actuator nodes. It is run in land.

Figure 3.1 illustrates the superframe structure of the maintenance mode. The superframe is delimited by two beacons, one at the start and one at the end corresponding to the one of the next superframe, plus a gap corresponding to the Interframe Spacing (IFS). It is formed by a downlink CFP and uplink CAP periods. The downlink period is used for transmitting Disassociation Requests, Association Responses and the ACKs of the frames received during the uplink period that request acknowledgement. The uplink period is used to transmit the Association Requests and the ACKs of the frames received during the downlink period that request acknowledgement.

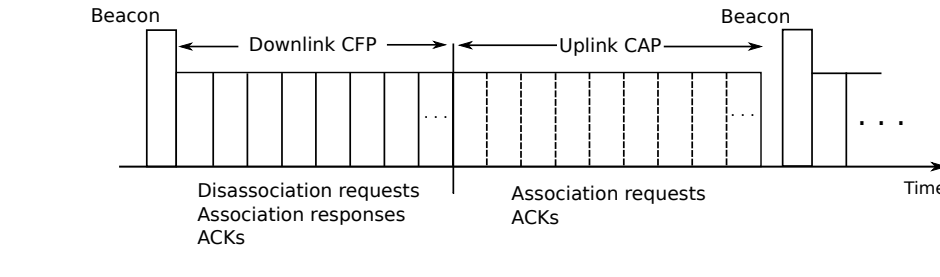


FIGURE 3.1: Superframe of the maintenance mode.

### 3.2.3.2 Operation mode superframe.

In operation mode is performed the data exchange between the sensor/actuator nodes and the coordinator node. Operation mode also is responsible of enabling the RSSI

<sup>1</sup> "The transmission of an acknowledgment frame in a nonbeacon-enabled PAN or in the CFP shall commence *macSIFS*Period after the reception of the last symbol of the data or MAC command frame. The transmission of an acknowledgment frame in the CAP shall commence either *macSIFS*Period after the reception of the last symbol of the data or MAC command frame or at a backoff period boundary."

measurement procedure to be used by the node localization service. It is run during the flight.

Figure 3.2 illustrates the superframe structure of the operation mode. The superframe is also delimited by two beacons. It is fully scheduled according to TDMA. During downlink CFP period are transmitted Disassociation Requests, the Data from the actuator nodes and the ACKs of the sensor node's frames. The uplink CFP period is used to transmit the Data from the sensor nodes and the ACKs of the actuator node's frames. The Inter-node link CFP period is used for transmitting RSSI broadcast and RSSI vector frames.

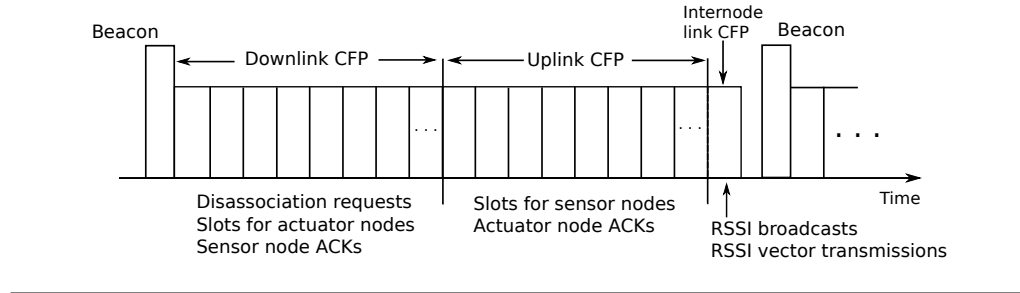


FIGURE 3.2: Superframe of the operation mode.

### 3.2.4 Frame formats.

#### 3.2.4.1 MAC frame.

In Figure 3.3 is illustrated the MAC Protocol Data Unit (MPDU) or MAC frame. The MAC frame follows the IEEE 802.15.4 specification, and it is composed by a MAC Header (MHR), a MAC Service Data Unit (MSDU) or MAC payload and a MAC Footer (MFR).

In the Frame Control field are defined the beacons and maintenance commands by MAC COMMAND frame type, and operation commands by DATA frame type. Also in this field is indicated if an acknowledgement is required from the recipient of the MAC frame.

The addressing fields will be set to 16-bit short address (2 bytes) or 64-bit extended address (8 bytes) in function of the working mode. For beacon and command frames sent by the PAN coordinator, the PAN ID Compression subfield will be set to one and the frame will contain only the Destination PAN ID field (the Source PAN ID field will be assumed equal to that of the destination). The PAN coordinator will always use 16-bit short source address. However, sensor nodes will use 64-bit extended address if the running mode is maintenance mode or 16-bit short address if operation mode.



The Sequence Number field specifies the sequence identifier for the frame. The Frame Check Sequence (FCS) field is calculated in function of the MHR and MAC payload parts.

Field	Frame Control	Sequence Number	Destination PAN ID	Destination Address	Source PAN ID	Source Address	Frame Payload	FCS
Octets	2	1	2	2/8	2	2/8	[0:N]	2
<b>MHR</b>							<b>MSDU</b>	<b>MFR</b>

FIGURE 3.3: MAC frame.

### 3.2.4.2 Beacon frame.

The beacon frames rely on the same format as the MAC frames and include the information of the superframe structure and the current time of the TDMA-SWAN. The Source and Destination Address fields will be set as 16-bit short address. The Destination PAN ID and Destination Address fields will contain the broadcast address 0xFFFF. The Source PAN ID field will be not included and the Source Address field will contain the 16-bit address of the PAN coordinator. The Beacon payload is formatted as illustrated in Figure 3.4 and in Table 3.1 are explained each one of the beacon payload fields.

Field	Beacon ID	# DL CFP slots	# UL CFP slots	# UL CAP slots	#Internode link slots	Slot Duration	Timestamp	Internode link node	Internode link options
Octets	1	1	1	1	1	2	4	2	1
<b>Beacon Payload</b>									

FIGURE 3.4: Beacon payload.

### 3.2.4.3 Maintenance command payload.

The MLME commands are sent during maintenance and operation modes. They are better explained in the MLME commands subsection 3.2.5. In Figure 3.5 is illustrated its format.

Field	Maintenance Command ID	Payload
Octets	1	[0:N]
<b>Maintenance Command Payload</b>		

FIGURE 3.5: Maintenance command payload.

Field	Description
Beacon ID	Superframe type: 0x01 Operation mode superframe 0x02 Maintenance mode superframe 0x03 Ranging measurement beacon
Number of downlink CFP slots	Number of downlink CFP slots defined for the TDMA superframe.
Number of uplink CFP slots	Number of uplink CFP slots defined for the TDMA superframe. It only takes effect during the Operation mode.
Number of uplink CAP slots	Number of uplink CAP slots defined for the TDMA superframe. It only takes effect during the Maintenance mode.
Number of internode-link CFP slots	Number of internode-link CFP slots defined for the TDMA superframe. It only takes effect in Operation mode.
Slot duration	Duration of the time slot in seconds in the TDMA superframe.
Timestamp	The current time in seconds of the TDMA-SWAN.
Internode-link node	16-bit address of the sensor node allowed transmitting in the internode-link CFP.
Internode-link options	Options defined for the transmission in the internode-link CFP.

TABLE 3.1: Beacon payload fields.

#### 3.2.4.4 Operation payload.

The operation payload is only sent during operation mode and its function is the interchange of data between coordinator and sensor/actuator nodes. It is illustrated in Figure 3.6.

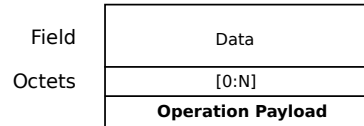


FIGURE 3.6: Operation command payload.

#### 3.2.5 MLME commands.

The MLME commands are sent in the MSDU or MAC payload. There are two types of MLME commands, the ones sent from coordinator node to sensor/actuator node and

the ones sent from the sensor/actuator nodes to the coordinator node. In Figure 3.7 are illustrated these commands.<sup>2</sup>

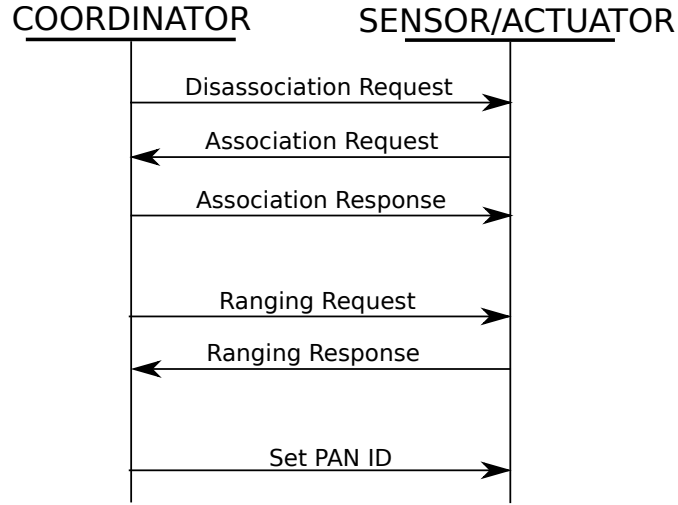


FIGURE 3.7: MLME commands exchanged in the wireless network.

The Beacon Mode determines the working mode that the system is running at the moment, i.e. maintenance mode or operation mode. MLME beacon mode corresponds to maintenance mode and MCPS beacon mode corresponds to operation mode. Therefore, MLME commands can be sent during maintenance and operation modes.

The acknowledgement of the MLME commands is an optional feature and it has to be considered during the MAC layer implementation.

### 3.2.5.1 From coordinator to sensor/actuator.

In Table 3.2 are listed the MLME commands sent by the coordinator node to the sensor/actuator nodes.

MLME command identifier	MLME command name	Beacon Mode
0x06	Association response	MLME
0x04	Disassociation request	MLME and MCPS
0x07	Ranging request	MLME and MCPS
0x0F	Set PAN ID	MLME and MCPS

TABLE 3.2: MLME commands from coordinator to sensor/actuator.

<sup>2</sup>The MLME commands exchanged within the Internode link period corresponding to the RSSI measurements are not considered in this thesis.

**Association response.**

The Association response payload is illustrated in Figure 3.8. Its function is to send to the sensor/actuator node the response of an Association request command during the association to the PAN procedure. This command is always sent to an unicast address corresponding to the MAC address of a sensor/actuator node. The first field transports the command ID and the second field transports the association response data. The association response data may be:

- **0xFFFE**: the association to the PAN is temporary not allowed but the sensor/actuator node can keep trying on associate.
- **0xFFFF**: the association is not allowed and the sensor/actuator node will be set to a still state.
- **New Associated Slot**: the association is allowed and it shall be acknowledged. The **New Associated Slot** is the slot position within the TDMA superframe that the node will occupy during the operation mode.

Field	MLME command	Association Response
Octets	1	2
<b>MLME frame payload</b>		

FIGURE 3.8: Association response payload.

**Disassociation request.**

The Disassociation request payload is illustrated in Figure 3.9. Its goal is to disassociate a group of sensor/actuator nodes or a specific sensor/actuator node from the PAN. It can be send to the broadcast or an unicast address, but it is being studied the possibility to send it to a multicast address. The first field is the command ID and the second field is the disassociation setting:

- **0xFFFE**: the sensor/actuator node will next attempt an Association request.
- **0xFFFF**: the sensor/actuator node will be set to a still mode.

**Ranging request.**

The Ranging request payload is illustrated in Figure 3.10. Its function is to request a ranging measurement from a sensor/actuator node. It is send to an unicast address.

Field	MLME command	Disassociation Setting
Octets	1	2
<b>MLME frame payload</b>		

FIGURE 3.9: Disassociation request payload.

The command includes the settings to configure and perform the ranging procedure. After receiving a Ranging request command, the coordinator node sets automatically a special wireless frame to perform a phase ranging measurement among the target sensors. When this process finishes, the coordinator sets the wireless frame back to the previous settings. This lasts at least two superframes during which no other commands are transmitted over the wireless link.

Field	MLME command	Antenna	Frequency step	Frequency start	Frequency stop	Number of frequencies	Number of PMUs
Octets	1	1	1	2	2	1	1
<b>Beacon Payload</b>							

FIGURE 3.10: Ranging request payload.

### Set PAN ID.

The Set PAN ID payload is illustrated in Figure 3.11. Its objective is to assign to a sensor/actuator node a new PAN ID, that is to associate the sensor/actuator node to another PAN different from the current. It is transmitted in unicast and shall always be acknowledged. The command payload is the ID of the new PAN.

Field	MLME command	New PAN ID
Octets	1	2
<b>MLME frame payload</b>		

FIGURE 3.11: Set PAN ID payload.

#### 3.2.5.2 From sensor/actuator to coordinator.

In Table 3.3 are listed the MLME commands sent by the sensor/actuator nodes to the coordinator node.

#### Association request.

The Association request payload is illustrated in Figure 3.12. Its function is to try the association to a PAN by a sensor/actuator node. It can only be sent by an already

MLME command identifier	MLME command name	Beacon Mode
0x05	Association request	MLME
0X08	Ranging response	MLME and MCPS

TABLE 3.3: MLME commands from sensor/actuator to coordinator.

unassociated device. The Ranging in association field is a boolean value and indicates if the device is able to handle a ranging procedure. If this value is set to false the rest of fields are ignored, otherwise the field Antenna diversity is also a boolean value indicating whether the measurement included in the command was performed with one (false) or two (true) antennas. In case of true, both consecutive fields shall be taken into account and include the result of the measurement, otherwise only the content of Measurement Antenna 1 field is valid.

Field	MLME command	Ranging in Association	Antenna diversity	Measurement Antenna 1	Measurement Antenna 2
Octets	1	1	1	4	4
<b>MLME frame payload</b>					

FIGURE 3.12: Association request payload.

### Ranging response.

The Ranging response payload is illustrated in Figure 3.13. Its objective is to send the results of a ranging procedure done by a sensor/actuator node to the coordinator when a phase-based ranging process is finished. The Antenna Diversity field is a boolean value indicating with false that the resulting distance value is provided only in Distance Antenna 1 field or with true that the resulting distance value is provided by both fields, Distance Antenna 1 and Distance Antenna 2.

Field	MLME command	Antenna diversity	Distance Antenna 1	Distance Antenna 2
Octets	1	1	4	4
<b>MLME frame payload</b>				

FIGURE 3.13: Ranging response payload.

## 3.2.6 MAC procedures.

### 3.2.6.1 Association.

The sensor/actuator nodes are able to associate only if first it is done a disassociation, i.e. a MAC sublayer reset. The MAC sublayer of an unassociated sensor/actuator node

will initiate the association procedure sending an Association Request command to the PAN coordinator. As optional feature, the coordinator will confirm this reception sending an ACK frame. If the Association Request is not well sent, the node will attempt again in the next superframe. Finally, the higher layers of the stack make the decision of the association but the MLME generate the Association Response command and the sensor node responds mandatory with an ACK to the Association Response. In Figure 3.14 is illustrated the association procedure.

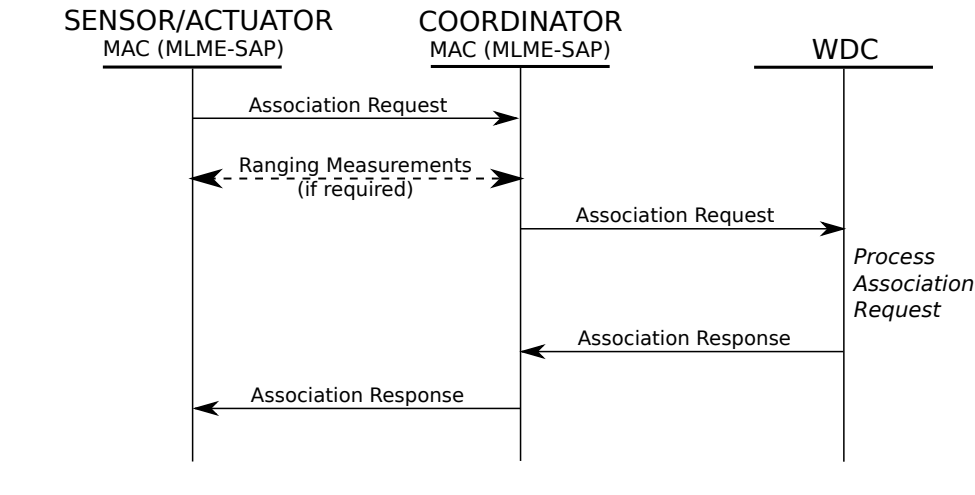


FIGURE 3.14: Sequence chart of the association procedure.

### 3.2.6.2 Disassociation.

It is initiated by the higher layers of the PAN coordinator, issuing a Disassociation Request to the MLME. If the Disassociation Request is not broadcast, the node who received it will send back an ACK. If transmission fails, MAC sublayer will notify to the higher layers of the PAN coordinator. In Figure 3.15 is illustrated the disassociation procedure.

### 3.2.6.3 Ranging measurements.

The ranging measurements are defined by the MLME-SAP service. The ranging procedure shall be initiated by the next higher layers of the PAN coordinator in Operation mode, or by the MLME itself if requested during an association procedure in Maintenance mode. It allows estimating the real physical distance between the PAN coordinator and the sensor node using phase measurements. The procedure, illustrated in Figure

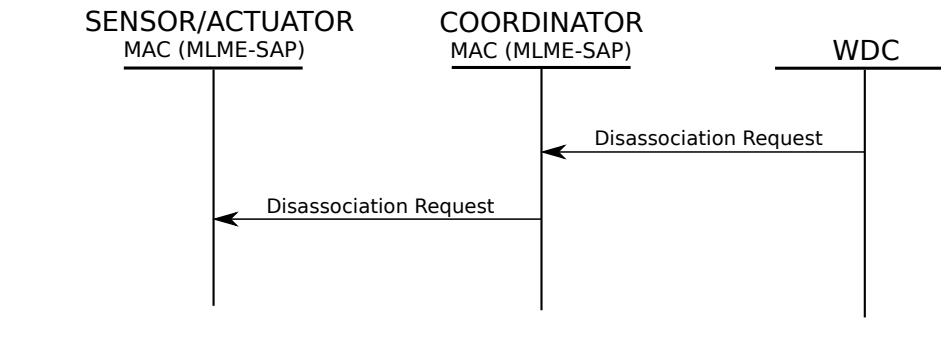


FIGURE 3.15: Sequence chart of the disassociation procedure.

3.16, is performed in a special superframe structure defined for this purpose and accordingly to the ranging measurement settings. In this superframe, illustrated in Figure 3.17 only the requested sensor will be allowed to transmit.

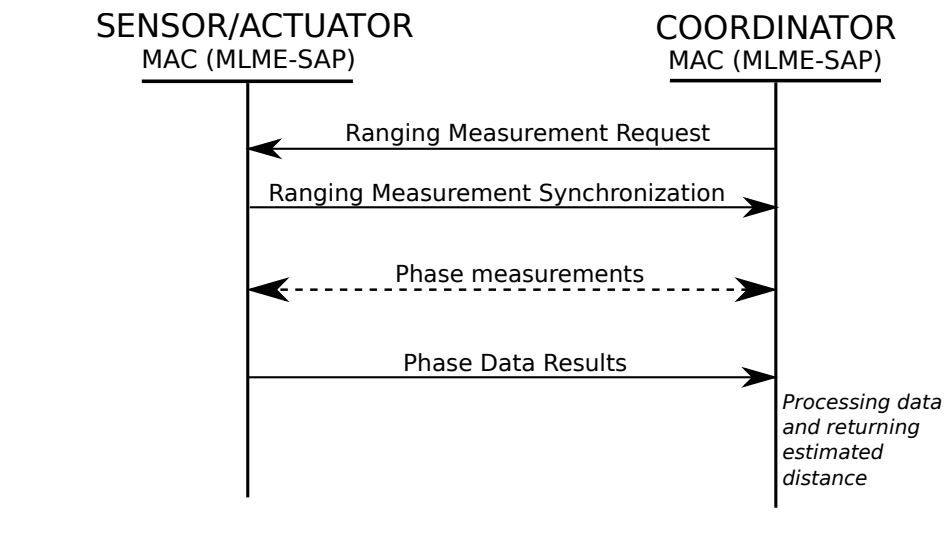


FIGURE 3.16: Sequence chart for ranging measurement.

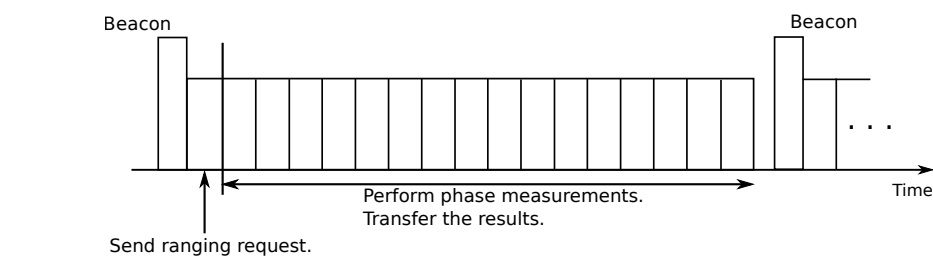


FIGURE 3.17: Superframe structure during ranging superframe.



## Chapter 4

# MAC sublayer: Maintenance Mode implementation.

### 4.1 Introduction.

The Maintenance Mode is one out of two SWAN operating modes. In Maintenance Mode the nodes can associate and disassociate to the PAN network and also they may be configured. In addition, the number of sensor and actuator nodes that will be operative during the Operation Mode are defined during this period, which is essential to schedule the messages of these nodes into the Operation Mode superframe.

Maintenance Mode is based on an hybrid protocol structure, combining TDMA, i.e. scheduled-based protocol, during downlink period and CSMA/CA, i.e. contention-based protocol, during uplink period. The Maintenance Mode can be divided into three different periods, namely beacon period, downlink CFP period and uplink CAP period (see [3.2.3.1](#)).

The beacon period is based on the transmission of the beacon frames. The beacon frames allow the synchronization of the nodes contained within the coordinator's range, and the transmission of the superframe structure information to that nodes (see [3.2.4.2](#)).

In the downlink period the coordinator sends Dissasociation Requests and Association Response frames to the sensor/actuator nodes within its coverage area. It is fully scheduled using TDMA and this scheduling is done in the transmitter side by defining the time duration of a slot. In each slot only one MAC frame can be sent. By self design definition, the Disassociation Request is sent to the broadcast address within the first slot and in the other slots are sent the Association Responses in unicast and the acknowledgments if they are required. The sensor/actuator nodes shall be listening the channel

during the whole downlink period, due they do not know when a packet is addressed to them. A sensor/actuator node only will be in sleep mode during the downlink period if it has received a frame during the actual downlink period, it is already associated to the PAN, or it has received a 'set to still mode' command. The number of slots during this period should be defined in function of the expected number of requests received during each uplink CAP period plus the expected number of possible ACKs.

