Modeling, simulation and implementation of an 802.15.4 based adaptive communication protocol in wireless sensor network: application to monitoring the elderly at home

Juan Lu

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)

Présentée et soutenue par :
Juan LU

le mardi 26 février 2013

Titre :

Modeling, simulation and implementation of an 802.15.4 based adaptive communication protocol in wireless sensor network: application to monitoring the elderly at home

---

École doctorale et discipline ou spécialité :
ED GEET : Génie Electrique

Unité de recherche :
LAAS-CNRS

Directeur(s) de Thèse :

Eric Campo, Professeur Université Toulouse 2, LAAS-CNRS
Adrien Van Den Bossche, Maître de Conférences Université Toulouse 2, IRIT-CNRS

Jury :
Rapporteur: Francis Lepage, Professeur Université de Lorraine, CRAN
Rapporteur: Ye-Qiong Song, Professeur Université de Lorraine, LORIA
Examinateur: Danielle Fournier-Prunaret, Professeur INSA-Toulouse, LAAS
Examinateur: Sylvain Durand, Maître de Conférences Université Motpellier 3, LIRMM
Invité: Thierry Val, Professeur Université Toulouse 2, IRIT
Invité: Thierry Gayraud, Professeur Université Toulouse 3, LAAS

Résumé

Le maintien à domicile des personnes fragiles vivant seules est devenu une préoccupation majeure de santé publique dans nos sociétés modernes. Parmi les différents aspects scientifiques traités dans le domaine de la surveillance à domicile, nous nous intéressons à l'étude et à la proposition d'une solution permettant à des capteurs répartis de communiquer entre eux de façon optimale et adaptée aux contraintes spécifiques de l'application. Plus précisément, nous souhaitons construire un réseau sans fil courte portée constitué de plusieurs nœuds capteurs échangeant entre eux des données selon un protocole de communication de niveau MAC (contrôle d'accès au médium) qui optimise à la fois l'énergie, le délai de transmission et la perte d'informations. Pour cela, nous avons finement analysé les avantages et les limites des technologies WPAN (réseau local personnel sans fil) et des protocoles de communication actuellement utilisés en rapport aux exigences de notre application. Nous avons ensuite proposé une méthode d'accès au médium déterministe, adaptative et économe en énergie basée sur la couche physique IEEE 802.15.4 et une topologie maillée. Elle permet de garantir le délai d'acheminement des messages avec un risque de collisions très fortement limité, grâce à une réutilisation spatiale du médium dans un voisinage à deux sauts. Cette proposition a été caractérisée par modélisation et simulation à l'aide du simulateur de réseau OPNET. Nous avons alors implémenté les mécanismes proposés sur des dispositifs matériels et déployé un réseau de capteurs en situation réelle afin de vérifier la pertinence du modèle et évaluer la proposition selon différentes configurations de test.

Abstract

Monitoring behavior of the elderly and the disabled living alone has become a major public health problem in our modern societies. Among the various scientific aspects involved in the home monitoring field, we are interested in the study and the proposal of a solution allowing distributed sensor nodes to communicate with each other in an optimal way adapted to the specific application constraints. More precisely, we want to build a wireless network which consists of several short range sensor nodes exchanging data between them according to a communication protocol at MAC (medium access control) level that optimizes energy consumption, transmission time and loss of information. To achieve this objective, we have analyzed the advantages and the limitations of WPAN (wireless personal area network) technology and communication protocols currently used in relation to the requirements of our application. We then proposed a deterministic, adaptive and energy saving medium access method based on the IEEE 802.15.4 physical layer and a mesh topology. It ensures the message delivery time with strongly limited collision risk due to the spatial reuse of medium in the two-hop neighborhood. This proposal was characterized by modeling and simulation using the OPNET network simulator. We then implemented the proposed mechanisms on hardware devices and deployed a sensors network in real situation to verify the accuracy of the model and evaluate the proposal according to different test configurations.

Acknowledgements

Completing the PhD study and writing this dissertation was an amazing journey that would not have been possible without the support and encouragement of many people.

My greatest appreciation and gratitude goes to my advisor Professor Eric Campo for his trust and support. I would like to thank him for advising me during all phases of my PhD, but also for giving me the freedom to find my own way. The environment he created is open, application-based and very efficient. It was a pleasure and a great honor to work with him.

I am especially grateful to my advisor Doctor Adrien Van Den Bossche, who not only guided me during the past three and a half years, but also shared valuable time, experience and vast knowledge in many discussions and a large number of joint research work. I also have to thank him for helpful career suggestions and enthusiastic help through this important period of my life.

I would like to thank the members of my PhD committee, Professors Francis Lepage, Ye-Qiong Song, Danielle Fournier-Prunaret and Doctor Sylvain Durand for their review work of this thesis and helpful suggestions in general. I also thank the two invited professors Thierry Val and Thierry Gayraud for their presences and interesting opinions and questions.

My gratitude extends to Thierry Val, Anne Wei and many professors of IUT-Blagnac, Who provided their supports and many insightful suggestions through my PhD research. I am also very grateful to many professors and staffs in the laboratory LATTIS, IRIT and LAAS for their help in a number of ways.

I will forever be thankful to my colleagues Rejane Dalce, Linqing Gui, Youssouf Zatout, Chiraz Houaidia, Asma Bel Hadj Med etc. I have had the pleasure of interacting with these great colleagues and some of which became precious friends. Moreover, I would like to thank students Vincent Bragard, Remy Phelipot, Damien Clerc and Nathan Dubot of IUT-Blagnac for their efficient work for a part of my PhD research.

Finally, I would like to express my deepest gratitude to my family and my friends. They all stood by me and provided unconditional love and care. They shared both the great and the difficult moments of life with me. Thank you so much!

Table of Contents

Introduction

Nowadays the aging population is constantly increasing so that monitoring behavior of the elderly and the disabled living alone has become a major public health problem in our modern societies (T. Fent *et al*., 2006; J.R. Boulanger and C. Deroussent, 2008). These individuals attach a great importance to the autonomy that allows them to live mostly at home, and in their immediate environment, providing them freedom and a better quality of life. But, in the case of an accident such as a fall, faintness…, that autonomy can quickly turn into dependence. To supply solutions, some poeple wear systems embedded on their body, such as physiological sensors or fall sensors (Jianchu Yao *et al*., 2005; Kwang Yong Lim *et al*., 2008; H. Mamaghanian *et al*., 2011). These devices are intrusive and limitations become apparent due to the fact that the patient is often unable to use an alert system because either he is not wearing his equipment or, if he suddenly feels unwell, is unable to perform the alert activation gesture.

The solution we consider is to instrument the environment of the person. Indeed, by monitoring the main environmental characteristics of their living space, it seems to be possible to get a lifestyle pattern of the person (V. Rialle *et al*, 2004; Y. Zatout and E. Campo, 2009; A. Anfosso and S. Rebaudo, 2011). For example, measuring temperature, humidity, luminosity, noise levels, presence..., in many strategic areas at home can provide useful data to interpret a physical activity in space and time. Data processing will determine circadian activity rhythms of the person and so will contribute to detect unusual situations and emergency cases. Generally, the challenge is to propose a suitable sensor network that allows uninterrupted data transmission in a bounded time.

Within this context, the objective of this work is to modelize and implement a complete heterogeneous sensor network allowing the measurement and the transmission of short-range data collected by the environmental sensors. The planned network will be deployed in a house or even building and transmit alert messages caused by a malfunction of environmental parameters via a continuous monitoring. So a limited scale, up to 50 nodes, seems to be sufficient for this home monitoring application. These nodes exchange data between them according to a communication protocol that optimizes energy consumption, transmission delay and loss of information. Another principle to consider is that when any node fails, the

network should repair automatically and must run normally with a minimal loss of information.

Transmission considered will use low power wireless technology combined if necessary with a power line communication. So we begin this work from the investigation of wireless and wired technologies used in the home monitoring field. We find that some wired technologies support both low rate and high rate communications. However, our work concentrates on low rate networks which mainly target to the sensor data transmission, even in unusual or emergency cases. On the other hand, a wireless sensor network allowing efficient monitoring for a few weeks or months could constitute a very interesting scenario, instead of penetrating walls to install a wired network. Therefore, we take advantage of the wireless technologies in terms of convenient installation, flexible deployment and comfortable environment for the monitored people. One of challenges in this case is the energy-constrained sensor devices.