In the uplink period the sensor/actuator nodes send Association Request frames and acknowledgments to the PAN coordinator. As this period is a contention-based period, the sensor/actuator nodes compete for the access to the wireless medium. The coordinator node is listening the channel during the duration of the whole uplink period to receive the frames sent by the sensor/actuator nodes. A sensor/actuator node only sets its transceiver to sleep state during this period if it has completed properly the association to the PAN. The length of this period is defined considering a wanted association rate, i.e. how many sensor/actuators can perform its association to the PAN in one superframe period. The length of the uplink period must be larger than the length of the downlink period due the inefficient nature of the CSMA/CA algorithm.

In the following subsections are explained the main concepts regarding the MAC sublayer implementation in general and the Maintenance Mode in particular. First is explained the platform in which the implementation has been carried out, then the parameters which define the PHY layer and the MAC sublayer, later the MAC sublayer logic during the Maintenance Mode and finally some implementation extensions.

## 4.2 Implementation platform.

The implementation of the MAC sublayer's Maintenance Mode is done under Castalia simulator [3], which is based on the OMNeT++ platform. Castalia is a simulator for Wireless Sensor Networks, Body Area Networks (BAN) and generally networks of low-power embedded devices. The main features are the following:

- Advanced channel model based on empirically measured data.
- Advanced radio model based on real radios for low-power communication.
- Extended sensing modelling provisions.
- Node clock drift, CPU power consumption.
- Highly parametric.
- Realistic node behavior.

- Designed for adaptation and expansion.

This simulator is not useful to do sensor platform-specific simulations, instead it is propitious to implement and validate an algorithm before moving to implementation on a specific sensor platform.

Omnet++ is an object-oriented modular discrete event network simulation framework with the following main characteristics:

- It is based on component architecture simulation models.
- Models are assembled from reusable components named modules.
- Modules can be connected among them via gates and they communicate by message passing.
- The module behavior and/or module topology can be customized and parametrized by module's parameters.
- Programed in C++ language.

These simulation platforms work under the event-driven programming paradigm. It is in this way because the behavior of the computer networks is perfectly modeled by asynchronous event generation. In the event-driven programming the classic programming style, based on sequential commands, is not allowed. The event-driven programming fundamentals are the following:

- Event: action that usually is initiated outside the scope of a program and is handled by a piece of code inside the program.
- Event generator: object where the event occurs.
- Event handler: object that will perform a task when the event occurs. May be several event handlers for a given event, each handler may need to do something different when the event occurs.
- Registration service: the event generator remember who all of the registered events handlers are.
- Finite state machine: used to structure the program flow.

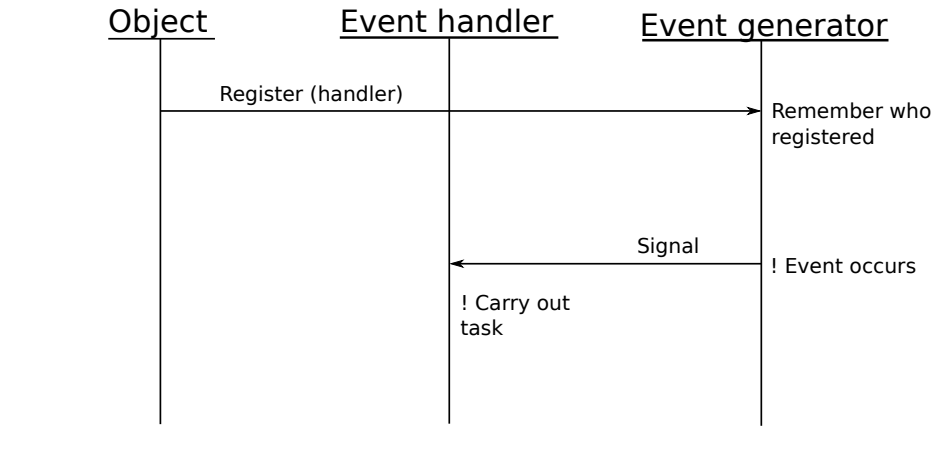


FIGURE 4.1: Event-driven programming paradigm schema.

### 4.3 Physical layer parameters.

The PHY 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 standard offers three operational frequency bands: 2.4 GHz, 915 MHz and 868 MHz. The data rate is in function of the operational frequency, and it is equal to 250, 40 or 20 kbps respectively. All the frequency bands are based on the Direct Sequence Spread Spectrum (DSSS) spreading technique.

The main tasks of the physical layer are the following:

- **Activation and deactivation of the radio transceiver:** in function of the radio transceiver state. The radio transceiver state may be *transmitting*, *receiving* or *sleeping*.
- **Energy Detection (ED) within the current channel:** it is an estimation of the received power signal within the bandwidth of the channel used. The duration should be equal to 8 symbol periods. It is used by network layer algorithm to select the channel or by the link layer to perform a Clear Channel Assessment.
- **Link Quality Indication (LQI):** characterizes the Strength/Quality of a received packet and measures the quality of a received signal on a link.
- **Clear Channel Assessment (CCA):** is responsible for reporting the medium activity state: busy or idle. It is performed in three operational modes: Energy Detection mode, Carrier Sense mode and Carrier Sense with Energy Detection mode.
- **Channel Frequency Selection:** the physical layer should be able to tune its transceiver into a specific channel if it is requested by an higher layer.

During the simulation the transceiver CC2420 of Texas Instruments is used. In fact any transceiver may be used if it is configured within a specific configuration file. The configuration file of the transceiver used is presented here:

---

```
# *****
# * Copyright: National ICT Australia, 2009 - 2010 *
# * Developed at the ATP lab, Networked Systems research theme *
# * Author(s): Athanassios Boulis, Yuriy Tselishchev *
# * This file is distributed under the terms in the attached LICENSE file. *
# * If you do not find this file, copies can be found by writing to: *
# * *
# * NICTA, Locked Bag 9013, Alexandria, NSW 1435, Australia *
# * Attention: License Inquiry. *
# * *
# *****/

RX MODES
# Name, dataRate(kbps), modulationType, bitsPerSymbol, bandwidth(MHz),
# noiseBandwidth(MHz), noiseFloor(dBm), sensitivity(dBm), powerConsumed(mW)
normal, 250, PSK, 4, 20, 194, -100, -95, 62
IDEAL, 250, IDEAL, 4, 20, 194, -100, -95, 62

TX LEVELS
Tx_dBm 0 -1 -3 -5 -7 -10 -15 -25
Tx_mW 57.42 55.18 50.69 46.2 42.24 36.3 32.67 29.04

DELAY TRANSITION MATRIX
# State switching times (time to switch from column state to row state, in msec)
# RX TX SLEEP
RX - 0.01 0.194
TX 0.01 - 0.194
SLEEP 0.05 0.05 -

POWER TRANSITION MATRIX
# RX TX SLEEP
RX - 62 62
TX 62 - 62
SLEEP 1.4 1.4 -

SLEEP LEVELS
idle 1.4, -, -, -, -
```

---

There are two RX modes, **normal** and **IDEAL**; the parameters configured in each one of the RX modes are: data rate, modulation type, bits per symbol, bandwidth, noise bandwidth, noise floor, sensitivity and power consumed. There are eight TX power levels: 0, -1, -3, -5, -7, -10, -15 and -25 dBm. There are two transition matrices referring to the switching of the transceiver state, one about delay and other one about power. In addition, there are other physical parameters referring to the PHY

layer which are defined in the simulation's configuration file, these parameters are: `phy layer overhead`, `delay for valid CS` and `guard time`.

From the previous parameters are calculated other relevant physical parameters such as the `Symbol Length` in seconds,

$$Symbol\ Length = \frac{1}{\frac{Data\ Rate \cdot 1000}{Bits\ Per\ Symbol}} \quad (4.1)$$

and the `Transmission time` of a MAC frame in seconds,

$$TXtime(x\ [bytes]) = (Phy\ Layer\ Overhead + x) \cdot \frac{1}{1000 \cdot \frac{Data\ Rate}{8}} \quad (4.2)$$

The `Data Rate` and `Bits per Symbol` parameters are crucial to determining correct timing in the MAC sublayer. All the parameters may be tuned in the simulation's configuration file changing their default values.

## 4.4 MAC sublayer parameters.

In this section are presented the most important variables which define, model and manage the behavior of the MAC sublayer. There are three main groups, namely the Finite State Machine (FSM) states, the MAC sublayer constants and the MAC sublayer attributes. Notice that this parameters do not correspond exactly to the ones defined in the IEEE 802.15.4 standard due this implementation is a variation of the previous standard.

### 4.4.1 FSM states.

The FSM states are used to structure the program flow. The main advantage is that the program, even being an asynchronous program, knows its actual state independly of the piece of code that it is executing at the moment. This allows to model better the behavior of the program, and specially in a complex program like the asynchronous one.

#### 4.4.1.1 MAC sublayer states.

The possible FSM states of the MAC sublayer are presented in Table 4.1. They indicate the state in which the MAC sublayer is at each moment.

State	Description
MAC_STATE_SETUP	MAC sublayer in setup or configuration mode. It is configuring the MAC sublayer parameters.
MAC_STATE_IDLE	MAC sublayer in idle state. It is not performing any task.
MAC_STATE_SLEEPING	MAC sublayer in sleep state. Set when the transceiver is in SLEEP state.
MAC_STATE_CSMA_CA	MAC sublayer in CSMA/CA state. MAC sublayer is performing the CSMA/CA access method.
MAC_STATE_CCA	MAC sublayer in CCA state. The MAC sublayer is performing the Clear Channel Assessment method.
MAC_STATE_IN_TX	MAC sublayer in Transmission state. The MAC sublayer is transmitting some frame.
MAC_STATE_PROCESSING	MAC sublayer is in Processing state. The MAC sublayer is performing some internal task.

TABLE 4.1: MAC sublayer states.

#### 4.4.1.2 Superframe states.

The possible FSM states of the TDMA superframe are presented in Table 4.2. They indicate the state in which the superframe is at each moment.

State	Description
TDMA_STATE_BEACON_SLOT	The superframe is in the Beacon period.
TDMA_STATE_DOWNLINK_CFP	The superframe is in the Downlink CFP.
TDMA_STATE_UPLINK_CFP	The superframe is in the Uplink CFP.
TDMA_STATE_UPLINK_CAP	The superframe is in the Uplink CAP.
TDMA_STATE_SCAN_PERIOD	The node is running the Scan period.
TDMA_STATE_INACTIVE	The node is in the Inactive period.

TABLE 4.2: Superframe states.

#### 4.4.2 MAC constants.

The MAC constants define the characteristics of the MAC sublayer. Only the ones used in the implementation are presented in the Table 4.3.

Constant Name	Description	Default Value
baseSlotDuration	The number of symbols forming the basic time period of a single slot.	100
baseSuperframeDuration	The number of symbols forming a superframe.	$baseSlotDuration \cdot numSuperframeSlots$

*Continued on next page*

Table 4.3 – Continued from previous page

Constant Name	Description	Default Value
<code>maxLostBeacons</code>	The number of consecutive lost beacons that will cause a MAC sublayer of a receiving device to declare a loss of synchronization.	4
<code>unitBackoffPeriod</code>	The number of symbols forming the basic time period used by the CSMA/CA algorithm.	20
<code>downlinkCFPSlots</code>	The number of slots forming the Downlink contention-free period.	20
<code>uplinkCAPSlots</code>	The number of slots forming the Uplink contention-access period.	30
<code>uplinkCFPSlots</code>	The number of slots forming the Uplink contention-free period.	20
<code>numSuperframeSlots</code>	Number of slots contained in any superframe.	<i>uplinkSlots</i> + <i>downlinkSlots</i>
<code>numCoord</code>	Number of coordinators nodes forming a PAN.	3

TABLE 4.3: MAC constants.

#### 4.4.3 MAC PIB attributes.

The MAC PIB comprises the attributes required to manage the MAC sublayer of a device. The ones used in the present implementation are presented in Table 4.4.

Attribute Name	Type	Description	Default Value
<code>associatedPAN</code>	Integer	ID of current PAN. (-1 if not associated).	-1
<code>nodeID</code>	Integer	Node short address.	0x00
<code>panID</code>	Integer	The identifier of the PAN on which the device is operating.	0xFF

*Continued on next page*



Table 4.4 – Continued from previous page

Attribute Name	Type	Description	Default Value
<code>isPANCoordinator</code>	Boolean	Indicate if the node is a coordinator node.	FALSE
<code>enableSlottedCSMA</code>	Boolean	Indicate if CSMA/CA algorithm is slotted.	TRUE
<code>macMinBE</code>	Integer	Minimum value of the backoff exponent in the CSMA/CA algorithm.	4
<code>macMaxBE</code>	Integer	Maximum value of the backoff exponent in the CSMA/CA algorithm.	6
<code>macMaxCSMABackoffs</code>	Integer	Maximum number of backoffs the CSMA/CA algorithm will attempt before declaring a channel access failure.	4
<code>macMaxFrameRetries</code>	Integer	The maximum number of retries allowed after a transmission failure.	3
<code>workingMode</code>	Integer	SWAN working mode running at the moment.	Maintenance Mode
<code>alreadyDisassociated</code>	Boolean	Indicate if the node has been already disassociated from the PAN.	FALSE
<code>macState</code>	Integer	Current MAC sublayer FSM state.	MAC_STATE_IDLE
<code>nextPacketRetries</code>	Integer	Number of retries left to be sent before discarding the <i>nextPacket</i> frame type.	1
<code>lostBeacons</code>	Integer	Number of consecutive lost beacon packets.	0

*Continued on next page*

Table 4.4 – *Continued from previous page*

Attribute Name	Type	Description	Default Value
NB	Integer	Number of Backoffs. Number of times the CSMA-CA algorithm was required to backoff while attempting the current transmission.	0
CW	Integer	Contention Window length. Number of backoff periods that need to be clear of channel activity before the transmission can commence.	2
BE	Integer	Backoff Exponent. Related to how many backoffs periods a device shall wait before attempting to assess a channel.	macMinBE
tdmaState	Integer	Current TDMA superframe FSM state.	TDMA_STATE_BEACON_SLOT
responseACKsent	Boolean	Control signal of the MAC layer that indicates if the Association Response ACK has been sent by the sensor/actuator node.	FALSE
stillMode	Boolean	Control signal of the MAC layer that indicates if the sensor/actuator node is in still mode.	FALSE
frameCounter	Integer	Control signal of MAC layer to complete properly the Association process. Inform the number of superframes runned after the last sending of the Association Request ACK.	0

*Continued on next page*

Table 4.4 – *Continued from previous page*

Attribute Name	Type	Description	Default Value
<b>frameTreshold</b>	Integer	Control signal of MAC layer to complete properly the Association process. Maximum number of superframes runned before putting the transceiver into SLEEP state.	5
<b>associatedNodeSlot</b>	Integer	Slot within the TDMA frame awarded to the sensor/actuator node after association process.	-1
<b>actualDownlinkSlot</b>	Integer	Control signal of TDMA that indicates the actual slot within the superframe at the moment.	0
<b>actualBP</b>	Integer	Control signal of the CSMA/CA algorithm that indicates the sum of the backoff periods done. It is used into the CSMA/CA control mechanism during Uplink CAP period.	0
<b>scanPeriod</b>	Boolean	Control signal that indicates if the MAC layer is running the Scan Period.	FALSE
<b>selectedCoord</b>	Integer	Coordinator selected by a sensor/actuator node to perform the Association process.	NULL

TABLE 4.4: MAC PIB attributes.

## 4.5 MAC sublayer logic.

This section explains the logic of the MAC sublayer. As the MAC sublayer is based on event-driven programming paradigm and as it is not feasible to represent the whole

program logic in one single flow-chart, its representation is divided into pieces of code which ones correspond to the execution after the occurrence of a specific event. The events are differentiated among four main groups: timer events, reception of frames from the PHY layer, transmission of frames to the PHY layer and reception of control commands from upper layers.

#### 4.5.1 Timer events.

The timers present within the MAC sublayer generate a determined event when they expire. These events are handled by the timer events handlers. The timer events are internal MAC sublayer events which are generated and processed by the same MAC sublayer. Usually they indicate the end of a time period or they force the MAC sublayer to take an action immediately.

##### 4.5.1.1 FRAME\_START

In Figure 4.2 is illustrated the behavior of the MAC sublayer when the FRAME\_START timer expires, that indicates that a new superframe is starting.

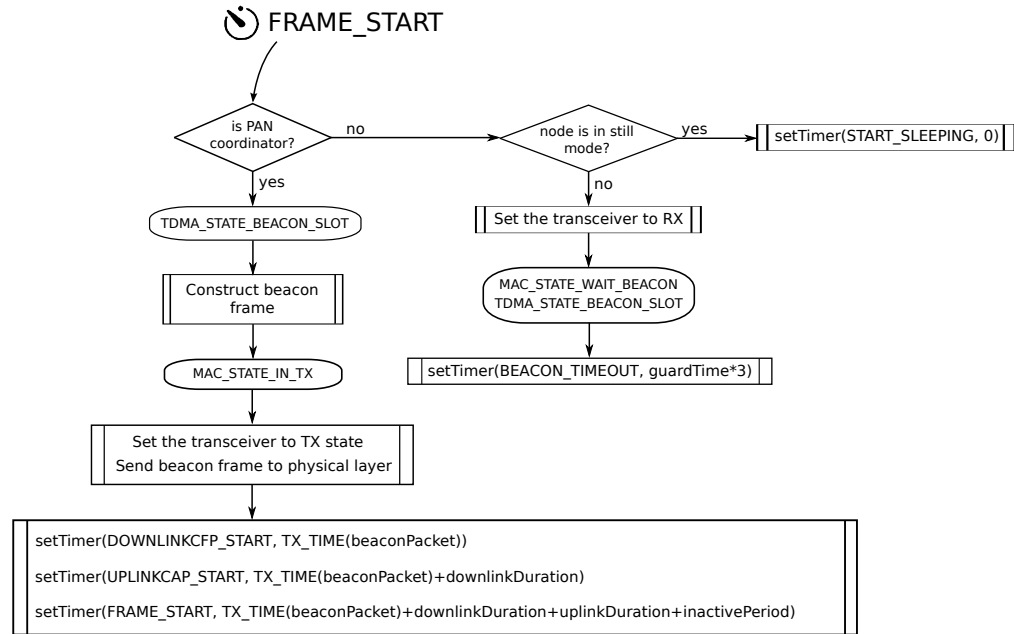


FIGURE 4.2: FRAME\_START event flow-chart.

If the node is not PAN coordinator, it sets its transceiver to SLEEP state if the node is in still mode, or to RX state to receive the beacon frame and setting the BEACON\_TIMEOUT timer.

If the node is PAN coordinator, it constructs the beacon frame and sends it directly to the physical layer to be transmitted. Also it sets the timers to determine the start of the Downlink and Uplink periods, and the start of the next superframe.

#### 4.5.1.2 BEACON\_TIMEOUT

In Figure 4.3 is illustrated the behavior of the MAC sublayer when the **BEACON\_TIMEOUT** timer expires, that indicates that the beacon frame sent by the coordinator node has not been received by the sensor/actuator node's MAC sublayer within the **guardTime** interval.

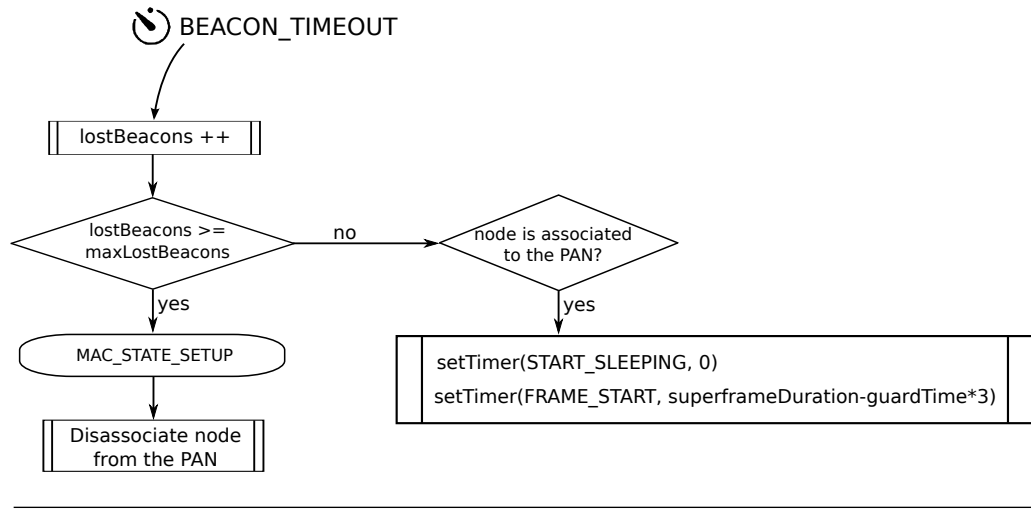


FIGURE 4.3: BEACON\_TIMEOUT event flow-chart.

If the number of lost beacon frames is higher than a predefined threshold, the sensor/actuator node MAC sublayer disassociates from the PAN. If this number is not higher than the threshold, it sleeps during the duration of the actual superframe and sets the **FRAME\_START** timer to be synchronized to receive the beacon frame of the next superframe.

#### 4.5.1.3 DOWNLINKCFP\_START

In Figure 4.4 is illustrated the behavior of the MAC sublayer when the **DOWNLINKCFP\_START** timer expires, that indicates that the Downlink CFP period is starting.

If the node is PAN coordinator, it sends the Dissasociation Request MLME command in the first Downlink slot, and sets the **DOWNLINK\_SLOT\_EXPIRATION** timer to indicate when the actual slot period ends.