In fact, from the network point of view, key emphasis of this work is on WPAN which tries to provide low power, low cost and short-range solutions. Among them, IEEE 802.15.4 is considered as a promising way in terms of energy saving and guaranteed medium access. Many other main technologies such as ZigBee, IEEE 802.15.5 and 6LoWPAN are based on IEEE 802.15.4 MAC or backwards compatible with this standard. Therefore, we consider IEEE 802.15.4 as a starting point for our work. In fact, we use IEEE 802.15.4 physical layer as it is, without any change. On the other side, we optimized the IEEE 802.15.4 medium access control layer to better suite our specific constraints. The MAC layer has a fundamental and significant impact in a protocol stack. The upper layers including network layer, transport layer, application layer, etc. will be considered after a robust MAC layer.

Hence our works are focused on MAC layer of the OSI model. In this way, we improve IEEE 802.15.4 standard in order to satisfy our particular application. The new communication protocol should have the ability of giving different priorities to various data flows based on their requirements by controlling the medium sharing. It actually means the need of different medium access methods. As well known, CSMA/CA is a contention-based access method which provides a best-effort service. However, our application requires communications with low latency and without packet loss, especially for the alert messages which may directly affect the safety and health of the monitored people. The guaranteed medium access method is therefore urgently expected.

Meanwhile, we decide to take advantage of the mesh architecture to build and maintain the wireless network. Mesh architecture generally enables automatic organization, no central management using a “super node” and fast route recovery since communications are possible with all neighbor nodes. Unlike ZigBee, we desire that all the sensor devices, including routers, can sleep in the mesh network for energy saving. Intelligent time schedule mechanism is also expected to extend the network lifetime as much as possible. Another benefit of mesh architecture lies in its robustness. Any sensor devices including routers can fail, but the rest of the network should work properly thanks to mesh link redundancy. At last, unlike star or tree topology in which typically a supernode is previously fixed and schedule the shared resources, mesh topology can better adapt to topological changes and strengthen flexibility and security of the monitoring.

In general, we work at MAC layer of the wireless mesh sensor network; this MAC protocol enables different QoS levels with rational energy consumption. Modeling and simulation are important working methods helping to verify our communication protocol, to evaluate its performances, and to improve the propositions. Prototype implementation is also achieved with available sensor application boards in real situations. This work can verify the feasibility and accuracy of the simulation model and serves to optimize the protocol model in return.

Hence the manuscript is structured in the following way: Firstly, our application context and challenges are detailed in the Chapter 1 explaining the motivation and the objectives of this work. The main wired and wireless technologies on habitat monitoring, including IEEE 802.15.4, are studied. Their limitations for our application requirements lead to the need of a new adaptive protocol, which enables determinism medium access and energy saving for all nodes, including routers.

In Chapter 2, a novel MAC protocol is proposed in order to improve robustness and flexibility of multi-hop sensor network. This chapter includes the description of the network formation, node architecture, protocol function and its operation details.

In Chapter 3, we simulate the proposed protocol with OPNET network simulator to evaluate the scope of our contribution. The simulation results show performances in the respects of protocol cost, QoS capability and energy consumption, etc.

Finally, Chapter 4 presents the prototype in order to verify our proposal and improve the protocol by solving the challenges not considered in simulation. We implement the proposed protocol on the integrated sensor boards and deploy the network consisting of several sensor devices in a real environment in a smart home.

A final conclusion and some perspectives are given in the last part of this manuscript.

Chapter 1

Wireless Sensor Networks and Technologies for Home Health Monitoring

In the first chapter, the motivation of this thesis is addressed. Firstly, our application context and challenges are introduced. Several projects on habitat monitoring using a wireless sensor network are quickly presented and their common characteristics are identified and discussed. This analysis will give us the guidelines of our work. Secondly, we present an overview on wireless or wired home networking standards. The emphasis is on comparison of advantages and shortcomings of these technologies for our application. More specifically, IEEE 802.15.4 standard is studied and the problematic is presented. Some related works are analyzed in order to explain the relevance of our approach presented in Chapter 2.

1. Application context and challenges

The aging population is constantly increasing and most European countries are now facing an urgent requirement to provide appropriate home environment solutions for their citizens. In this part, several projects on habitat monitoring are investigated and their system architectures are summarized. Our work focuses on one sub-layer of this multi-tier system. The application requirements and crucial constraints are discussed at last, for example quality of service and energy.

1.1. Common architecture for habitat monitoring networks

Elderly requiring healthcare services must move to distant medical centers and this is often not feasible due to their health state. A home designed with advanced communication networks and Internet technologies could allow the elderly and the disabled to live alone with high levels of comfort and safety. Within a smart home environment, the tiny embedded devices with sensing, computation and communication capabilities can help to keep recordings of patients, to share data between hospitals, to monitor activities of persons and detect problems (fall, faintness…), but also to keep an eye on their living environment.

Nowadays, many healthcare integrated in smart home applications are available in France and over the world [1.1]. However, it is difficult to find the system that matches exactly all the end-user requirements because of specific home constraints and functionalities. Moreover, some systems are costly and complicated, and beyond what most people need, or even want in their homes. There is still space to improve in the technology and innovation of smart home. We present below some main projects in France using wireless technologies.

LORIA laboratory of Nancy presented a universal user-oriented healthcare system to allow elderly to be medically monitored and assisted in their home [1.2]. Thanks to ZigBee technology [1.33], a wireless monitoring network is organized by sensors installed at home. The gateway in each home attempts to integrate this wireless network to other wired networks, Internet for example. Finally, the users can access the system from anywhere thanks to web services technology.

TOPCARE [1.3] proposed technical devices and telecommunication structures for the elderly and patients at home. The system includes a Telematic Homecare Platform (THP°)

backbone, the development of Telematic Home Stations (THS°) and Health Professional Stations (HPS°). A communication server manages the network administration, the THS registration, the device communication and the Internet access.

TIMC-IMAG laboratory of Grenoble developed a project called System of Information and Communication of the Intelligent Home for Health (SIC-HITCH) [1.4]. Objective is to monitor individuals with ZigBee sensors installed in their home, by triggering off alarms in appropriate emergency centers. Then doctor at hospital could acquire real-time news of the monitored people by using a remote controller. This system is an experimentation and simulation platform.

The objective of GERHOME project in Nice is to develop, try out and certify technical solutions supporting the assistance services for enhancing independence of the elderly at home, by using RF technology standard for house automation to ensure autonomy, comfort of life, security, monitoring and assistance to place of residence [1.5].

The Homecare project of LAAS in Toulouse aims to support autonomous living, and to sound alarms in emergencies inside a long medical care unit. A wireless presence sensors network installed in the room of patients combined with a ZigBee radio communicating patch worn by the patients to identify them allow the assessment of mobility and activity [1.6].

In [1.7] the authors proposed a design and energy-efficient multi-tier network solution for monitoring people at home (WSN-HM). This system allows integrating heterogeneous sensors with both medical and environmental/visual sensing capabilities, to realize a variety of functionalities at home. At last, sink node connects this sensor network with Internet backbone.

Clearly, home health monitoring application is a complex system that requires the integration of various sub-systems and usually has a multi-tier architecture as shown in Figure 1.1. We summarize and discuss the elements of this architecture:

- Environmental sensors in Personal Area Network (PAN°): this network must involve sensors distributed in environment (room, hall, kitchen, toilet…). These sensors are various: temperature, humidity, movement, acoustics, magnetic, video, etc. Actuators may also be integrated in this network to act on the opening the window/door in case of fire for example.

- Medical sensors in Body Area Network (BAN°): this network consists of very small portable devices equipped with a variety of sensors for medical monitoring, patient localization and identification. This network may be combined with PAN directly or connect to the Internet by its own gateway. BAN could use Wi-Fi [1.8], Bluetooth [1.9] or other technologies to access the backbone when the monitored persons are outside.

- Gateway or sink: it may be a mini PC installed in each home or a mobile system such as PDA. Gateway connects BAN and PAN to the Internet. The data collected from medical and home automation sensors could be preprocessed in this part.

- Internet backbone: this system uses web services to interact between the client and the server, such as a monitored person at home and the hospital center. It aims to address the need to standardize the transmission, processing and storage of data for the monitoring service.

- Graphical user-interfaces: a very easy-to-use graphical user-interface is quite important. 65% of the elderly do not accept to wear sensors and most of them want simple and efficient products [1.10]. This interface can be ordered by using a PDA or a TV remote controller.