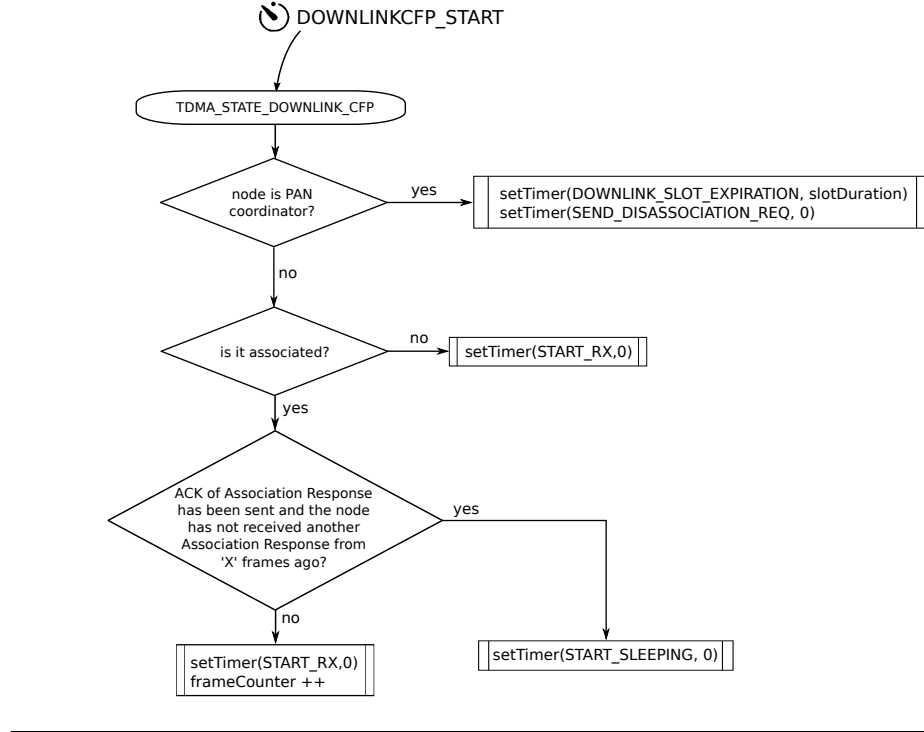


FIGURE 4.4: DOWNLINKCFP\_START event flow-chart.

If the node is not PAN coordinator and it is not associated to the PAN, it sets the transceiver to RX state; but if it is associated to the PAN (has received the Association Response), it performs a control mechanism to determine if the association procedure has been completed properly. If the sensor/actuator node does not receive an Association Response in  $x$  superframe periods means that the association procedure has been completed properly and sets the transceiver to SLEEP state; but if not, it sets the transceiver to RX state waiting to receive another Association Response from the coordinator indicating that the previous Association Response ACK has not been received by the coordinator and therefore the association procedure has not been completed properly.

#### 4.5.1.4 DOWNLINK\_SLOT\_EXPIRATION

In Figure 4.5 is illustrated the behavior of the MAC sublayer when the DOWNLINK\_SLOT\_EXPIRATION timer expires, that indicates that a specific time slot of the Downlink CFP period has finished.

This timer is only implemented in the PAN coordinator. First it checks if the actual slot is the last one of the downlink CFP period. If it is not, sets the DOWNLINK\_SLOT\_EXPIRATION timer again. Later it checks if there are frames to be transmitted in the TX buffer. If there are, it transmits the first frame of the buffer queue using the actual time slot. These

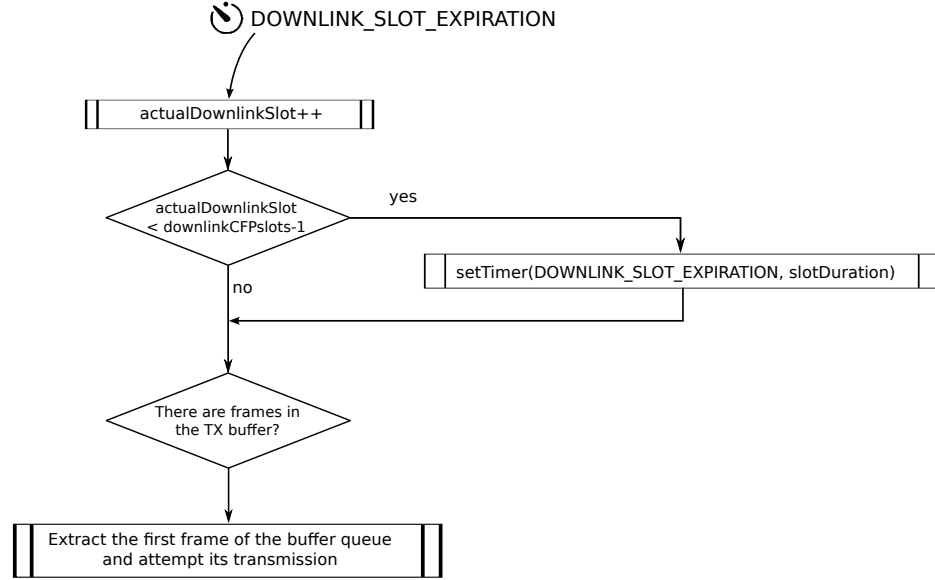


FIGURE 4.5: DOWNLINK\_SLOT\_EXPIRATION event flow-chart.

buffered frames are the Association Responses MLME commands. They are constructed in another part of the program once the coordinator node receives the association response payload from the upper layers.

#### 4.5.1.5 START\_SLEEPING

In Figure 4.6 is illustrated the behavior of the MAC sublayer when the START\_SLEEPING timer expires, that indicates that the node is setting its transceiver to SLEEP state to save energy.

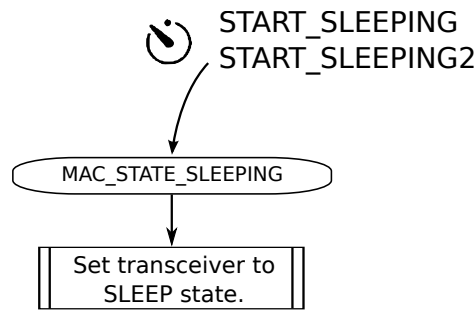


FIGURE 4.6: START\_SLEEPING event flow-chart.

#### 4.5.1.6 UPLINKCAP\_START

In Figure 4.7 is illustrated the behavior of the MAC sublayer when the UPLINKCAP\_START timer expires, that indicates that the Uplink CAP period is starting.

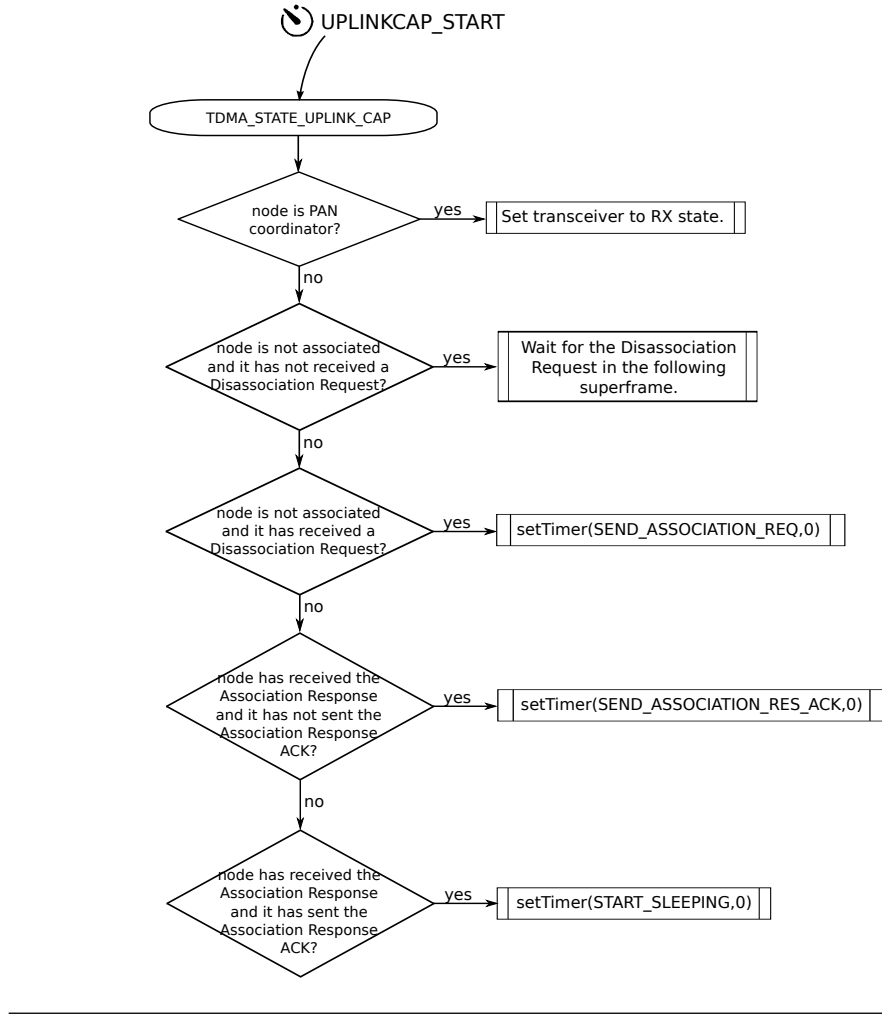


FIGURE 4.7: UPLINKCAP\_START event flow-chart.

If the node is PAN coordinator, during the Uplink CAP period it sets its transceiver to RX state to receive the frames sent by the different sensor/actuator nodes.

If the node is not PAN coordinator, it checks step by step the association procedure sending the corresponding association MLME commands.

#### 4.5.1.7 START\_SLEEPING\_INACTIVE

In Figure 4.8 is illustrated the behavior of the MAC sublayer when the START\_SLEEPING\_INACTIVE timer expires, that indicates that the node is putting its transceiver to SLEEP state before its own Inactive period.



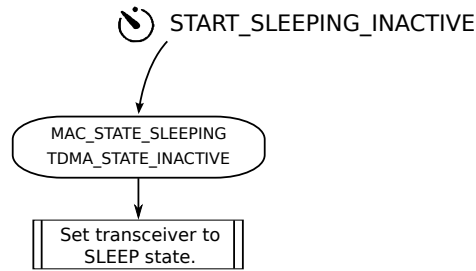


FIGURE 4.8: START\_SLEEPING\_INACTIVE event flow-chart.

#### 4.5.1.8 SEND\_ASSOCIATION\_REQ

In Figure 4.9 is illustrated the behavior of the MAC sublayer when the SEND\_ASSOCIATION\_REQ timer expires, that indicates that the MAC sublayer of a sensor/actuator node wants to transmit the Association Request MLME command.

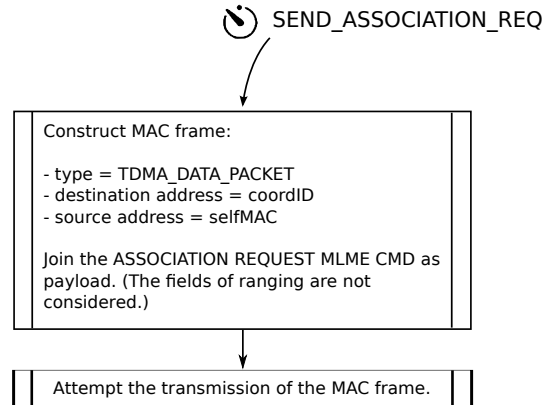


FIGURE 4.9: SEND\_ASSOCIATION\_REQ event flow-chart.

#### 4.5.1.9 SEND\_DISASSOCIATION\_REQ

In Figure 4.10 is illustrated the behavior of the MAC sublayer when the SEND\_DISASSOCIATION\_REQ timer expires, that indicates that the MAC sublayer of a coordinator node wants to transmit the Disassociation Request MLME command.

#### 4.5.1.10 SEND\_ASSOCIATION\_RES\_ACK

In Figure 4.11 is illustrated the behavior of the MAC sublayer when the SEND\_ASSOCIATION\_RES\_ACK timer expires, that indicates that the MAC sublayer of a sensor/actuator node wants to transmit the Association Response ACK.

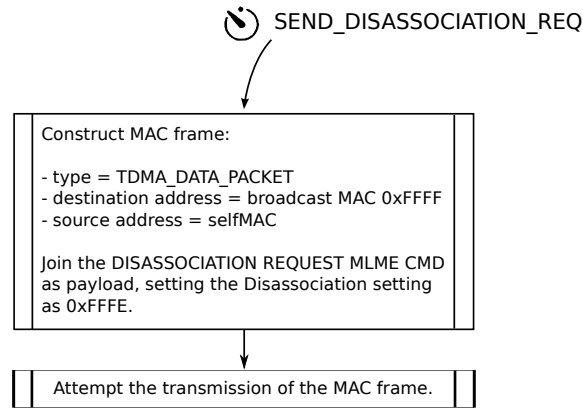


FIGURE 4.10: SEND\_DISASSOCIATION\_REQ event flow-chart.

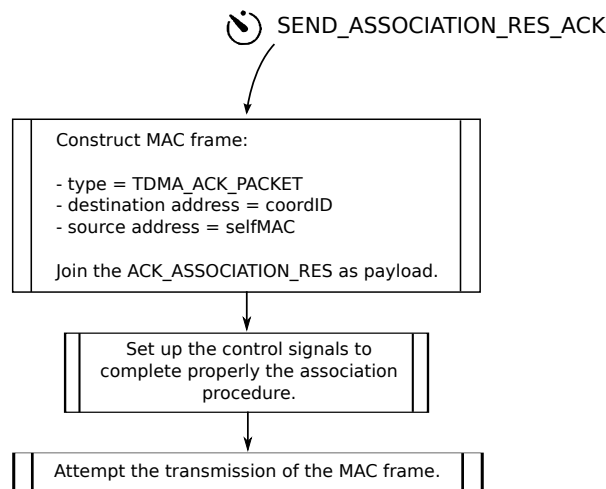


FIGURE 4.11: SEND\_ASSOCIATION\_RES\_ACK event flow-chart.

## 4.5.2 Reception of MAC frames from the physical layer.

In this subsection are explained the operations of the MAC sublayer when it receives the different MAC frames from the PHY layer used during the Maintenance Mode. There are three types of frames, i.e. beacon frames, MLME command frames and ACK frames. They are not usual data frames that the MAC sublayer only has to receive and send to the upper layers without the necessity to analyse the payload, otherwise these frames generate an action of the MAC sublayer in function of their field values.

### 4.5.2.1 TDMA\_BEACON\_PACKET

In Figure 4.12 is illustrated the behavior of the MAC sublayer when is receiving a Beacon frame from the physical layer.

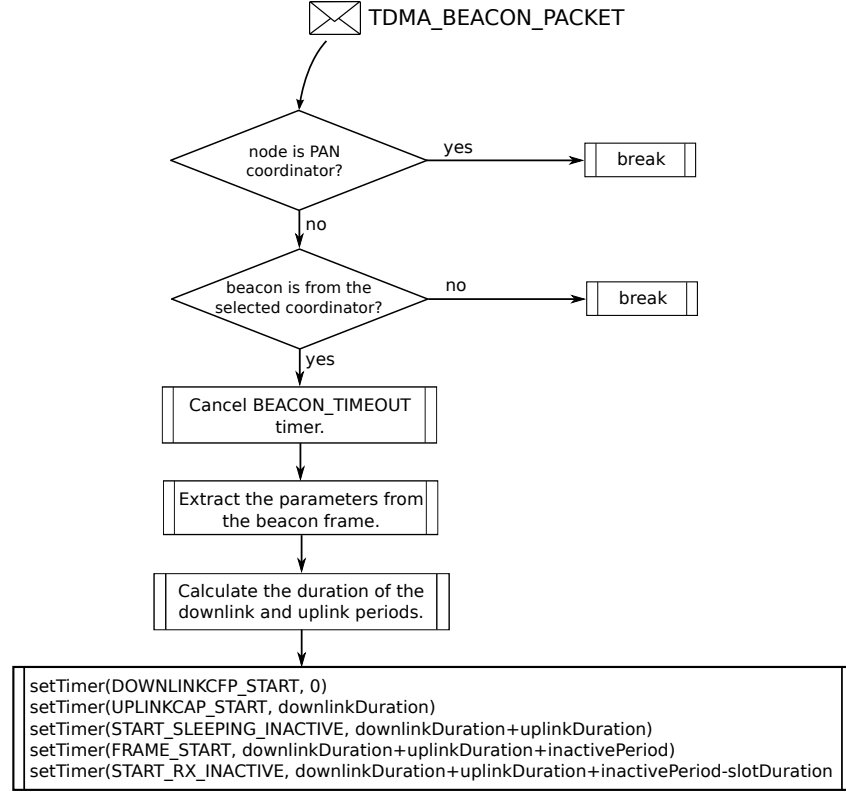


FIGURE 4.12: TDMA\_BEACON\_PACKET event flow-chart.

When the sensor/actuator node receives the beacon frame, first it cancels the `BEACON_TIMEOUT` timer. Then it extract the parameters from the beacon payload, calculate the duration of the Downlink and Uplink periods and set the corresponding timers to be synchronized with the PAN coordinator and the other devices of the wireless network.

Notice that the nodes set the transceiver to RX state one time slot before the end of the Inactive period, due the transceiver needs a period of time to do the transition from SLEEP state to RX state. If this transition is done late and the transceiver is not in RX state when the PAN coordinator sends the beacon frame, the node will not receive it.

#### 4.5.2.2 DISASSOCIATION\_REQUEST\_CMD

In Figure 4.13 is illustrated the behavior of the MAC sublayer when is receiving a Disassociation Request MLME command frame from the physical layer.

The objective of the Disassociation Request MLME command is to disassociate the sensor/actuator nodes from the PAN to perform a MAC sublayer reset. Depending of

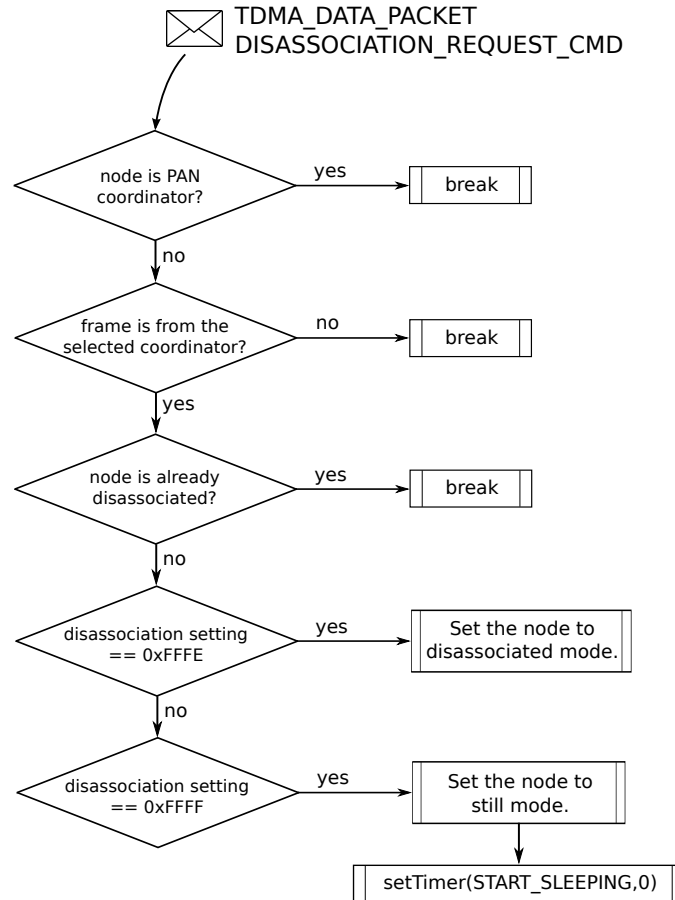


FIGURE 4.13: DISASSOCIATION\_REQUEST\_CMD event flow-chart.

its setting the nodes will put themselves to still mode `0xFFFF` or try to associate to the PAN in the following superframe `0xFFFE`.

In this type of frame there is no MAC address filtering because it is sent to the broadcast address by algorithm definition.

#### 4.5.2.3 ASSOCIATION\_REQUEST\_CMD

In Figure 4.14 is illustrated the behavior of the MAC sublayer when is receiving an Association Request MLME command frame from the physical layer.

The Association Request MLME commands are received by the PAN coordinators. When a coordinator receives it, the MAC sublayer sends the control message `ASSOCIATION_REQUEST_RECEIVED` to the upper layers to indicate which node is trying to associate to the PAN. The upper layers, that in the simulator implementation is the application layer, will process this control message consulting to a pseudo data base

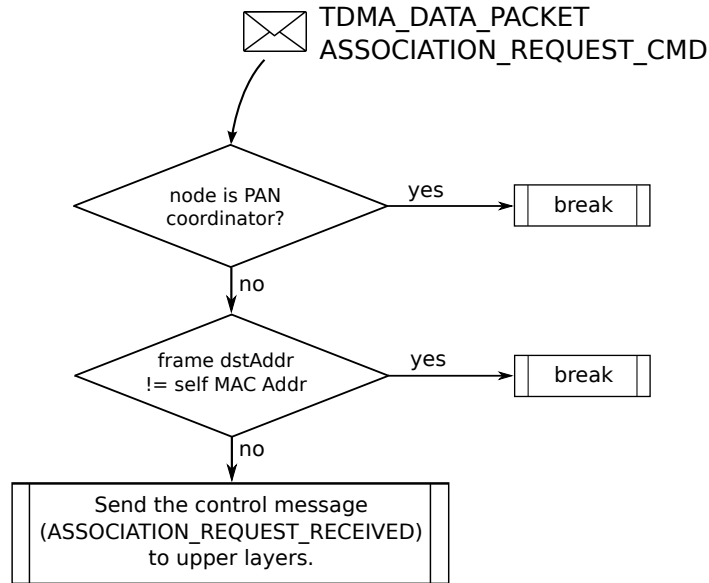


FIGURE 4.14: ASSOCIATION\_REQUEST\_CMD event flow-chart.

the response to this request. Notice that the MAC sublayer performs a MAC address filtering.

#### 4.5.2.4 ASSOCIATION\_RESPONSE\_CMD

In Figure 4.15 is illustrated the behavior of the MAC sublayer when is receiving an Association Response MLME command frame from the physical layer.

The Association Response MLME commands are received by the sensor/actuator nodes as response of an Association Request. The setting `0xFFFE` indicates that the association is temporary not allowed, but the sensor/actuator node will try to associate during the following superframe. The setting `0xFFFF` means that the sensor/actuator node cannot associate to the PAN and it will be no operative during the Operation Mode. Another value different from the previous means the TDMA time slot awarded to the sensor/actuator node during the Operation Mode.

#### 4.5.2.5 ACK\_ASSOCIATION\_RESPONSE

In Figure 4.16 is illustrated the behavior of the MAC sublayer when is receiving an Association Request ACK frame from the physical layer.