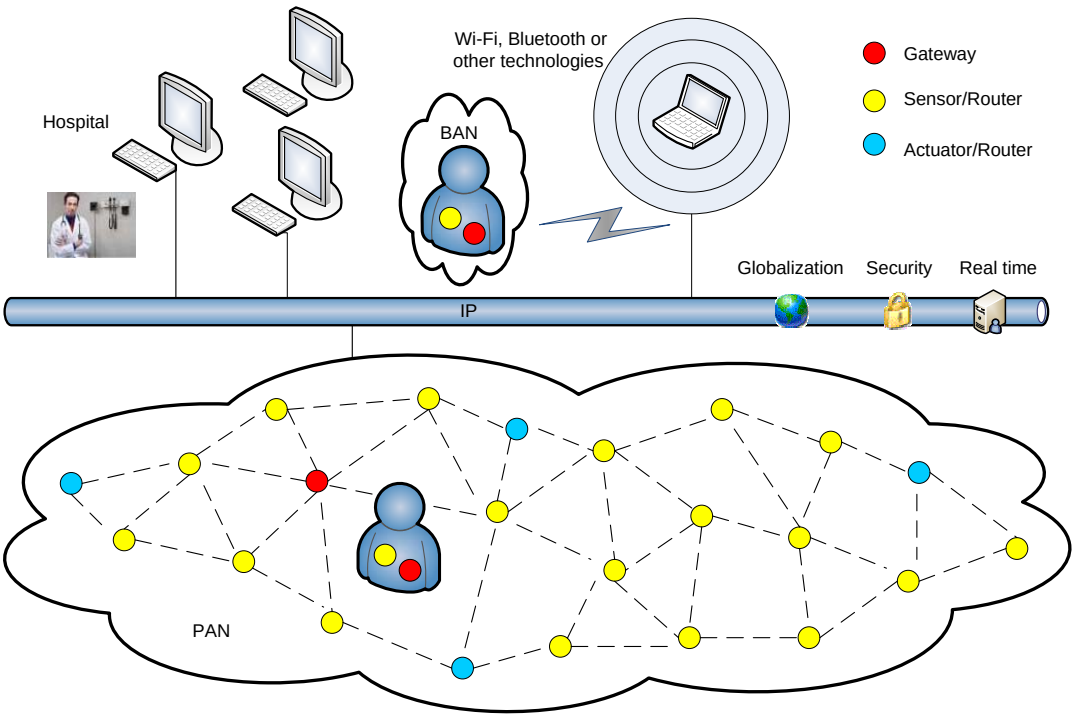

Figure 1.1 General architecture of home health monitoring application

1.2.Our application focus

This thesis is focused on PAN and BAN for remote measurement of environmental and health parameters. The communication network is a key element of the system because it allows collecting data and transmitting them to the recipient. In the following of this work, PAN and BAN will be considered as Wireless Sensor Networks (WSN°).

1.2.1.Application context

Our objective is to enable the multi-sensors network composed by home automation equipments and health sensors worn by the user to monitor its life activities and to provide him safety, comfort and assistance at home.

The planned network deployment is short-range and temporal-bounded. It enables to respond to a continuous monitoring of environmental and physiological parameters via the transmission of regular sensor messages or alerts due to a potential risk on the person (fall, faintness, getting lost, etc.). The referred network scale is limited, typically with several tens of nodes in a zone of 100 m * 100 m at most.

1.2.2.Wireless sensor networks

WSN have drawn a great attention in home monitoring field [1.11] by the ability to collect information from the physical environment, to perform simple processing on the extracted data and to transmit it to remote locations. In our application, the WSN has the following characteristics:

- Indoor environment: the devices/nodes are put on the ceiling, wall or furniture at home. Therefore, the wireless transceivers are restricted to short range communication with low radiation power (e.g. about 0 dBm) [1.12].

- Low bandwidth: there are generally two types of application traffics in our application. Environmental parameters such as temperature are reported periodically (e.g. once/hour). On the other hand, burst traffics such as temperature alarm in fire or fall alarm of people faintness should be delivered with guaranty. Generally the throughput is low (e.g. from several bits to hundred kbits per second).

- Variable number of multi-sensors: as mentioned before, the WSN includes environmental sensors such as temperature, humidity, luminosity, etc. and health sensors such as accelerometers or physiological sensors. Our WSN should support approximate 50 sensor nodes.

Two other important characteristics of our WSN are network topology and energy consumption which are discussed in 1.2.1.1 and 1.2.1.2.

1.2.2.1. Network topology

Logical topology, referring to how data is actually transferred in the network as opposed to physical topology [1.13] [1.14], is used in this thesis. Figure 1.2 shows examples of three typical network topologies in a WSN.

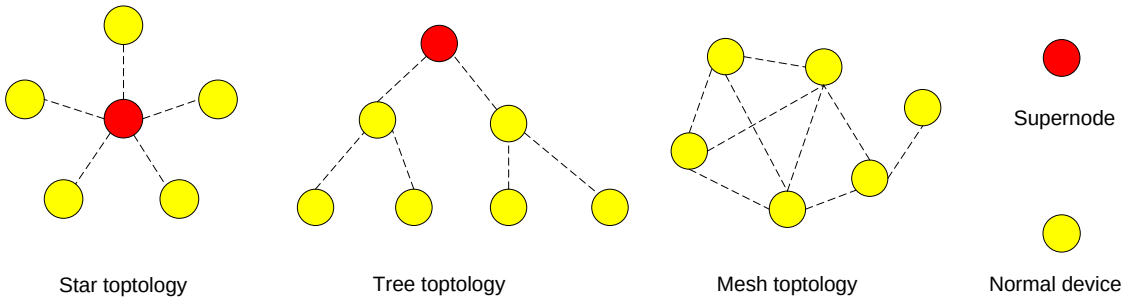

Figure 1.2 Network topology examples

- Star topology: each device is connected to a central node in the star topology. It is considered the easiest topology to design and implement. The primary disadvantage of this topology is that the supernode represents a single point of failure. The network coverage is limited to the communication range of supernode and so the number of nodes in the network is restricted.

- Tree topology: this is the hierarchy topology with a central node at the top level. Each node has a specific fixed number which indicates the deep of the node and usually is allocated by supernode. The flexibility and mobility of network are therefore limited by this topology. Furthermore, the communications are constrained between parent nodes and children nodes. Therefore a single point of failure also occurs in the tree topology network.

- Mesh topology: this is the distributed topology and each node has the same capability. The nodes of the network are connected to more than one other node

with a point-to-point link. This makes it possible to take advantage of some redundancy provided by the physical radio. However, mesh topology is more difficult to implement than the two above topologies.

In our application, most of sensor nodes will be fixed to the wall and furniture and only few nodes are with low mobility. For example, the elderly/disabled wearing the accelerometer sensor should be free and easy at home.

In a wireless network, topological changes may be caused by the failure or exhaustion of any node, the movement of some mobile nodes, or the unreliability of wireless medium. However, our application requires that some vital messages, such as fall alarm, must be transmitted in time without loss. When one node fails, no matter router or not, rest of the network should find a new way to send the vital messages. In another word, the routers could fail in our application. Unfortunately, in star or tree topology, the supernode which represents a single point of failure may bring risks for the whole network. It's the reason why we need a mesh topology which can better adapt to topological changes and strengthen security and robustness of the monitoring.

1.2.2.2. Energy consumption

Energy consumption is a fundamental concern in WSN. The sensor node, being a micro-electronic device, can only be equipped with a limited power source. However, in our application scenario, replenishment of power resources might be difficult. For example, it is not possible to ask the elderly/disabled to change sensors' battery. Sometimes, about 50 sensor nodes are installed at home and some of them may be depleted rapidly.

Furthermore, our application requires a multi-hop mesh network as explained before. So each node could play the dual role of data generator and data router. The malfunction of few nodes can cause significant topological changes and might require re-routing of packets and re-organization of the network. Hence, we expect not only to maximize the lifetime of battery-constrained sensor nodes but also to extend the lifetime of the whole network. All the nodes, including routers, could sleep for energy saving.

1.2.3.Challenges in our application

Our works are focused on Medium Access Control (MAC°) layer. According to the application requirements expressed in the previous part and the problematic study of WSN [1.11] [1.12], the performance criteria to take into account in MAC layer are:

- Quality of service (QoS°): simply or practically, QoS brings the ability of giving different priorities to various users, applications, and data flows, frames or packets based on their requirements by controlling the resource sharing [1.15] [1.16] [1.17]. In our application, the crucial messages like security alarms should be delivered promptly without packet loss. Therefore, how to provide a guaranteed mechanism which makes sure all the data could arrive at destination in time? Two factors of QoS, dropped packets tolerance and latency tolerance, so are mainly concerned in this study.