The Association Request ACKs are received by the PAN coordinators to indicate the succesful association to the PAN of a specific sensor/actuator node. When the coordinator receives this type of frame, the MAC sublayer sends the control message

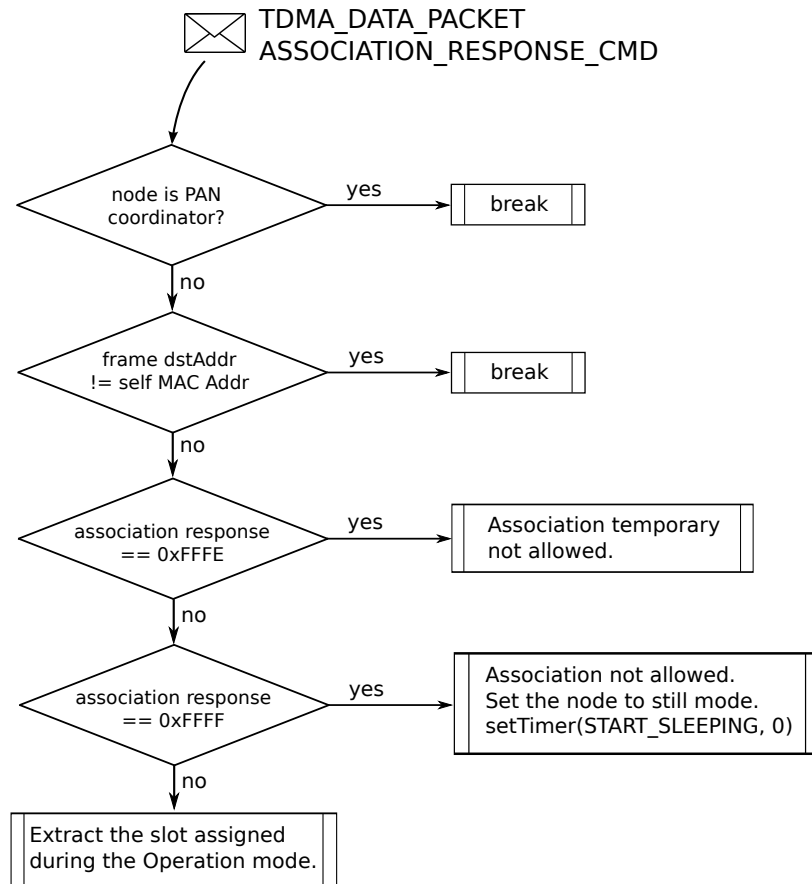


FIGURE 4.15: ASSOCIATION\_RESPONSE\_CMD event flow-chart.

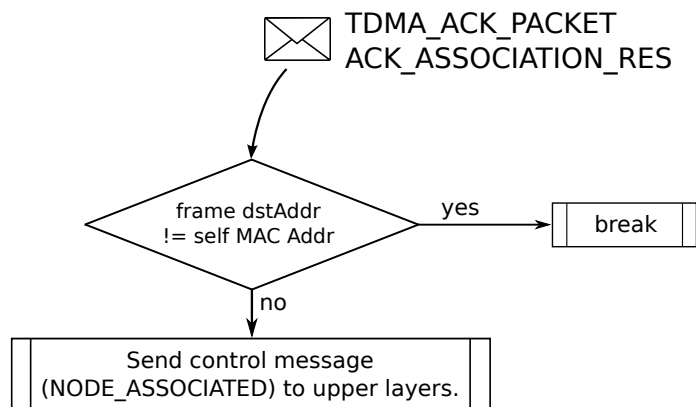


FIGURE 4.16: ACK\_ASSOCIATION\_RESPONSE event flow-chart.

NODE\_ASSOCIATED to the upper layers to indicate that the association procedure with this specific sensor/actuator node has been completed successfully.

### 4.5.3 Transmission of MAC frames to the physical layer.

In this subsection is explained how the MAC sublayer behaves when it wants to transmit a frame to the physical layer. As it will be shown, the transmission is not direct and it should pass a serie of controls to determine if the packet is allowed to be sent and how it will be sent. These determinations are done considering the type of frame to transmit, the type of node, the MAC sublayer state, the TDMA superframe state, and the number of remaining retries of the frame.

In Figure 4.17 and Figure 4.18 is illustrated the behavior of the MAC sublayer when is transmitting a frame to the physical layer.

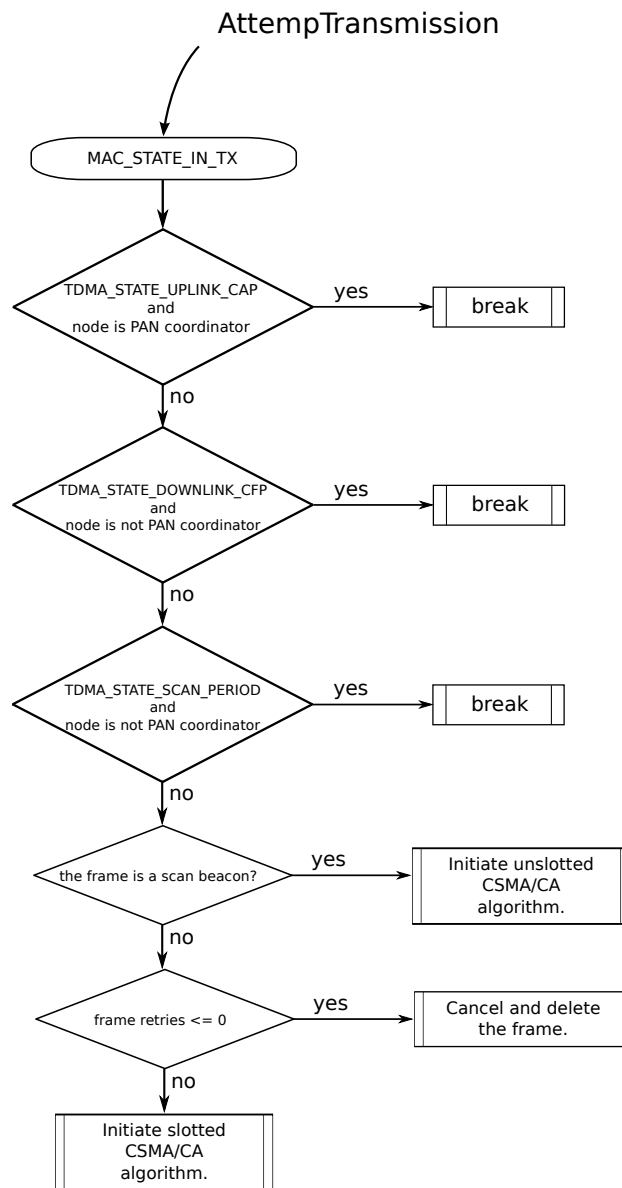


FIGURE 4.17: Attempt Transmission event flow-chart.

When the MAC sublayer finishes to determine if the frame can be send and how it will be sent, the CSMA/CA algorithm is executed. A description of how the CSMA/CA algorithm works can be found on the IEEE 802.15.4 standard document [1].

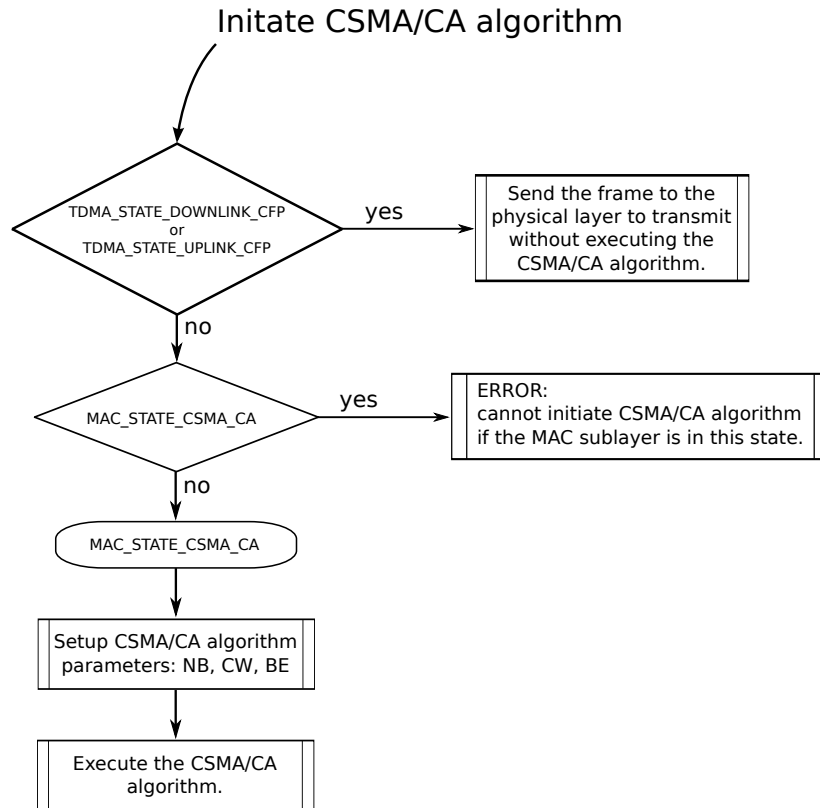


FIGURE 4.18: Initiate CSMA/CA algorithm event flow-chart.

#### 4.5.4 Reception of control commands from upper layers.

The control commands are data packets sent by the upper layer to the MAC sublayer. Its objective is to transport information from the upper layers to indicate the MAC sublayer to perform a specific action or as a response of a control command sent by the MAC sublayer to the upper layers. In the real implementation, i.e. over hardware platforms, these communications are not exactly like this and they depend strongly on the hardware platform, but for the purpose of this thesis it is enough to implement these communications in this way.

In the implementation of the Maintenance Mode, only is necessary a control command from the upper layers to the MAC sublayer and it is illustrated in the Figure 4.19. This command is the response of the control command `ASSOCIATION_REQUEST_RECEIVED` sent



by the MAC sublayer when it receives an Association Request from a specific sensor/actuator node. Within this control command is transported the association response field that will be sent within the Association Response MLME command frame.

Notice that after receiving this control command, the MAC layer automatically constructs the Association Response frame, and buffers it in the MAC sublayer transmission buffer to be transmitted during the Downlink CFP period.

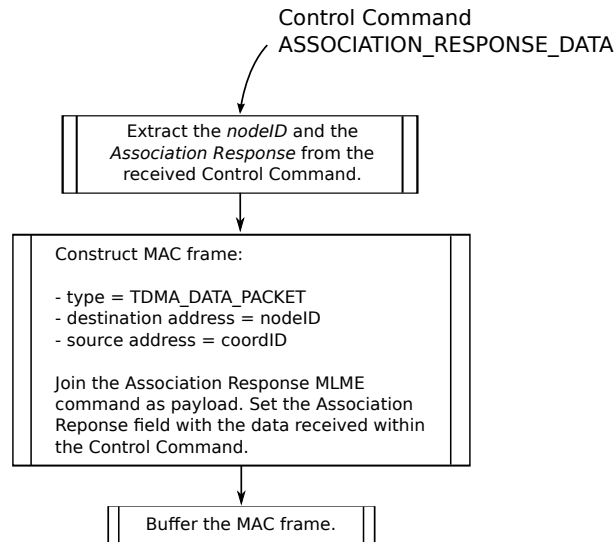


FIGURE 4.19: Control Command Association Response event flow-chart.

## 4.6 MAC implementation extensions.

In this section are explained three extensions of the MAC sublayer implementation. They are not directly related with the MAC sublayer itself or they are not considered in the actual SWAN's MAC sublayer specifications, but they affect and are somehow necessary to the well functioning of it, achieving a more realistic scenario and therefore a more realistic behavior of the MAC sublayer.

### 4.6.1 Communication with the upper layers. Data base consultation.

In the first implementation all the association process operations were carried out by the MAC sublayer. It means that the association answer to a specific node, transported by the Association Response, was not consulted to the SWAN server. On the contrary all the nodes willing to associate were able to do it without restrictions. This way was not the correct and the one defined by SWAN specifications. SWAN specifications determine that the association responses shall be saved into the SWAN server and these responses

will be in function of the application requirements, such as the throughput or if there are free slots in the TDMA frame.

This extension is implemented to emulate the communication between the coordinator node and the SWAN server. In this implementation the Application layer of the same coordinator node acts as the SWAN server. In the Castalia simulator, the communication between layers can be achieved by control messages. Control messages are messages that may be sent between different layers of the same emulated device, so they are not usual frames or packets.

From the practical point of view regarding the MAC sublayer and the wireless communication, it affects in a very similar way as a real scenario might do. It is illustrated in the Figure 4.20.

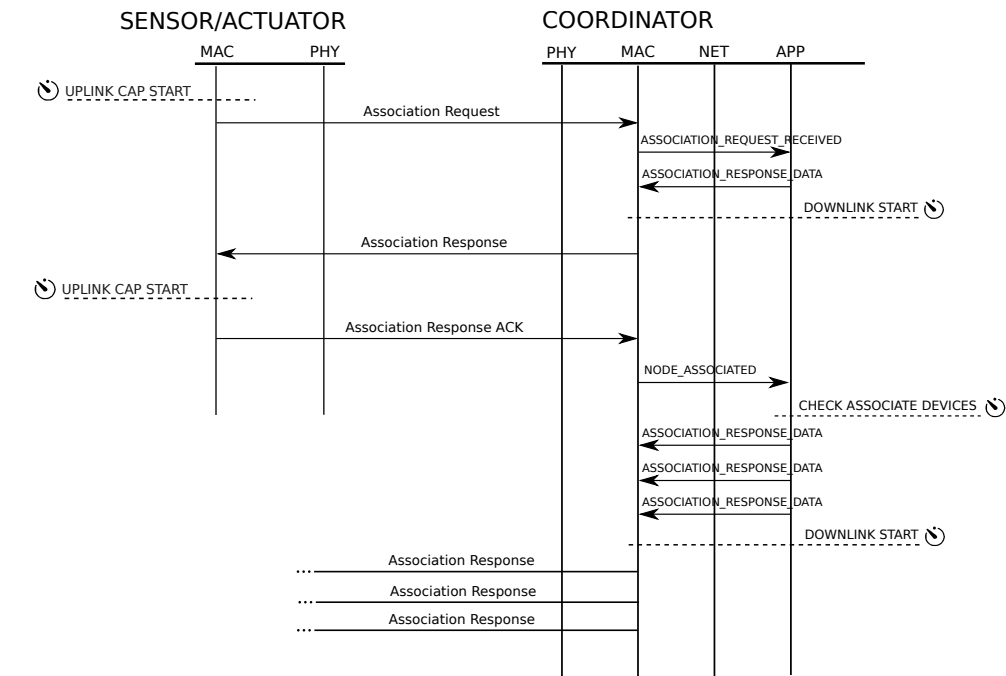


FIGURE 4.20: Control command messages exchange between MAC sublayer and upper layers.

The communication between the MAC sublayer and the Application layer, i.e. the communication between the coordinator node and the SWAN server, is necessary to determine the responses of the Association Requests sent by the sensor/actuator nodes and to inform the SWAN server that the association procedure with each sensor/actuator node willing to associate has completed properly.

The Application layer (SWAN server) once it receives an `ASSOCIATION_REQUEST_RECEIVED` control command from the MAC sublayer, adds the node ID to the `willingtoAssociate` vector and consults to its internal data base the association response for that specific

node. Later it constructs the `ASSOCIATION_RESPONSE_DATA` control command attaching the association response and sending it to the MAC sublayer.

Moreover, when the MAC sublayer receives an Association Response ACK indicating that the association procedure with a specific sensor/actuator node has completed properly, it sends to the Application layer a `NODE_ASSOCIATED` control command with the node ID inside informing about it. The Application layer deletes the node from the `willingtoAssociate` vector and adds the node to the `associatedNodes` vector.

In addition, the Application layer has an internal timer which at its expiration the timer handler checks the `willingtoAssociate` vector where are saved all the sensor/actuator nodes IDs that have performed an Association Request to the coordinator and they have not yet completed the association procedure properly sending to the coordinator node an Association Response ACK, maybe because the Association Response ACK has not been received by the coordinator due packet collisions in the wireless channel or that the sensor/actuator nodes had no possibility to access to the wireless medium when they were executing the CSMA/CA algorithm. For these sensor/actuator nodes the Application layer constructs `ASSOCIATION_RESPONSE_DATA` control commands and sends them to the MAC sublayer, where the new ones Association Responses frames will be constructed and sent again to the sensor/actuator nodes requesting from them the Association Response ACKs.

#### 4.6.2 Several clusters within the same wireless sensor network.

The implementation of only one coordinator, consequently only one cluster, is very limited. Considering the case that the wireless sensor network only has one coordinator, all the nodes shall associate to it, with intrinsic problems such as low coverage or high computation effort.

A solution to solve that is to implement a wireless sensor network managed by several coordinators, each of them creating its own cluster. This implementation will improve the coverage, the computation effort in each coordinator device will decrease exponentially and may increase the channel capacity.

By specification definition, all the coordinators have to run its own clusters using the same frequency band, due the localization service requirements. Therefore, the communications within the overlapping clusters can not run at the same time and as is illustrated in Figure 4.21, they run them by time multiplexing. Consequently, a synchronization mechanism should be carried out among the several clusters to know when each

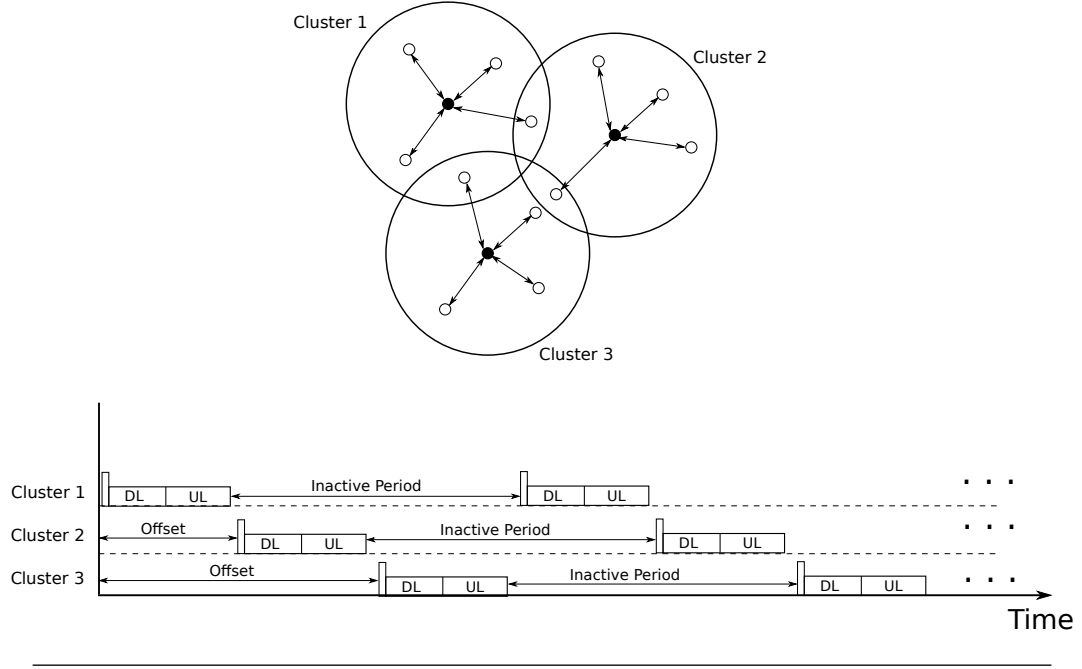


FIGURE 4.21: Several clusters within the same WSN.

cluster can start its internal communications without interfering with the communications of the other clusters.

To do the synchronization among the different clusters three parameters should be calculated in the SWAN server and send to each coordinator. These parameters are:

- *Offset*: displacement of the superframe respect to the wireless network start instant.
- *Active period*: duration of the superframe.
- *Inactive period*: time within which the cluster is not operative.

As can be seen in the Figure 4.21, the Offset values are equal to the sum of the superframe durations of the clusters that have started before than the current cluster. The Inactive periods values are equal to the sum of the superframe durations of the rest of clusters not considering the current and in addition, there are summed to the Inactive period some intersuperframe spacings, i.e. a time interval between two consecutive superframes from different clusters.

#### 4.6.3 Scan period.

As is explained in the previous subsection, a WSN can be formed by several clusters each of them managed by a coordinator. The sensor/actuator nodes once the WSN is

started and the Maintenance Mode is running they will try to associate to it. If there are several clusters, the sensor/actuator nodes have to decide to which of them they will try to do the association.

To do this decision a *Scan period* has been implemented based on a passive scanning method which aim is to provide to the sensor/actuator nodes the parameters for synchronizing the communications among the different clusters of the WSN, i.e. the offset and inactive period, and to provide a signal to perform RSSI measurements to decide to which coordinator node do the association in function of the average RSSI received from each one of them.

During the Scan period, illustrated in Figure 4.22, the coordinator nodes send scan beacons to the wireless medium accessing to the channel using unslotted CSMA/CA access method, however the coordinators must be configured by the SWAN server before starting the Scan period to provide to them some parameters to transmit their scan beacons in an ordered way.

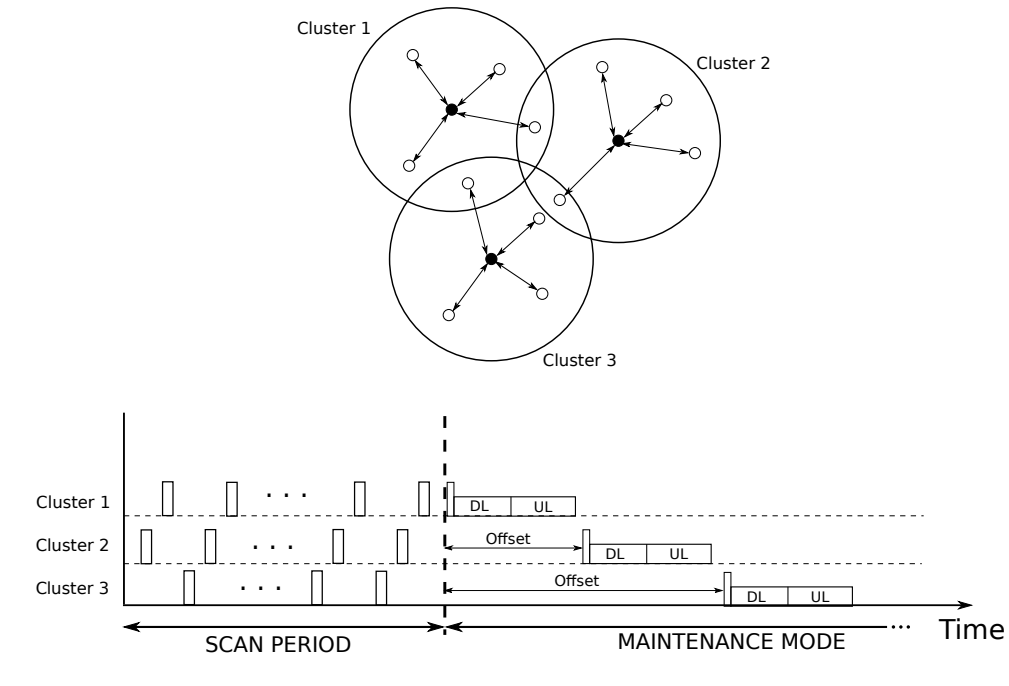


FIGURE 4.22: Scan period.

The scan beacon payload is illustrated in Figure 4.23. The fields within the scan beacon payload are:

- **Beacon ID:** may be indicated that the wireless network is running the scan period.
- **Inter Scan Beacon Spacing:** it is a time value that indicates the time period between two consecutive scan beacon frames.

Field	Beacon ID	Inter Scan Beacon Spacing	Offset	Inactive Period	Number of coordinators	Start association
Octets	1	8	8	8	1	1
<b>Scan Beacon Payload</b>						

FIGURE 4.23: Scan Beacon payload.

- **Offset:** it is a time value that indicates the offset of the superframe of the current cluster once the Maintenance Mode starts.
- **Inactive Period:** it is a time value that indicates the inactive period of the current cluster during the Maintenance Mode.
- **Number of coordinators:** is the total number of coordinator nodes or clusters that build up the WSN.
- **Start association:** is a flag that indicate the sensor/actuator nodes if the Scan period has finished and they have to prepare its own MAC sublayers to start the Maintenance Mode and therefore start the association procedure.

On the other hand, during the Scan period the sensor/actuator nodes only listen the wireless medium receiving the scan beacon frames from the coordinators that are in range. They process the information within the scan beacon payload and also construct a table with the average RSSI received from each coordinator. When the scan period ends, which is indicated setting the **Start association** field equal to **true**, the nodes decide to which coordinator/cluster perform the association in function of the high RSSI average. Also they configure its own MAC sublayers to prepare them to only be operative during the communications of the coordinator/cluster selected, setting the **Offset** and **Inactive period** parameters.

In Figure 4.24 is illustrated the behavior of the coordinator's MAC sublayer when the **SCAN\_START** timer expires, that indicates that a subperiod of the Scan period is starting.

If the node is not PAN coordinator, it only listens the channel to receive the scan beacon frames sent by the coordinator nodes. If the node is a PAN coordinator node, it constructs the scan beacon frames. If the scan period is finished, which is determined by an external time variable defined in the configuration file, the coordinator sets the field **startAssociation** equal to 1 which indicates that the nodes that receive this flag should change the state of his MAC sublayer going from Scan Period state to Maintenance Mode state, in which they can start the association procedure. Also, if the scan period is finished the coordinator will change its MAC sublayer state indicated with the flag **scanPeriod**. Notice that the beacon frames are sent using unslotted CSMA/CA.

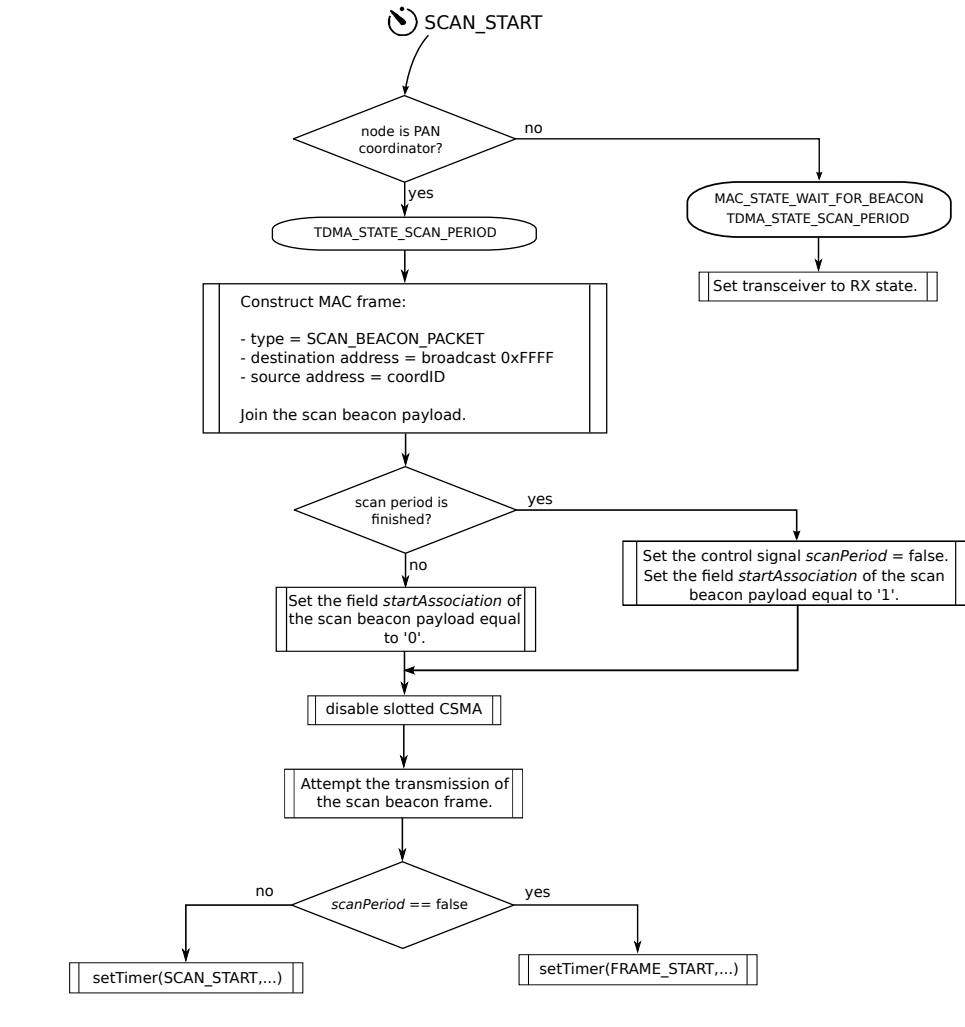
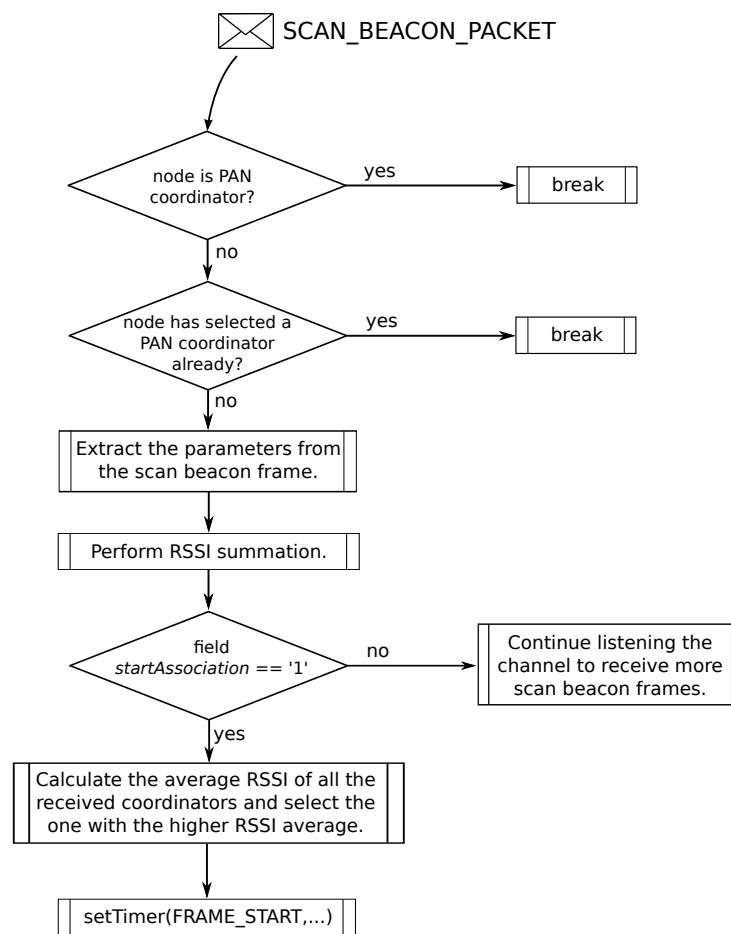


FIGURE 4.24: SCAN\_START event flow-chart.

In Figure 4.25 is illustrated the behavior of the sensor/actuator node's MAC sublayer when is receiving a Scan Beacon frame from the physical layer.

FIGURE 4.25: `SCAN_BEACON_PACKET` event flow-chart.



## 4.7 Analysis, experiments and results.

This section consists of the deployment and the simulation of a WSN within a well defined space and the measurement of some previously determined parameters while modifying the superframe structure and the position and the number of coordinators, to test and analyze the performance of the implemented MAC layer's Operation Mode.

The space within which the deployment is simulated has the dimensions of a Boeing A321 passenger's cabin as is illustrated in Figure 4.26.

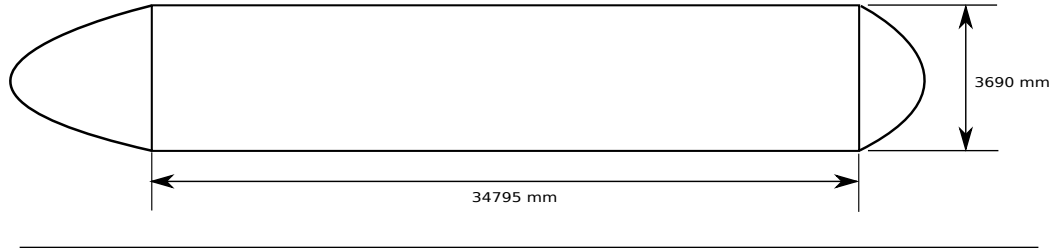


FIGURE 4.26: Boeing A321 passenger's cabin dimensions.

The position of the nodes is permanent but the number and the position of the coordinators vary between scenarios, as is shown in the Figure 4.27.

The Operation Mode superframe structure takes different values in all the scenarios considering the number of downlink slots and uplink slots as is presented in Table 4.5.

Structure	Downlink slots	Uplink slots
A	25	35
B	25	50
C	30	65

TABLE 4.5: Superframe structure.

To test and analyze the performance of the Operation Mode the following parameters are measured in each scenario:

- The number of nodes associated to each coordinator.
- The time required to associate all the nodes to the coordinators.
- The quotient between the Association Response ACKs received by coordinator and the Association Responses transmitted by the same coordinator, as is defined in the following formula.

$$Q = \frac{\text{Association Responses ACKs RX}}{\text{Association Responses TX}}, \quad Q \in (0, 1] \quad (4.3)$$

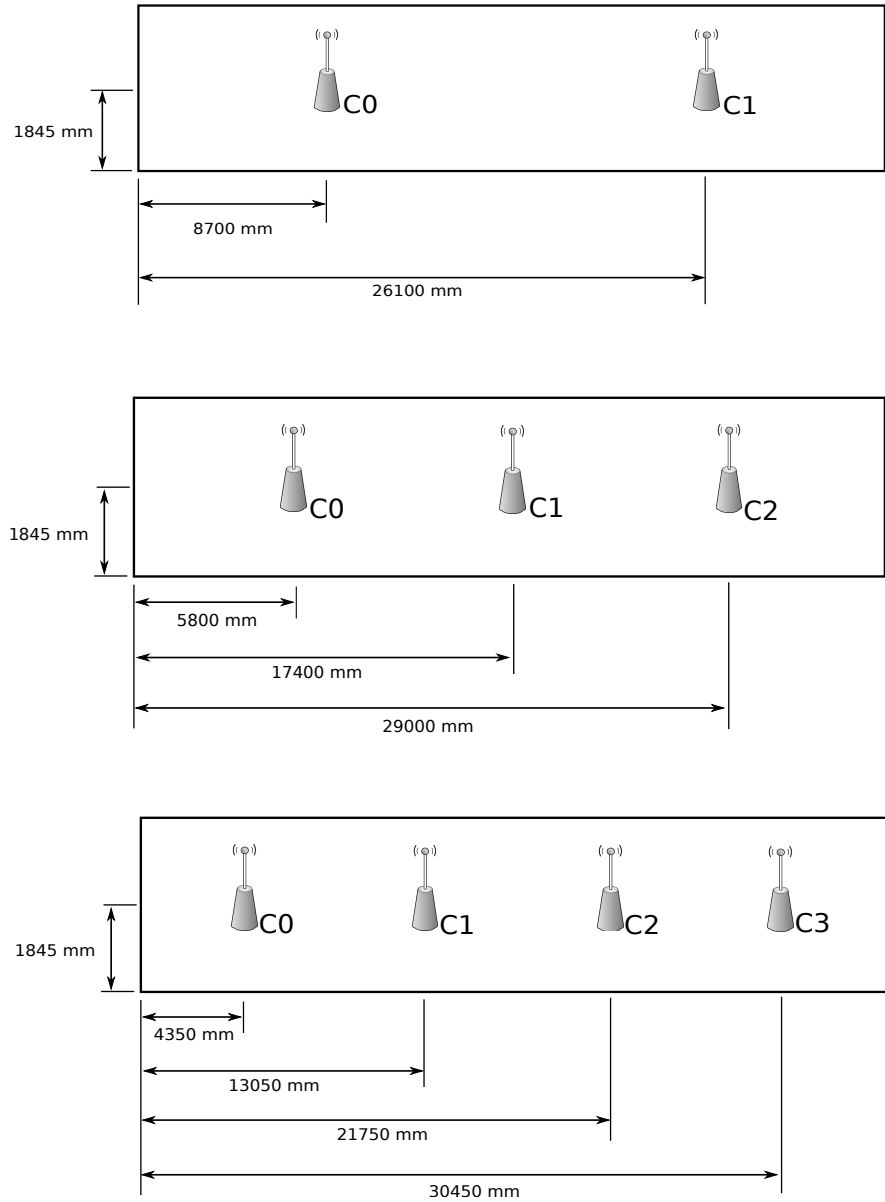


FIGURE 4.27: Position of the coordinators within the cabin.

If the quotient is close to 1, means that there are few packet retransmissions. On the contrary, if the quotient is close to 0 means that there are too much retransmissions, and therefore the MAC layer's operation efficiency is lower.

In all the deployments is done a simulation of 10 seconds, where the first 300ms are equal to the scan period, i.e. time to allow the nodes to select a coordinator to whom associate.

### 4.7.1 Deployment 1.

This deployment is done in an uniform distributed 6x30 mesh, which consists of 6 rows and 30 columns of nodes, summing a total of 180 nodes, but due simulator bugs in each scenario this number vary. The results are presented in the following Tables 4.6, 4.7 and 4.8.

Scenario	Structure	Total	C0	C1	C2	C3
2 coordinators	ABC	179	108	71	-	-
3 coordinators	ABC	179	72	72	35	-
4 coordinators	ABC	178	54	54	54	16

TABLE 4.6: Number of nodes associated to each coordinator.

Scenario	Structure	C0	C1	C2	C3
2 coordinators	A	5870	4780	-	-
	B	6720	5850	-	-
	C	6850	6330	-	-
3 coordinators	A	5920	5770	3240	-
	B	5500	7090	4280	-
	C	6970	7100	6360	-
4 coordinators	A	6660	5970	6460	3020
	B	8200	8320	7950	4150
	C	4070	5470	5620	2090

TABLE 4.7: Time (ms) required to associate all the nodes to the coordinators.

Scenario	Structure	C0	C1	C2	C3	Average
2 coordinators	A	0.4865	0.5420	-	-	0.5142
	B	0.5294	0.4797	-	-	0.5046
	C	0.4909	0.4539	-	-	0.4724
3 coordinators	A	0.4730	0.4500	0.6140	-	0.5123
	B	0.5669	0.4675	0.6250	-	0.5532
	C	0.4675	0.5255	0.6481	-	0.5471
4 coordinators	A	0.5192	0.5400	0.4779	0.8000	0.5843
	B	0.5094	0.5243	0.4749	0.6400	0.5379
	C	0.4074	0.3519	0.3293	0.3333	0.3555

TABLE 4.8: Q factor.

### 4.7.2 Deployment 2.

This deployment is done in an uniform distributed 5x18 mesh, which consists of 5 rows and 18 columns of nodes, summing a total of 90 nodes, being less dense than Deployment 1. The results are presented in the following Tables 4.9, 4.10 and 4.11.

Scenario	Structure	Total	C0	C1	C2	C3
2 coordinators	ABC	90	45	45	-	-
3 coordinators	ABC	90	30	30	30	-
4 coordinators	ABC	90	15	20	45	10

TABLE 4.9: Number of nodes associated to each coordinator.

Scenario	Structure	C0	C1	C2	C3
2 coordinators	A	3300	3400	-	-
	B	4510	3900	-	-
	C	2830	3600	-	-
3 coordinators	A	3300	3110	3340	-
	B	3310	4140	4600	-
	C	4990	4700	4140	-
4 coordinators	A	3120	4390	5980	2930
	B	2810	3430	6200	2890
	C	3450	3430	5070	3380

TABLE 4.10: Time (ms) required to associate all the nodes to the coordinators.

Scenario	Structure	C0	C1	C2	C3	Average
2 coordinators	A	0.6000	0.6490	-	-	0.6215
	B	0.6338	0.6338	-	-	0.6338
	C	0.6818	0.7031	-	-	0.6925
3 coordinators	A	0.7317	0.6250	0.7317	-	0.6961
	B	0.6250	0.7143	0.5882	-	0.6425
	C	0.5769	0.7826	0.6977	-	0.6857
4 coordinators	A	0.6818	0.6415	0.7419	0.7692	0.7086
	B	0.7892	0.7407	0.6716	1	0.8004
	C	0.3571	0.4091	0.4340	0.5	0.4250

TABLE 4.11: Q factor.

### 4.7.3 Analysis and conclusions.

The analysis and the conclusions are done globally with the data and the results gathered in both deployments.

Regarding the association time the analysis is:

- As less nodes trying to associate, less time required to these nodes to associate to the network.
- As more number of coordinators present in the scenario, the time between two consecutive frames of the same cell is higher and therefore it increments the association time.
- There is not a clear relation between the superframe structure and the association time.

Regarding the performance of the MAC layer (Q factor) the analysis is:

- As less nodes higher Q factor, therefore better MAC layer performance.
- The number of coordinators do not affect the Q factor.
- The superframe structure seems to do not affect the Q factor, but the scenario 4coordinators-C is the worst case in both deployments.

The general conclusions are:

- We can assume that at lower Q factor, i.e. lowest MAC layer performance, high energy consumption by the nodes of the cluster due higher packets retransmissions.
- To improve the efficiency of the MAC layer, the estimated number of nodes that will associate to each cluster must be calculated previously, and hence calculate the number of coordinators that will deal with the total number of nodes.
- It may be implemented a frequency reuse between two clusters enough separated, reducing the the time between two consecutive superframes of the same cluster and therefore decreasing the association time.
- It must be investigated further if the superframe structure affects the performance of the MAC layer.
- The position of the coordinators does not affect directly to the performance of the MAC layer, only affect the number of sensor nodes that these coordinators manage.

## 4.8 Conclusions and open problems.

In this chapter it has been explained the implementation of the MAC sublayer's Maintenance Mode adapted to the requirements stated by the MAC sublayer's design papers of the SWAN project. This implementation has been done using the Castalia simulator based on the OMNeT++ platform, under the event-driven programming paradigm. In addition, there have been presented the parameters of the MAC sublayer, which define, model and manage the behavior of the MAC sublayer. Also it has been presented the logic of the MAC sublayer using flow-charts, where each flow-chart correspond to the code execution after the occurrence of a specific event. Moreover, it has been explained the implementation of the MAC sublayer's extensions, which are necessary to the well functioning and the ability to simulate the current SWAN MAC sublayer implementation. Finally, there have been conducted some experiments deploying a simulated WSN to test and analyze the performance of the Maintenance Mode implementation.

Some questions and problems to solve regarding the implementation of this specific MAC sublayer running mode are:

- May it be possible to modify the MAC sublayer's logic aiming to improve its performance, and therefore reducing the energy consumption, maintaining the design and the general schema?
- As it is well known the CSMA/CA access mode is not efficient, but how the CSMA/CA algorithm influence on the overall MAC sublayer's performance and how it may be tuned to improve it?
- Which are the limitations of the current design of the Maintenance Mode and how them may be improved?
- How the implementation may change once it is transferred to a hardware platform?
- How the system and the MAC sublayer will handle the orphaned devices?
- How the MAC sublayer will recover from a failure?
- How the MAC sublayer implementation may change if we implement redundancy in the wireless part to improve its reliability?
- As the MAC sublayer is the first logic gate to the WSN system, which security solution may be implemented on it?

## Chapter 5

# TDMA Scheduling algorithm.

### 5.1 Introduction.

In the last years the wireless communications, and specifically the ones of small and/or mobile devices, have become a very important field of investigation and study. In these systems the wireless medium is shared among several nodes sometimes by means of time multiplexing, that is to divide the time axis in slots within which the nodes are allocated to perform the wireless communication. This kind of medium access method is named Time Division Multiple Access (TDMA).

In contrast, there are other medium access methods for example Carrier Sense Multiple Access (CSMA), Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA), Spatial Division Multiple Access (SDMA), etc. Regarding small devices like the wireless sensor/actuator network nodes, CSMA has the drawback that is non-deterministic and conflict with timeliness and power constraints, FDMA may be inefficient for periodic messages in real-time systems and in our case do not accomplish the system requirements due the localization service, and CDMA or SDMA are complex and cannot be implemented with the limited resources of wireless sensor/actuator networks nodes. Consequently the implementation of these other medium access methods may be harder increasing the network interface complexity and sometimes not feasible.

Another important characteristic in WSNs is that the main limitation of the node devices is the power supply and considering that the main source of power drainage is the radio communication transceiver, the study of scheduling algorithms that allow the transceiver to be turned off for most of the time is a main goal. In addition, in many applications such as monitoring apps, real-time tasks, multimedia apps, sensing apps, tasks that require QoS, etc., the messages need to be scheduled at some predefined rate determining which

node has access to the medium in real-time. In these cases, the scheduling algorithms carry out the allocation of the different messages into the slots trying to accomplish the requested application periods.

In wireless sensor/actuator networks, the TDMA communications are quite dependent from the cluster coordinator specially in centralized networks as the SWAN WSN, and the nodes require a perfect synchronization in time normally performed by the use of beacon frames. Traditionally TDMA is implemented using a lookup table who determines which message has access to the medium for each unit of time. The maximum size of this table is the time length that the TDMA schedule defines, often the number of time units after which the schedule repeats or is updated. The minimum size of this table is the total number of messages in the schedule, which is achieved by removing from the table time units in which no messages are scheduled. If each node of the network does not need to know the entire schedule but only the subset of the schedule that involves the messages that it either sends or receives, then the table may be optimized for each node.

When periodic allocation must be performed the scheduling algorithm, given a set of messages each one with its own length and requested period, produces an assignment of time to nodes while trying to optimize two different measures, i.e. approximation and smoothness. To have a good approximation is when the assigned period of the job, that is the occurrence between two consecutive events of the same job, is close to the requested period. Smoothness may be defined as the small variation of the latency; a schedule is said to have a good smoothness if the occurrences of each job are as evenly spaced as possible. The best approximation is when the granted periods are exactly the requested periods, whereas the best possible smoothness is achieved by perfectly periodic schedules, where each job is scheduled exactly every  $p$  time units, for some  $p$  called the period of that job. Notice that is easiest to optimize one measure neglecting the other. On the other hand, there are some cases in which perfect schedules are not possible. Consider a slot model with two messages, the former with period 2 and the latter with period 3. The message with period 2 will occupy all the even-numbered slots or all the odd-numbered slots, while the message with period 3 will occupy sometimes even-numbered slots and sometimes odd-numbered slots; this leads to the possibility of slot allocation collisions and non-cyclic schedules. Therefore there are cases in which the periods must be adapted and consequently the granted periods will not match the requested ones.

Finally, the quality of the schedule is measured by the ratios between the requested period and the allocated period for each message, either by the weighted average or by the maximum of these ratios over all messages [4].



## 5.2 Related work.

Some general and fundamental aspects of real-time scheduling as well as some scheduling algorithms are demonstrated in the pinwheel schedule [5]. Even it is using the satellite communications context, it may be applied to any TDMA communication.

In [4, 6] are explained the properties of perfectly periodic schedules, in which each client gets a slot precisely every predefined number of time slots. A mathematical basis, models to analyze the quality of the algorithms, a method to develop these kind of schedules, i.e. tree scheduling, and some perfect periodic scheduling algorithm examples are presented.

A specific contention-free periodic message scheduler focused on wireless sensor/actuator networks is studied in [7]. It explains some basic but essential mathematic definitions and rules in how define a periodic TDMA communication. Also it shows the rules of the synchronization protocol in function of the synchronization protocol overhead and the clock error. In addition it presents three methods for finding a contention-free message set using period and phase assignment based on perfect periodic scheduling.

In [8, 9] are described a slot pre-allocation protocol (SPAP) and a self-organizing slot allocation protocol (SRSA) respectively, which aim is to reduce the interferences in multi-cluster sensor networks, improving the efficiency as well as reducing the packet losses and therefore the energy consumption in such networks. They are totally based on TDMA, do not using some other medium access mechanisms such as CDMA or FDMA.

The Guaranteed Time Slot (GTS) mechanism used in the IEEE 802.15.4 MAC standard protocol is analyzed in [10]. The paper analyzes the performance of GTS mechanism giving a full understand of its behavior with regards to delay and throughput metrics. In addition, [11] analyzes the application of decay rate for WSN buffer dimensioning in GTSs based networks, implementing a differentiation of service mechanism to provisioning a QoS.

In [12] is presented a slot allocation approach which can be used in any TDMA based MAC design to improve the latency of the packets. It is based on event-driven applications and the main peculiarity is that the nodes only transmit the difference between the threshold and the measurement values, and the nodes with the same data are grouped transmitting the data only once. While in [13] is exposed a Low-energy Adaptive Slot Allocation (LASA) which replaces the fix slot size in classical TDMA schemes by a variable slot size that dynamically adapts to the data size generated at the sensor nodes. In addition, in [14] is showed a proportionally fair rate allocation in regular WSN in which the main goal is to maximize the network throughput by finding the optimal rate allocation for individual/competing end-to-end sessions.

### 5.3 Basics of the implemented scheduling algorithm.

The scheduling algorithm implemented has two main characteristics: it is perfect periodic and contention-free. Contention-free means that no two messages may be scheduled in the same time unit, so there is no slot allocation collision and therefore the schedule is free of contention. Perfect periodic means that each message is scheduled exactly every  $p$  time units, for some  $p$  called the period of that message.

Periodic message model with a contention-free message set has the property that messages are sent on release and packets are transmitted to completion within one time unit. Besides, contention-free enable each node to only schedule the messages that concerns it. To achieve the perfect periodic scheduling the periods are converted to power of two harmonic periods, being the granted periods less or equal than the requested periods. In addition, perfect periodic schedules have the best fairness among all schedules.

The nodes only transmit one packet per message, that means that the messages are composed only by one packet, consequently the length of the time slot should be adapted to the larger packet length. The algorithm is executed off-network, that is running in a computer not within the sensor/actuator wireless network but connected to it through a gateway which in the SWAN system is the SWAN Server, being a centralized scheduling algorithm.

#### 5.3.1 Scheduling definitions and terminology.

Here are defined the main concepts concerning periodic scheduling:

- *The maximum requested period  $T_{max}$* : the maximum period is associated with the maximum period of a message that must be scheduled. This measure is important because the *maximum harmonic period  $T_{max}^h$*  defines the length of the *hyperperiod*.
- *The minimum requested period  $T_{min}$* : the minimum period is associated with the minimum period of a message that must be scheduled. This measure is important because the *minimum harmonic period  $T_{min}^h$*  defines the length of a single *subperiod*.
- *The subperiod*: it is determined by the *minimum harmonic period  $T_{min}^h$*  and defines the maximum length of the subscheduling. It is illustrated in Figure 5.1.
- *The hyperperiod*: it is determined by the *maximum harmonic period  $T_{max}^h$*  and is the period in which the whole schedule repeats, being the scheduling a cyclic

scheduling. The hyperperiod of a set  $M$  is defined as:

$$\hat{T}_M = \text{lcm}_{i=1 \dots |M|}(T_i) \quad (5.1)$$

It is illustrated in Figure 5.1.

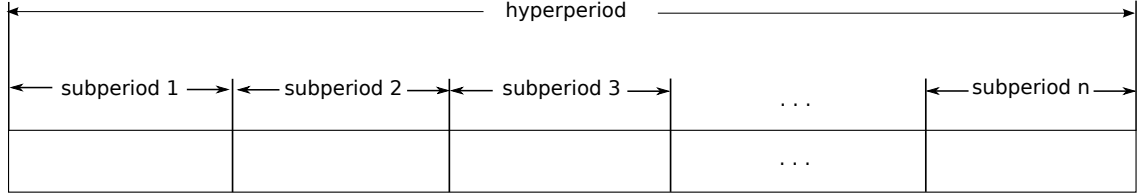


FIGURE 5.1: Basic TDMA structure.

- *Cyclic scheduling*: is a fixed length scheduling which repeats infinitely over the time.
- *Harmonic message set  $M$* : a message set is harmonic if and only if each message period is a positive integer multiple of all smaller message periods. Equivalently, each message period divides all larger message periods.
- *Power of 2 harmonic message set*: is an harmonic message set with all the possible periods equal to an integer power of 2. In the implemented algorithm the conversion is done using the following formula

$$T^h = 2^{\lfloor \log_2(T) \rfloor} \quad (5.2)$$

Notice that always  $T^h \leq T$ .

- *Utilization of a message set*: bound that determines if a message set is contention-free schedulable or not. It is defined by the following formula

$$U_M = \sum_{i=1}^{|M|} \frac{1}{T_i} \quad (5.3)$$

- $U_M > 1$ : the message set  $M$  is impossible to schedule, always requiring more slots than require in any given cycle length.
- $U_M \leq 1$ : if the different  $T$  are multiples, i.e.  $i < j \Rightarrow T_i | T_j$ , the message set  $M$  is schedulable.
- $U_M \leq 0.5$ : may always be scheduled by reduction to periods consisting solely of powers of 2.

Notice that the  $T_i$  value must be defined in number of slots units and not in time units.

- *The relative deadline*: specifies the length of time after release by which the message transmission must be complete. As the slot duration is adapted to the largest unit of time of all the possible messages, the deadline will be always inferior or equal to the slot duration. In other words, all the packets will be sent within a single slot.
- *The message unit of time*: is the time it takes to transmit one packet on the wireless network. We consider that all the messages have the same length, or in other words the slot duration will be equal to the largest unit of time of all the messages to have the possibility to allocate all the node messages using the same type of slot length.

### 5.3.2 TDMA structure.

The schema of the basic TDMA structure used by the algorithm is illustrated in Figure 5.1. It is composed by  $N$  subperiods that all together form an *hyperperiod*. The length of each subperiod is equal to  $T_{min}^h$  while the length of the hyperperiod is equal to  $T_{max}^h$ . By the properties of power of two harmonic periods the following relation is accomplished

$$T_{max}^h = C \cdot T_{min}^h \quad \text{with} \quad C = 2^n \quad \text{for all} \quad n \in \mathbb{Z} \quad (5.4)$$

or in other words, the total number of subperiods in the whole TDMA structure will be equal to

$$\text{Number of subperiods} = \frac{T_{max}^h}{T_{min}^h} = C = 2^n \quad (5.5)$$

Hence, the division between the hyperperiod length and the subperiod length will be always equal to an integer power of two. This allows to have a perfect periodic scheduling being trivial to allocate the messages, achieving a contention-free scheduling. Moreover, the scheduling is a cyclic scheduling with cycle period equal to the hyperperiod length.

The subperiods have an internal structure adapted to the possibility to share the TDMA frame among different clusters and also adapted to the SWAN's superframe structure. In this subdivision of the subperiod the higher structure is named *hyperframe* which length corresponds to the length of a subperiod, while the subdivisions of an hyperframe are the *superframes*. Each superframe corresponds to the superframe of a cluster, and these superframes are divided in *Downlink*, *Uplink* and *InterNodelink segments* as it is defined in the SWAN documents. The whole TDMA structure is divided into slots of length equal to the slot duration (see 5.4.1.2). The schema of this substructure is presented in Figure 5.2.

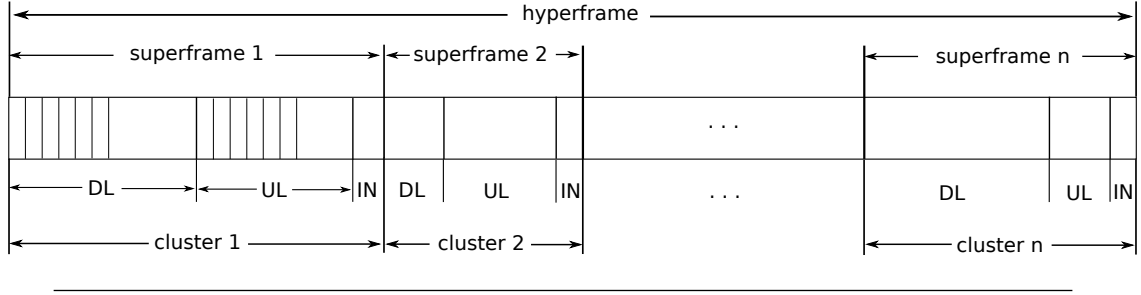


FIGURE 5.2: Subperiod structure.

To maintain the perfect periodicity of the node's messages, the distance between segments of the same cluster must be constant over the different subperiods and the node messages must occupy always the same position within these segments. This is achieved forcing that the length of the superframes of the same cluster over the different subperiods and the length of the segments over the different superframes must be constant. Notice that the length of the superframes of the different clusters may not be equal among them, or in other words, all the clusters may not have the same *capacity*. Also notice that the Downlink and Uplink segments of the same cluster have different lengths or different capacity. On the contrary, the length or capacity for the InterNodeLink segment is the same over all the clusters due the algorithm design definition. In resume, the lengths of superframes and segments of the same cluster are equal over subperiods but their length may be different among clusters.

The length or capacity of these substructures is calculated in the first part of the scheduling algorithm (see 5.4.2). Besides, in the section 'Analysis, experiments and results' are presented some examples (see 5.5).

### 5.3.3 TDMA frame sharing among clusters.

All the clusters of the SWAN system must work using the same frequency band due the requirements of the localization algorithm based on Received Signal Strength Indicator (RSSI), which states that the localization is more precise if all the nodes are running on the same frequency band; hence a combination of TDMA-FDMA access methods is discarded. Neither the nodes implement other channel access methods, such as CDMA, SDMA in combination with TDMA, that will make possible the simultaneous communication of the different clusters. Therefore, in the overlapping clusters working at the same frequency its internal communications may cause interferences or be interfered by the communications of the other clusters. The name of these interferences is *inter-cluster TDMA interferences* and a graphical representation is in Figure 5.3.

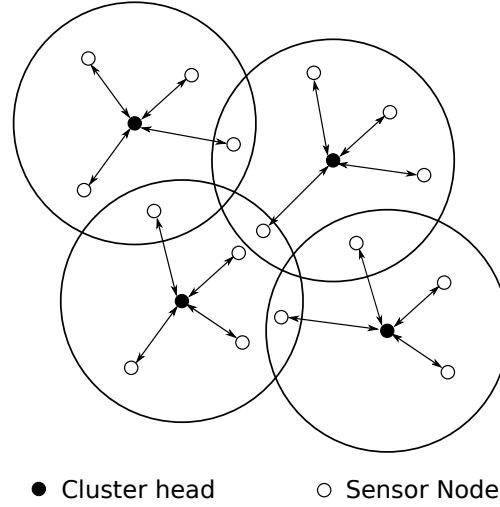


FIGURE 5.3: Inter-cluster TDMA interferences.

The straightforward and most simple solution to avoid this problem if no other channel methods can be implemented is to multiplex in time the communications of the different clusters; *for this reason the hyperframe is divided in superframes, each one corresponding to the internal communications of a particular cluster*. Notice that not all the clusters from the whole WSN should share the same TDMA frame, only those who may cause interferences among them. Given the contrary case, if two clusters are enough spaced that there is no possibility of interferences from one to the other, their own TDMA frames may be overlapped in time. In Figure 5.3 can be appreciated both cases. First, the top-left cluster and the bottom-right one are not overlapped in space and therefore their own TDMA superframes may be overlapped in time. On the contrary, the rest of cluster relations are prone to interferences and they should share a common TDMA frame multiplexing their own superframes in time.

The advantages of multiplexing the superframes in time are: the network is more reliable due that the inter-cluster interferences are avoided, and there is energy saving due the inactive periods between superframes. On the other hand, as drawback, there is a possible synchronization problem due the possibility to do not receive the synchronization frame within an optimal period [7]. Moreover, there are other techniques to improve the whole wireless network capacity reducing the inter-cluster TDMA interferences [8, 9] that may be studied in future works in this subject.

## 5.4 Scheduling algorithm implementation.

The scheduling algorithm has been implemented under the MATLAB platform, due it is an ideal platform to do the first steps of an algorithm design, implementation

and testing. The implemented scheduling algorithm consist of two differentiated parts, namely a *capacity calculation function* which determines if the scheduling is feasible or not, and the *main allocation algorithm* that allocates the messages of the different nodes into the slots of the TDMA frame. The first part, i.e. the capacity calculation function is not strictly necessary to allocate the messages into the TDMA frame but it is useful to perform previous operations to the well functioning of the main allocation algorithm as is explained later. A representation of the algorithm flow chart is pictured in the Figure 5.4.

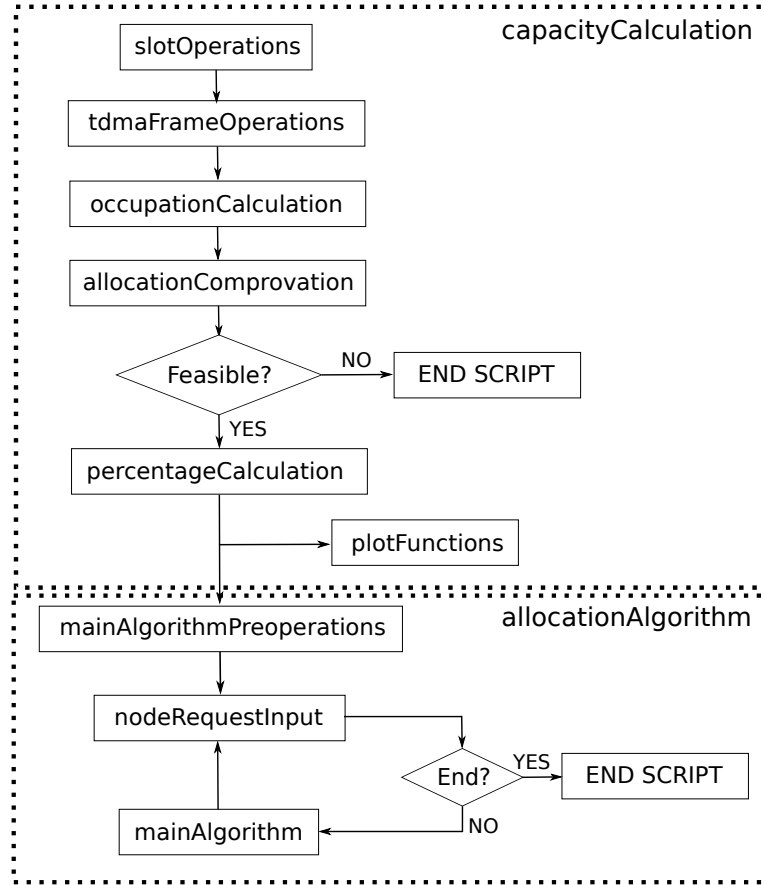


FIGURE 5.4: Flow Chart of the Scheduling Algorithm.

#### 5.4.1 Input parameters.

The input parameters must be defined before running the scheduling script.

##### 5.4.1.1 Input Matrix.

The **Input Matrix** is a data structure which defines the number of nodes and their characteristics, that will form part of the WSN. The values of the **Input Matrix** must

be considered as an estimation or expected value, due in principle the exact number of nodes that will form the WSN are not known in advance. The way to estimate these values is out of the scope of this thesis.

The format of the **Input Matrix** is defined in the Table 5.1 with some example values.

Quantity of nodes	Node type	Requested period	Requested ACK	Cluster
100	Sensor	2	Yes	1
70	Actuator	5	No	2
50	Actuator	1	No	1
80	Sensor	1	Yes	3

TABLE 5.1: Input Matrix table.

- **Quantity of nodes:** indicate the number of nodes forming a set that will run the same application.
- **Node type:** in the SWAN system there exist two types of nodes, sensor nodes and actuator nodes.
- **Request period:** the time period in seconds requested by the specific application that will run in this set of nodes.
- **Requested ACKs:** indicate if the nodes require acknowledgement or not.
- **Cluster:** indicate the cluster to which the nodes are associated to.

The drawback that can bring to do not have a good estimation of the possible nodes that will join to the different clusters is the possibility to do not have enough capacity to allocate all the node's messages within the TDMA frame. Therefore, it will be needed a *control system* to do not overload the TDMA frame and if it happens, that is to have more node messages than capacity, propose some solution.

#### 5.4.1.2 Slot Parameters.

The slot parameters define the **slot duration** and they are the **number of bits per symbol**, the **data rate** and the **base slot duration**. The two first parameters, namely the **number of bits per symbol** (*bits/symbol*) and the **data rate** (*bps*), are determined by the hardware, specifically the transceiver, that will be used. The **base slot duration** (*symbols*) parameter is determined by the maximum length of the packets that will be transmitted.

The **slot duration** is calculated using the previous parameters and it defines the duration in seconds of one slot of the TDMA frame. Straightforward is noticed that the **slot**



`duration` value is inversely proportional to the number of slots that would be allocated in a fixed length TDMA frame.

The formula to calculate the `slot duration` is the following

$$Slot\ Duration = Base\ Slot\ Duration \cdot Symbol\ Length \quad (5.6)$$

while the symbol length is equal to

$$Symbol\ Length = \frac{1}{\frac{Data\ Rate \cdot 1000}{Bits\ Per\ Symbol}} \quad (5.7)$$

#### 5.4.1.3 Occupation percentage of the InterNodelink segment.

It defines the occupation percentage of the InterNodelink segment inside a fixed length superframe. The output of the calculation is the integer number of slots assigned to the InterNodelink period in each superframe.

#### 5.4.2 Capacity calculation.

The `capacityCalculation` function is the first function executed in the allocation script and its goal is to determine if the scheduling of the messages is feasible or not. This determination is done considering the input parameters and the hyperframe maximum capacity. As it is stated before (see 5.3.2) the hyperframe capacity is in function of the  $T_{min}^h$  period.

The `capacityCalculation` function has different nested functions, each one with a different purpose. Their execution is sequential, as is shown in the Figure 5.4. In the following list is explained the purpose of the nested functions:

- `slotOperations`: calculate the slot duration.
- `tdmaFrameOperations`: extract the *minimum requested period*  $T_{min}$  and the *maximum requested period*  $T_{max}$  from the Input Matrix, calculate the number of slots corresponding to these periods ( $T_{min}^s, T_{max}^s$ ) and convert them to harmonic periods ( $T_{min}^h, T_{max}^h$ ). From these two last variables is calculated the number of subperiods (see Eq. 5.5). Also it is extracted from the Input Matrix the number of clusters that will share the TDMA hyperframe.
- `occupationCalculation`: calculate the occupation in number of slots of each type of message, namely *actuator* messages, *actuatorACK* messages, *sensor* messages

and *sensorACK* messages. Therefore, is calculated the probability of subperiod occupation (see Eq. 5.8). From the previous variables is determined the occupation of the downlink and the uplink segments of each cluster.

The *probability of subperiod occupation* (Eq. 5.8) is the occupation probability of a specific subperiod that has a message with a specific harmonic period.

$$\text{Subperiod Occupation} = \frac{T_{min}^h}{T^h} \quad (5.8)$$

- **allocationComprovation:** check if the scheduling of all the messages defined by the previous function is feasible. Basically all the slots occupied from all the clusters are added up and compared with the length of an hyperframe  $T_{min}^h$ . If the allocation is not feasible the script stops in this point informing about it with an error message box, but if it is feasible the script continues.
- **percentagesCalculation:** distribute the free slots remaining in the TDMA hyperframe among all the superframes in an uniform way and calculate the total number of slots (occupied and free) of downlink, uplink and internodelink segments of each superframe. Later calculate the occupation percentage of the hyperframe by each superframe and calculate the occupation percentage of the superframe by each segment.
- **plot1:** plot the number of slots occupied by each type of message within the superframes of each cluster.
- **plot2:** plot the the number of slots of each segment within the superframes of each cluster.
- **plot3:** plot the occupation percentage of the hyperframe by each superframe and the occupation percentage of each superframe by each segment.

#### 5.4.2.1 Slot occupation calculation.

1. The number of slots occupied are calculated cluster by cluster, so the algorithm first differentiate the rows of the **Input Matrix** corresponding to the same cluster.
2. Secondly the script extract the period value in seconds of a set of nodes and converts it to an harmonic period in number of slots:

$$\begin{aligned} T^s &= \left\lfloor \frac{T}{slotDuration} \right\rfloor \\ T^h &= 2^{\lceil \log_2(T^s) \rceil} \end{aligned}$$

3. Thirdly is determined the probability to occupy a slot within a subperiod in function of the node's harmonic period. This probability is equal to:

$$\text{SubperiodOccupation} = \frac{T_{min}^h}{T^h}$$

Remember that  $T_{min}^h$  and  $T^h$  are always power of 2 values and  $T_{min}^h \leq T^h$ .

4. Fourthly is calculated the total number of slots that will occupy a set of nodes with a determined period and it is added to the total. Notice that the type of node, i.e. actuator or sensor, is differentiated.

$$\begin{aligned} temp &= \left\lceil numberNodes(T^h) \cdot probabilityOccupation(T^h) \right\rceil \\ totalNumberSlots(type) &= temp + totalNumberSlots(type) \end{aligned}$$

5. Later is checked if this set of nodes require acknowledgement, and if they require sum to the total. Observe that the number of ACKs occupied is equal to the previous number of slots occupied  $temp$ .

$$totalNumberACK(type) = temp + totalNumberACK(type)$$

6. Finally, the total number of slots occupied in each segment of each superframe is computed.

$$OccupationDL = numActuators + numSensorsACK$$

$$OccupationUL = numSensors + numActuatorsACK$$

#### 5.4.2.2 Occupation percentages calculation.

1. Before calculating the occupation percentages, the free slots that have not been occupied are distributed in an uniform way among all the superframes within an hyperframe. The number of free slots is

$$totalSlotsFree = T_{min}^h - totalSlotsOccupied$$

2. The occupation percentage of the hyperframe by each superframe is

$$perCell = \frac{numSlotsDL + numSlotsUL + numSlotsIN}{T_{min}^h}$$

The occupation percentage is equal to the sum of the lengths of all the segments (Dowlink, Uplink and Internodelink), which corresponds to the superframe length, divided by the total length of the hyperframe or subperiod, which is equal to  $T_{min}^h$ .

3. The occupation percentage of the superframe by each segment is

$$perDL = \frac{numSlotsDL}{numSlotsDL + numSlotsUL + numSlotsIN}$$

$$perUL = \frac{numSlotsUL}{numSlotsDL + numSlotsUL + numSlotsIN}$$

$$perIN = percentageIN$$

The occupation percentage is equal to the length of the segment divided by the total length of the superframe. Perceive that the occupation percentage of InterN-odelink segment is determined as an input variable.

### 5.4.3 Allocation algorithm.

The `allocationAlgorithm` function is executed after `capacityCalculation` function if the allocation is feasible. Its aim is to execute the scheduling algorithm to allocate the messages from all the nodes. It has three nested functions explained in the following lines which execution is not sequential as it is shown in Figure 5.4.

- **mainAlgorithmPreoperations:** setup the main variables which will be used in the main algorithm. These variables are:
  - The number of slots of each superframe.
  - The slot number where each segment starts.
  - Pointers that indicate the next free slot in each segment.
- **nodeRequestInput:** the main goal of this function is to simulate the Association Requests of the nodes to perform the slot allocation node by node. The parameters of each node, namely node type, requested period, acknowledgement request and cluster to within perform the association, are extracted randomly from each row of the `Input Matrix` and transfered to the allocation `mainAlgorithm` function.
- **mainAlgorithm:** this is the scheduling algorithm itself. It performs the allocation of the node's messages in the TDMA frame. The algorithm differentiate between actuator nodes and sensor nodes allocating them in the corresponding segments. Also it allocates the acknowledgement messages if they are requested.

### 5.4.3.1 Superframe length calculation.

To calculate the length of the superframe of each cluster, the total hyperframe length  $T_{min}^h$  is multiplied by the occupation percentage of the cluster  $perCell(cell)$ . Observe that the floor function is applied to the result, in this way the sum of the length of all the superframes will not be higher than the total hyperframe length.

$$lengthCell(cell) = \left\lfloor T_{min}^h \cdot perCell(cell) \right\rfloor$$

### 5.4.3.2 The calculation of the slots where each segment start.

The slots where Downlink, Uplink and InterNodelink segments start are only calculated within the first subperiod. If is needed the calculation in the rest subperiods, it is trivial only needing to sum the length of the hyperframe  $T_{min}^h$  to the first subperiod value.

Calculation in the first cluster (first superframe):

$$\begin{aligned} startDL(1) &= 1 \\ startUL(1) &= startDL(1) + \lceil lengthCell(1) \cdot perDL(1) \rceil \\ startIN(1) &= startUL(1) + \lceil lengthCell(1) \cdot perUL(1) \rceil \end{aligned}$$

And in the rest of the clusters (rest of superframes):

$$\begin{aligned} startDL(cell) &= startDL(cell - 1) + lengthCell(cell - 1) \\ startUL(cell) &= startDL(cell) + \lceil lengthCell(cell) \cdot perDL(cell) \rceil \\ startIN(cell) &= startUL(cell) + \lceil lengthCell(cell) \cdot perUL(cell) \rceil \end{aligned}$$

### 5.4.3.3 Determine the initial position of the pointers that indicate the next free slot of each segment.

There are two type of pointers accordingly to the two types of slot allocation:

- Pointers for the perfect periodic slots which correspond to the node's messages allocation.
- Pointers for the non-perfect periodic slots which correspond to the node's ACKs allocation.

For the perfect periodic slot allocation type, the algorithm only requires to know the value of the pointer within the superframe of the first subperiod. It is calculated like

this:

$$\varphi_{DL}(cell) = startDL(cell)$$

$$\varphi_{UL}(cell) = startUL(cell)$$

The  $\varphi_{DL}(cell)$  and  $\varphi_{UL}(cell)$  are the variables where the values of the next free slot in Downlink and Uplink segments are stored. Notice that it is not calculated the pointer of the InterNodelink segment, due the allocation into this segment is not considered by this version of the algorithm.

For the non-perfect periodic slot allocation type it is necessary to know the value of the pointers within all the superframes in all subperiods. It is calculated like this:

$$for\ j = 0 : numSubperiods - 1$$

$$\varphi_{DL}^{Ack}(cell, j + 1) = j * T_{min}^h + startUL(cell) - 1$$

$$\varphi_{UL}^{Ack}(cell, j + 1) = j * T_{min}^h + startIN(cell) - 1$$

The variables are defined using a matrix where the rows correspond to the clusters and in the columns of each row are stored the pointer values for each subperiod.

The initial position of the perfect periodic type pointers is located at the beginning of the segments, while the initial position of the non-perfect periodic type is located at the end of the segments. The value of the pointer increase in perfect periodic type or decrease in non-perfect periodic type while the algorithm is running and is using slots to allocate new messages. In the Figure 5.5 is presented an schema about the pointers and the segment start slots.

The motivation to use these two types of slot allocation, i.e. perfect periodic and non-perfect periodic, is to increase to the maximum the superframe capacity having the maximum number of slots to allocate the node's messages, due there are no permanent gaps, i.e. unused slots, between to consecutive occupied slots of the same type. In contrast, there is the possibility to program the algorithm to handle all the messages, namely node messages and acknowledgements, in a perfect periodic way. The benefit is that for determine the slots awarded to each node only is needed to know the offset of the slot and the message period in both types of messages, i.e. node messages and acknowledgements, being easy to program the MAC layer as well as less memory usage to store this information. The drawback is that the capacity of the superframe to allocate messages will decrease, due there will be permanent gaps between two consecutive acknowledgement messages if some nodes do not require them.

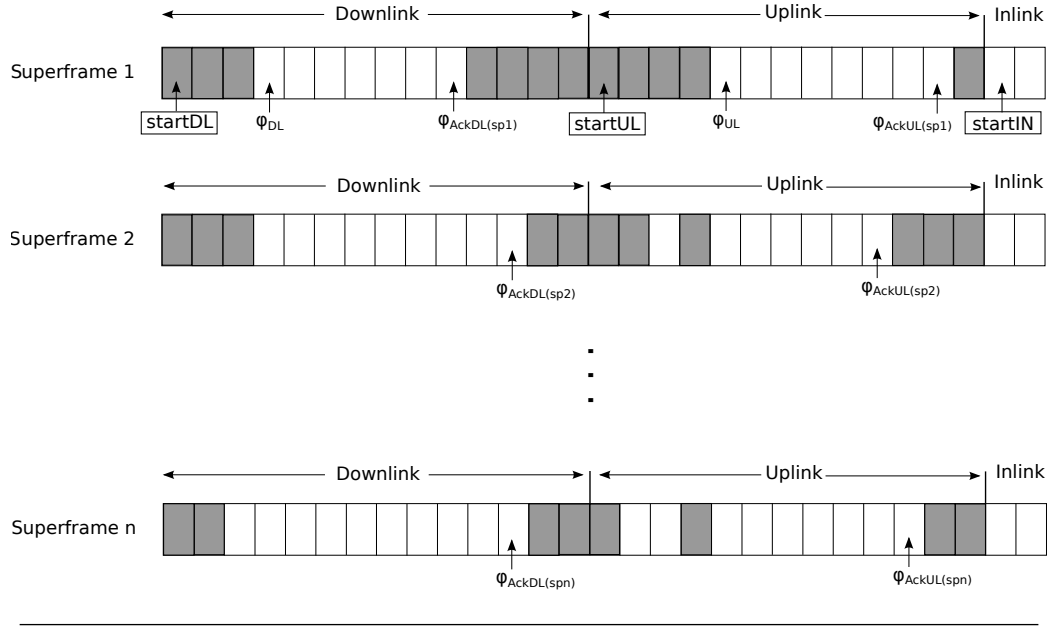


FIGURE 5.5: Schema of the frame pointers and the segment start slots.

#### 5.4.3.4 Extract data from the Input Matrix to simulate the node request.

To simulate the node requests to be more close to the real behavior, that is to allocate the nodes one by one at the node requests rate in a random order, the algorithm implements the `nodeRequestInput` nested function. In this function the node requests are simulated selecting a row from the `Input Matrix` in a random fashion and extracting the values of the actual node (type, period, acknowledgement request and cluster) and sending it to the `mainAlgorithm` nested function. Moreover, it has a control mechanism to detect when all the nodes have been extracted from the `Input Matrix`, sending control signals to the main loop indicating that the program must finish its execution.

#### 5.4.3.5 Main algorithm behavior.

To well understand the algorithm it is strongly recommended to study the Figure 5.6. In this figure it is presented how the algorithm schedules the node messages and the acknowledgements in function of its period and the node type. For simplicity has been taken a  $T_{max}^h/T_{min}^h$  relation of 4, therefore there are 4 subperiods within the hyperperiod which correspond to 4 superframes per cluster.

Referring to the TDMA structure, observe that only are considered the superframes of a particular cluster, each one of them located within a different subperiod; thus superframe 1 is located in subperiod 1, superframe 2 is located in subperiod 2, etc. As it is explained in 5.3.2 the lengths of all the superframes of the same cluster and the lengths of all the

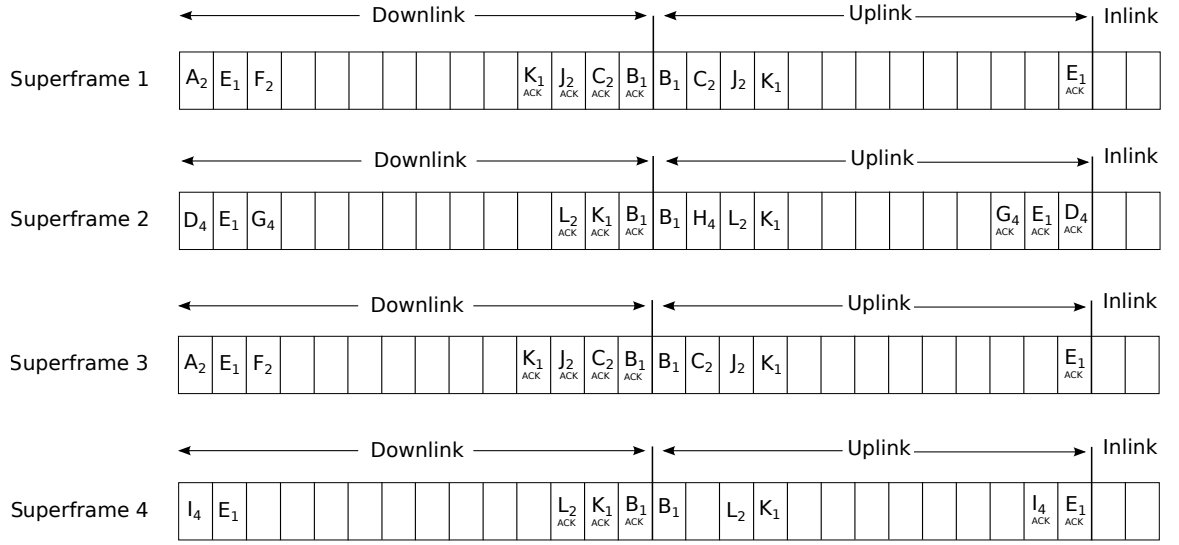


FIGURE 5.6: Basic example of message scheduling.

segments within a superframe are equal over subperiods. Notice that the distance from a specific slot within a specific superframe and the slot in the same position in any other superframe will be multiple of the hyperframe length value  $T_{min}^h$ .

Following the SWAN superframe structure requirements, the actuator node messages are allocated in the left side of the Downlink segment and the sensor node messages are allocated in the left side of the Uplink segment. On the other hand, the actuator node acknowledgements are allocated in the right side of the Uplink segment and the sensor node acknowledgements are allocated in the right side of the Downlink segment.

The letter defining an occupied slot means the allocation order, being 'A' the first or older and 'L' the last or newer allocated message. The number as a subindex of the letter means the period relation  $T^h/T_{min}^h$  of the node. Notice that all the periods relations are an integer power of 2.

The nodes that do not have acknowledgement allocation is due they do not require them. Observe that may be free slots within the superframes which may be used for future perfect periodic or non-perfect periodic message allocations.

#### ***Allocation of perfect periodic messages (node messages).***

1. First, the algorithm converts the requested period to an harmonic period and differentiates the type of node.
2. Later, it checks if there is room to start the allocation in any subperiod different from the first subperiod, otherwise the allocation will start in the first subperiod. This checking is done comparing if there exist free slots in the subperiods different



from the first which position in the superframe is lower than the pointer that determines the next free slot in the first subperiod. The maximum subperiod that a message with a determined period may start its occupancy depends on a bound defined by the  $T^h/T_{min}^h$  relation. For instance, a message of  $T^h/T_{min}^h = 1$ , can not start the occupancy in subperiods higher than the first subperiod, while messages of  $T^h/T_{min}^h = 2$  can start its occupancy in the first and second subperiod and messages of  $T^h/T_{min}^h = 4$  can start its occupancy from subperiod 1 to subperiod 4, and so on. This mechanism permits the full occupation of the TDMA frame efficiently.

3. Once the starting point has been selected, it is checked how many subperiods of the hyperperiod the message will occupy using the relation  $T_{max}^h/T^h$ . In function of this value the algorithm iterate over the TDMA frame marking the selected slots as occupied with the node ID in the corresponding subperiods. The distance between two consecutive slots occupied by the same node will be always equal to the node harmonic period value  $T^h$ .

#### *Allocation of non-perfect periodic messages (acknowledgements).*

The acknowledgements are allocated in the same subperiods that the node messages occupy. As each subperiod has its own pointer that determine the next free slot, the algorithm allocate these acknowledgements in function of this variable value. The consequence is that the distance between two consecutive occupied slots may not be equal to the  $T^h$  value.

#### **5.4.4 Output parameters.**

The output parameter is a vector named **Schedule** which length is equal to the hyperperiod length and each position in the vector represents a slot. The slots occupied are marked with the node ID. Notice that the vector is not formatted to differentiate the sections which define the superframes of the different clusters nor the sections that differentiate the different subperiods.

## 5.5 Analysis, experiments and results.

In this section are presented some experiments carried out using the scheduling algorithm script to demonstrate its functioning and to present some results.

### 5.5.1 Experiment 1.

In this experiment is presented a basic configuration of a SWAN WSN in Table 5.2.

Quantity of nodes	Node type	Requested period	Requested ACK	Cluster
100	Actuator	1	Yes	1
50	Sensor	2	No	1
76	Actuator	6	Yes	2

TABLE 5.2: Input Matrix of Experiment 1.

Analysing the table we see that the TDMA frame is shared between two different clusters. The sum of all the nodes is equal to 226. The minimum requested period is equal to 1 *sec* and the maximum requested period is equal to 6 *sec*.

All the periods have to be converted to power of two harmonic periods in number of slot units. The minimum and the maximum harmonic periods define the base structure of the TDMA frame.

First the slot duration should be calculated

$$Symbol\ Length = \frac{1}{\frac{Data\ Rate \cdot 1000}{Bits\ Per\ Symbol}} = \frac{1}{\frac{250 \cdot 1000}{4}} = 1.6 \cdot 10^{-5} sec$$

$$Slot\ Duration = Base\ Slot\ Duration \cdot Symbol\ Length = 60 \cdot 1.6 \cdot 10^{-5} = 9.6 \cdot 10^{-4} sec$$

The *Data Rate*, *Bits Per Symbol* and *Base Slot Duration* are typical values for IEEE-802.15.4 communications.

Now the maximum and minimum harmonic periods can be calculated

$$\begin{aligned} T_{max}^s &= \left\lceil \frac{T_{max}}{slotDuration} \right\rceil = \left\lceil \frac{6}{9.6 \cdot 10^{-4}} \right\rceil = 6250\ slots \\ T_{max}^h &= 2^{\lceil \log_2(T_{max}^s) \rceil} = 2^{\lceil \log_2(6250) \rceil} = 4096\ slots \end{aligned}$$

$$\begin{aligned} T_{min}^s &= \left\lceil \frac{T_{min}}{slotDuration} \right\rceil = \left\lceil \frac{1}{9.6 \cdot 10^{-4}} \right\rceil = 1041\ slots \\ T_{min}^h &= 2^{\lceil \log_2(T_{min}^s) \rceil} = 2^{\lceil \log_2(1041) \rceil} = 1024\ slots \end{aligned}$$

So the *hyperperiod* length will be equal to  $T_{max}^h = 4096 \text{ slots}$  and the *subperiod* length will be equal to  $T_{min}^h = 1024 \text{ slots}$ .

The number of subperiods will be equal to

$$\text{Number of subperiods} = \frac{T_{max}^h}{T_{min}^h} = \frac{4096}{1024} = 4$$

The *capacityCalculation function* determine if the allocation is posible in function of the characteristics and quantity of nodes, and the hyperframe capacity which is equal to the subperiod length. The results generated in *capacityCalculation function* are presented in Figure 5.7:

- *Message Box*: it indicates that the scheduling is possible, the capacity of the hyperframe, the total number of slots occupied and the remaining free slots. Notice that the number of slots occupied is in function of the quantity of nodes, the requeriment or not of ACKs, and the probability of occupation in function of the node period and  $T_{min}^h$ .
- *Superframe occupied slots*: indicate the number of slots occupied in the superframes of the different clusters and the type of messages, i.e. Actuator messages, Sensor ACKs messages, Sensor messages and Actuator ACKs messages, that occupy the slots. The red line is a reference point to indicate that the occupation of the superframes should be lower than this value.
- *Superframe distribution*: indicates the final length of the superframes and the segments of the different clusters. This length is equal to the sum of the occupied slots presented in the previous plot and the free or unoccupied slots. The sum of the lengths of all the superframes is equal to the hyperframe length.
- *Occupation percentage*: indicate the occupation percentages of the superframes and the different segments. These values are the ones who are transfered to the *allocationAlgorithm function* to build up the real TDMA frame to be able to allocate the messages.

The slot allocation is not presented because it is not feasible due the length of the vector that represents it. Anyway it is added to the package the *Schedule* vector in '.mat' format to perform further studies using Matlab.

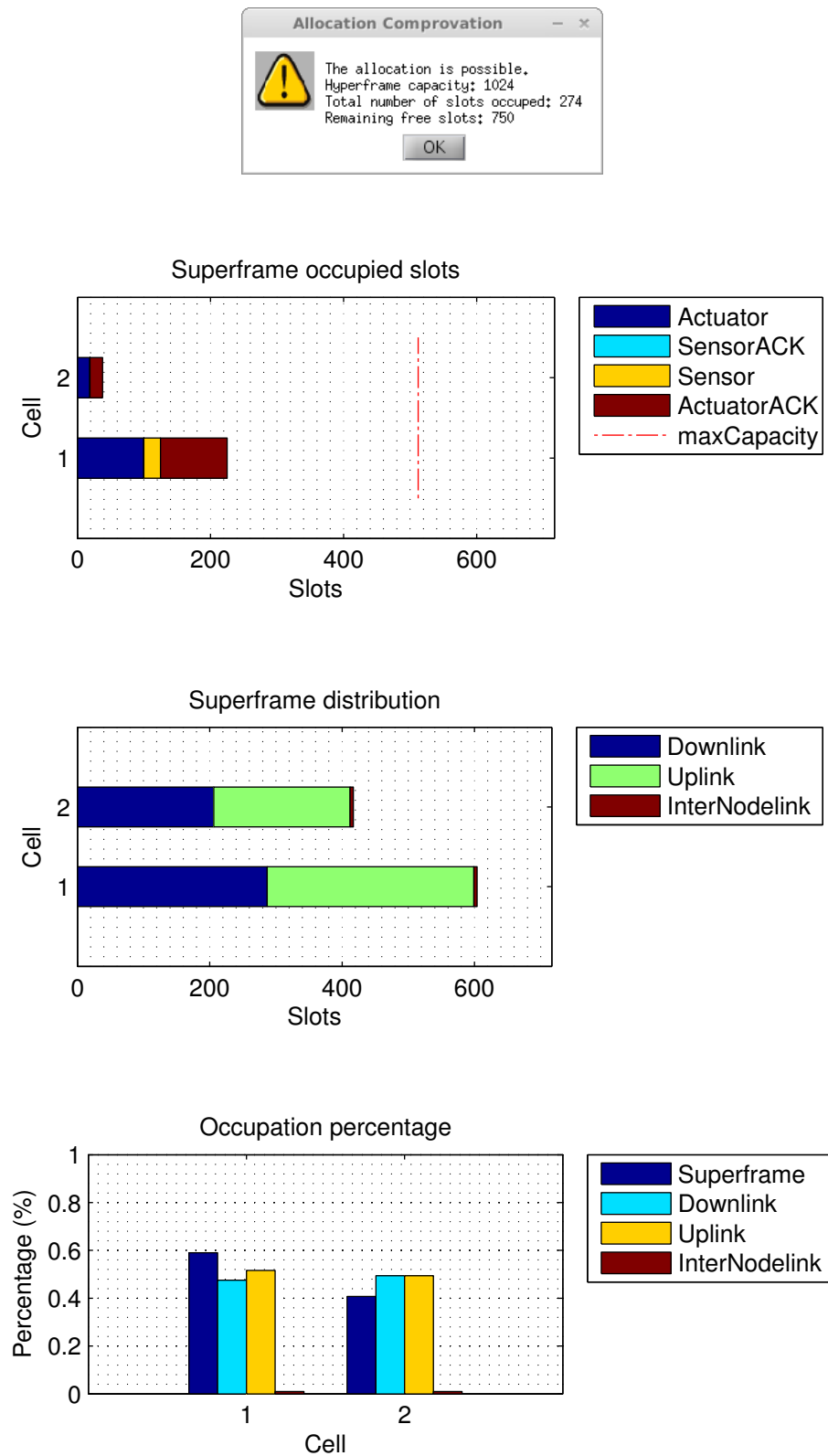


FIGURE 5.7: Plots of Experiment 1.

### 5.5.2 Experiment 2.

In this experiment a more complex SWAN WSN is built. The configuration of this WSN is in Table 5.3.

Quantity of nodes	Node type	Requested period	Requested ACK	Cluster
100	Sensor	2	Yes	1
30	Sensor	5	No	1
40	Sensor	8	Yes	2
60	Sensor	5	No	3
60	Sensor	2	Yes	4
25	Sensor	5	No	5
70	Actuator	8	Yes	6
50	Actuator	8	Yes	4
30	Sensor	5	No	3
40	Sensor	2	Yes	3
60	Actuator	2	No	1
60	Actuator	5	Yes	5
25	Sensor	8	No	6
70	Actuator	2	Yes	4
50	Sensor	5	Yes	6
70	Actuator	2	No	5
50	Actuator	5	No	3
80	Sensor	2	No	1
40	Sensor	8	Yes	1
25	Actuator	8	No	7
50	Sensor	5	Yes	7
70	Actuator	2	Yes	7
40	Sensor	2	Yes	8
60	Actuator	2	Yes	1

TABLE 5.3: Input Matrix of Experiment 2.

The process is the same than in Experiment 1 but with different parameters. The remarkable properties of this experiment are:

- The total number of clusters to share the TDMA frame is equal to 8.
- As the minimum period is equal to 2 seconds, the hyperframe capacity is higher being equal to 2048 slots.
- Although the length of the different superframes and its internal structure is very different from Experiment 1, the scheduling is possible maintaining the perfect periodic and contention-free properties.
- Still 1/4 of the hyperframe capacity is available after allocating all the messages from the nodes presented in the table.

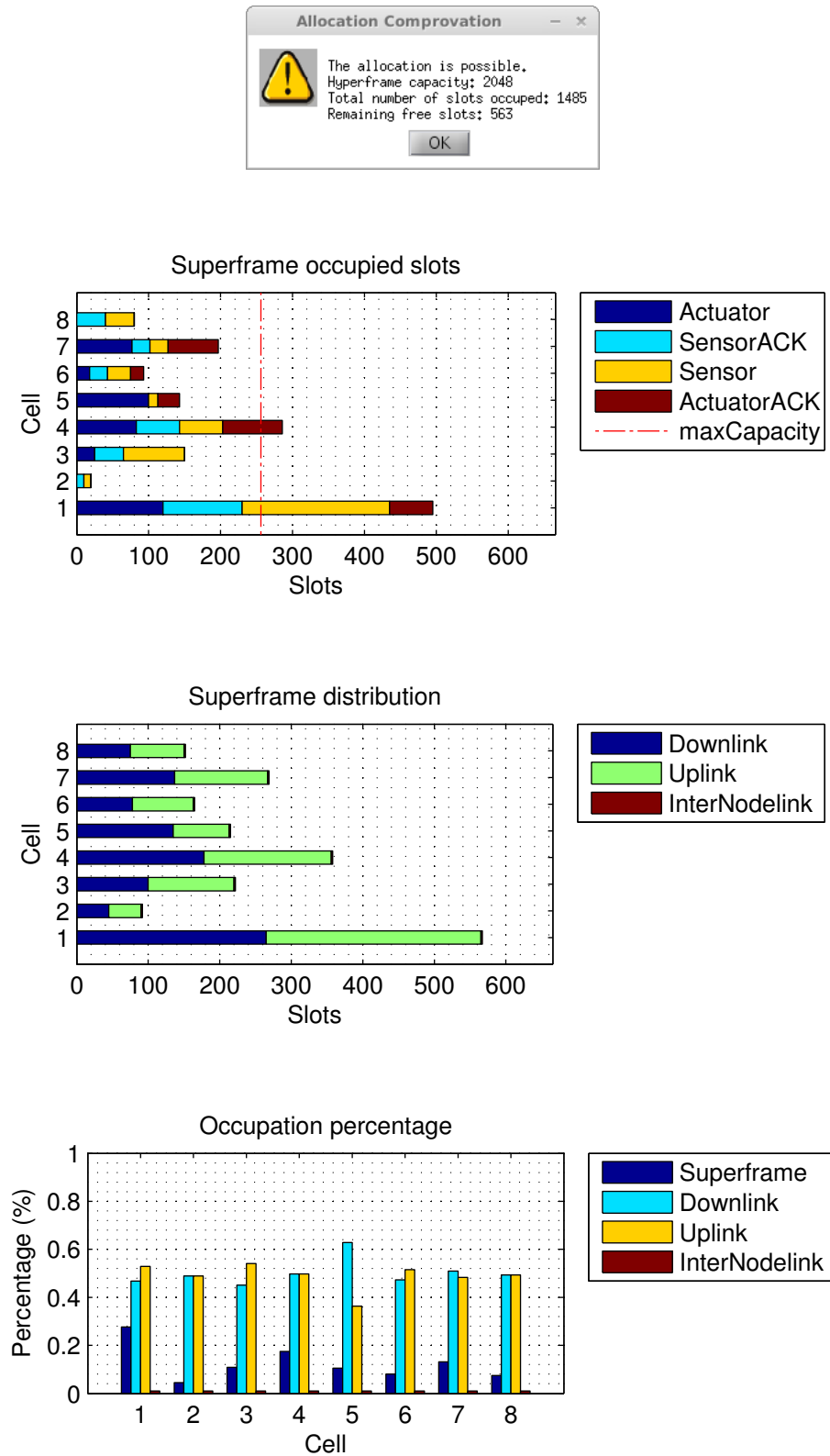


FIGURE 5.8: Plots of Experiment 2.

### 5.5.3 Experiment 3.

In Table 5.4 is presented a basic SWAN WSN configuration.

Quantity of nodes	Node type	Requested period	Requested ACK	Cluster
100	Actuator	1	Yes	1
150	Sensor	2	No	1
90	Actuator	0.5	Yes	2
65	Sensor	3	Yes	2
80	Sensor	1	Yes	3
40	Actuator	3	No	3

TABLE 5.4: Input Matrix of Experiment 3.

This experiment is to show that if the minimum requested period is low, in this case is equal to 0.5 seconds, the hyperframe capacity will be more limited, being in this case equal to 512 slots.

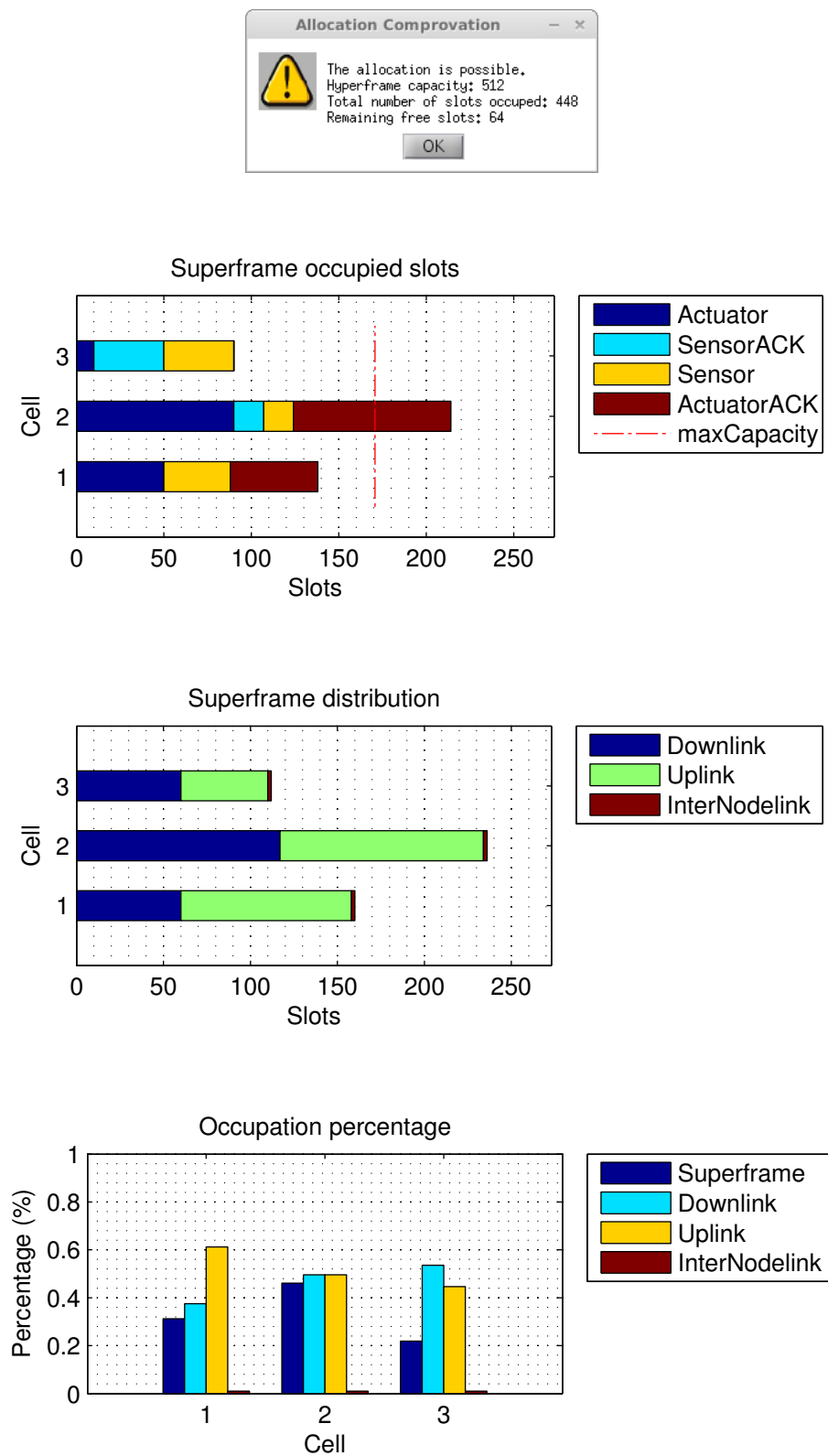


FIGURE 5.9: Plots of Experiment 4.



#### 5.5.4 Experiment 4.

The last experiment presented in Table 5.5 is to show that in some cases the scheduling is not possible.

Quantity of nodes	Node type	Requested period	Requested ACK	Cluster
100	Actuator	1	Yes	1
30	Sensor	2	No	1
200	Actuator	1	Yes	2
60	Sensor	3	No	2
140	Sensor	1	Yes	2
25	Actuator	3	No	1
70	Sensor	1	Yes	1

TABLE 5.5: Input Matrix of Experiment 4.

Observe in Figure 5.10 that number of slots occupied exceeds the hyperframe capacity.



FIGURE 5.10: Plots of Experiment 4.

## 5.6 Conclusions and open problems.

A contention-free perfect periodic scheduling algorithm based on power of two harmonic periods has been implemented. It is adapted to the following SWAN Medium Access Control requirements:

- The superframe division in Downlink, Uplink and InterNodelink segments.
- The sharing of the TDMA frame among several clusters.
- The type of nodes of the SWAN WSN, i.e. sensor nodes and actuator nodes.
- The type of messages that the algorithm has to allocate, i.e. node messages and acknowledgements.
- The MAC communication protocol, i.e. Association Request/Association Response.

It has been demonstrated that the algorithm is able to allocate the node messages following a perfect periodic scheduling schema and to allocate the acknowledgements in a non-perfect periodic schema with the aim to increase to the maximum the superframe capacity. It permits to use all the slots of the superframe allowing to have a considerable number of sensor/actuator nodes within a cluster.

Also, it permits the sharing of the whole TDMA frame among several clusters that are close enough to cause inter-cluster interferences among them, allowing to maintain the perfect periodic scheduling over all the clusters avoiding the inter-cluster interferences by multiplexing the superframes in time.

Furthermore, the scheduling is done in a discrete way allocating the nodes one by one in a random way, trying to simulate a real scenario where the nodes perform the Association Requests to the coordinator, and after the coordinator, as a reply to the Association Request, sends the Association Response with the number of the slots awarded to each node.

In addition, it has been implemented a function to calculate the maximum capacity of the TDMA superframes and to determine if the allocation will be possible in function of the input parameters such as the Input Matrix and the slot properties.

However, there are some limitations that may require a further study:

- The algorithm has to know from beforehand the periods of the nodes to schedule, specifically the minimum and the maximum periods, to be able to build up the whole TDMA frame before starting the allocations.

- The length or capacity of a superframe is strongly determined by the minimum period of the nodes that will be allocated into, due to higher packets delivery frequencies (lower periods) more data rate needed and therefore more slot occupation.
- The periods of the node messages should be converted to power of two harmonic periods to achieve the perfect periodic and contention-free scheduling schema. Therefore, the approximation may not be good and the granted periods may be different or very different from the requested periods.
- The nodes when are handling the perfect periodic messages only need to know three parameters to define the scheduling of these messages, i.e. the offset, the period and/or the subperiods which are being occupied. On the other hand, the non-perfect periodic messages require to define the awarded slots one by one with the impossibility to use a reduced representation as in the perfect periodic messages. This will cause an increment in signaling overhead to transfer the information of the awarded slots from coordinator to sensors, with the need to modify the actual SWAN MAC design regarding the Association Response command.
- In addition, the problem stated in the previous point will increase the memory requeriments to save a more complex scheduling representation, and specially in the coordinator side due it has to know the whole TDMA table.
- The programation of the MAC layer in the nodes may be a challenge if the slots are non-perfect periodic, due the slot assignment may change from one superframe to the other.
- It is necessary to add another control signal within the beacon frames to indicate in which subperiod (superframe) of the whole hyperperiod the system is running at each moment. The nodes should differentiate this signal to adapt its MAC layer, specifically the slot allocation in each different subperiod (superframe).

Appart from the problem listed above there are some other points that may be considered in following studies:

- Implement a script to convert from slot absolute positions in the Schedule vector to slot relative positions spliting the Schedule vector in subvectors each one corresponding to a different superframe of a different cluster.
- Study the efficient period of the synchronization message.
- Study mechanisms to combat the inter-cluster interferences with the aim to improve the capacity and performance of the whole WSN.

- Study an algorithm to improve the approximation, i.e. to approximate the granted periods to the requested ones.

## Chapter 6

# Conclusions.

This thesis is a part of a German research project named Sensor Wireless Application Network (SWAN), which objective is to study, implement and test WSNs on-board aircrafts. During this thesis it has been studied and presented an hybrid MAC sublayer protocol, based on the IEEE 802.15.4 LR-WPAN standard. The MAC sublayer is a complex issue that requires a precise job development.

The workflow has been divided in various stages. During the first stage it has been studied the SWAN system focusing on the wireless part. In the second stage, it has been studied several MAC sublayer protocols for WSN, the SWAN's MAC protocol design and its characteristics. During the third stage, it has been implemented one out of two running modes of the MAC sublayer, named Maintenance Mode, in which the nodes are able to associate to the WPAN. In the fourth stage it has been studied, designed and implemented a TDMA scheduling algorithm conformed to SWAN's MAC sublayer requeriments. The fifth and last stage has been the writing of this document.

The MAC sublayer implemented is an hybrid protocol which uses a contention-based approach based on the CSMA/CA mechanism to allow the nodes associate to the WPAN and permit network scalability, and uses a reservation-based approach based on TDMA to allow the communication between peers during the downlink and the uplink periods once the nodes are already associated to the network, due it is more energy efficient, ensure fairness, reduce collisions and assures a minimum bandwidth. Furthermore, the design and implementation of MAC protocols in WSN is not trivial. As the WSN devices are constrained regarding processing capabilities and limited batteries, and as the wireless medium is broadcast by nature, their design must avoid as far as possible the effects impacting on energy consumption. In all the cases, the nodes making up the WPAN are synchronized using beacons.

Besides the access method to the shared medium that is explained in the previous paragraph, the MAC sublayer's communication protocol is also important. It is defined by the MAC frames and the MLME commands. Both of them, MAC frames and MLME commands, determine the capabilities of the MAC layer and the actions and procedures that it can develop, such as association, disassociation or ranging among others. The frame fields and the MLME commands are configurable when designing a particular MAC protocol, therefore the actions or procedures in the MAC sublayer level required by any application can be achieved by the creation, modification or extension of the previous frame fields and the MLME commands.

As the behavior of the computer networks are perfectly modelled by asynchronous events, the MAC sublayer has been implemented under the event-driven programming paradigm. The events present in the MAC sublayer are split in four groups, i.e. timer events, reception of frames, transmission of frames and reception of control commands from upper layers. In addition, during the implementation it was necessary to determine the MAC sublayer parameters, i.e. the FSM states, the MAC sublayer constants and the MAC sublayer attributes. All these parameters are the most important variables which define, model and manage the behavior of the MAC sublayer. Furthermore, there are various important points such as the communication with the upper layers, the possibility of various clusters or a scan period that were not considered in the main design. It has been demonstrated that the implementation works and there have been presented some results and some conclusions regarding the MAC sublayer performance.

Finally, to achieve the application requirements of the SWAN system, a TDMA scheduling algorithm has been designed and implemented. As the applications need to send or receive a message periodically from the sensor/actuator nodes, its design has been based in two main characteristics, it is perfect periodic and contention free. Also it has been considered the possibility to share the TDMA frame among several clusters. By some experiments and tests it has been demonstrated the validity of the design and the implementation and there have been drawn some conclusions regarding the performance, the limitations and the future job on this field.

# Bibliography

- [1] A Kouba, M Alves, and E Tovar. Ieee 802.15.4 for wireless sensor networks: A technical overview. 2005.
- [2] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung. Mac essentials for wireless sensor networks. volume 12, pages 222–248, 2010. doi: 10.1109/SURV.2010.020510.00058.
- [3] Athanassios Boulis. Castalia: revealing pitfalls in designing distributed algorithms in wsn. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 407–408, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-763-6. doi: <http://doi.acm.org/10.1145/1322263.1322318>.
- [4] Amotz Bar-Noy, Vladimir Dreizin, and Boaz Patt-Shamir. Efficient algorithms for periodic scheduling. *Computer Networks*, 45(2):155 – 173, 2004. ISSN 1389-1286. doi: 10.1016/j.comnet.2003.12.017. URL <http://www.sciencedirect.com/science/article/pii/S1389128604000052>.
- [5] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, and D. Varvel. The pinwheel: a real-time scheduling problem. In *System Sciences, 1989. Vol.II: Software Track, Proceedings of the Twenty-Second Annual Hawaii International Conference on*, volume 2, pages 693 –702 vol.2, jan 1989. doi: 10.1109/HICSS.1989.48075. URL <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=48075>.
- [6] Zvika Brakerski, Aviv Nisgav, and Boaz Patt-Shamir. General perfectly periodic scheduling. In *Proceedings of the twenty-first annual symposium on Principles of distributed computing*, PODC '02, pages 163–172, New York, NY, USA, 2002. ACM. ISBN 1-58113-485-1. doi: 10.1145/571825.571854. URL <http://doi.acm.org/10.1145/571825.571854>.
- [7] T.W. Carley, M.A. Ba, R. Barua, and D.B. Stewart. Contention-free periodic message scheduler medium access control in wireless sensor/actuator networks. In *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*, pages 298 – 307,

- dec. 2003. doi: 10.1109/REAL.2003.1253276. URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=1253276>.
- [8] Li Li and Jing Zhao. Reducing inter-cluster tdma interference by slot-pre-allocation-protocol in sensor networks. In *Industrial and Information Systems (IIS), 2010 2nd International Conference on*, volume 2, pages 230 –233, july 2010. doi: 10.1109/INDUSIS.2010.5565634. URL <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5565634>.
- [9] T. Wu and S. Biswas. A self-reorganizing slot allocation protocol for multi-cluster sensor networks. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 309 – 316, april 2005. doi: 10.1109/IPSN.2005.1440940. URL <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1440940>.
- [10] A. Koubaa, M. Alves, and E. Tovar. Gts allocation analysis in ieee 802.15.4 for real-time wireless sensor networks. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, page 8 pp., april 2006. doi: 10.1109/IPDPS.2006.1639415. URL <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1639415>.
- [11] J. Stewart, R. Stewart, and M.N. Hassan. The application of decay rate analysis for wsn buffer dimensioning. In *Wireless Telecommunications Symposium (WTS), 2012*, pages 1 –6, april 2012. doi: 10.1109/WTS.2012.6266145. URL <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6266145>.
- [12] N. Sazak, I. Erturk, E. Koklukaya, and M. Cakiroglu. An event driven slot allocation approach to tdma based wsn mac design and its effect on latency. In *Computer Engineering Conference (ICENCO), 2010 International*, pages 22 –25, dec. 2010. doi: 10.1109/ICENCO.2010.5720419. URL <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5720419>.
- [13] S. Hussain, A.S. Zahmati, and X. Fernando. Lasa: Low-energy adaptive slot allocation scheduling algorithm for wireless sensor networks. In *Sarnoff Symposium, 2009. SARNOFF '09. IEEE*, pages 1 –6, 30 2009-april 1 2009. doi: 10.1109/SARNOF.2009.4850360. URL <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4850360>.
- [14] S. Narayanan, Jung Hyun Jun, V. Pandit, and D.P. Agrawal. Proportionally fair rate allocation in regular wireless sensor networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pages 549 –554, april 2011. doi: 10.1109/INFCOMW.2011.5928874. URL <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5928874>.