- Energy saving: a long lifetime network, several months or even years, is eagerly expected. As the largest energy consumption of the sensor nodes is due to the time spent in idle listening [1.18], especially for low rate traffic, so the sleep-awake schedule of radios is strongly recommended to economize energy. At the same time, how to provide appropriate mechanisms to maximize the network lifetime is one emphasis.

- Flexibility and robustness: the failure of critical nodes can lead to the entire network failure and harm the safety of people monitored by this WSN. Therefore, the network should be self-organizing, quickly deployed and re-deployed. However, how to construct a mesh WSN with the robustness against link failure or link establishment?

- Scalability: this WSN is expected to handle growing amount of sensor nodes flexibly. For example, the join of new sensor nodes at home, the transition from home monitoring to building monitoring, the integration of several WSNs. Thence scalability is also a great challenge when designing the network and its mechanisms.

In general, these challenges have to be guardedly considered when analyzing specific application requirements. In our application, the first three criteria will be studied and verified by a simulation work and prototyping. The scalability criterion is not involved here.

2. The existing technology/standard

Several wired and wireless standards for home WSN are studied in this part. The focus is on analyzing their advantages and shortcomings with respect to our application. At last, a comparison table summarizes them and gives the selected options.

2.1. Wired technology/standard

We begin this section with XIO and Ethernet protocol. Afterwards, KNX and HART which have the more complete protocol model, from physical layer until to application layer, are studied.

2.1.1. XIO

XIO is a packet-based power line protocol. It forms a bus between high-performance system devices and the controller. Hence, a star topology, using a router to connect up to 8 fully symmetrical devices, is usually employed in the network [1.19].

XIO has two source-synchronous channels [1.20], and one in each direction. The channels are clocked at 400 MHz to achieve peak rates of 800 MB/s. Each device can utilize the full bandwidth, as the router/controller prevents collisions by being able to route between any two points.

Obviously, XIO is used in high performance data transfer applications which constrain limited devices and require unlimited power. In a smart home, XIO may connect gateway nodes or wired devices to the Internet backbone.

2.1.2. Ethernet

Ethernet is a family of computer networking technologies for Local Area Networks (LAN). It defines wired physical layer and MAC layer [1.21].

The original Ethernet used coaxial cable as a shared medium. Later the coaxial cables were replaced by twisted pair and fiber optic links in conjunction with hubs or switches. Data rates were periodically increased from the original 10 MB/s to 100 GB/s.

The MAC is the portion of Ethernet core that handles the Carrier Sensor Multiple Access with Collision Detection (CSMA/CD °) mechanism. Each frame contains source and destination addresses and error-checking data so that damaged data can be detected and re-transmitted by the MAC layer.

In home monitoring application, Ethernet is usually used for the high rate backbone network with a star or a tree topology. Unlike XIO, users can access their automated home from anywhere in the world thanks to the power of internet. In addition, Ethernet is direct compatible with wireless Wi-Fi technology. Unfortunately, there are few available Ethernet products in the market because of the lack of standardization in the upper layers.

2.1.3.KNX

KNX is a standardized network communications protocol for intelligent home and buildings [1.22]. This protocol specifies 5 layers: physical layer, data link layer, network layer, transport layer and application layer [1.23].

Firstly, KNX defines several physical communication mediums:

- Twisted pair: this communication medium has the bit-rate of 9600 bits/s. The devices will operate and communicate with each other across the separate bus cables, hierarchically structure in lines and areas.
- Power line: with this communication medium, KNX devices will operate and communicate on the same electrical distribution network. Bit-rate is 1200 bits/s.
- Radio frequency: KNX devices supporting this communication medium use radio signals to transmit KNX telegrams. Telegrams are transmitted in the 868 MHz frequency band, with a maximum radiated power of 25 mW and bit-rate of 16.384 kb/s. It allows unidirectional and bidirectional implementations for small and medium size installations which only requires re-transmitters in exceptional cases.

- IP/Ethernet: KNX telegrams can also be transmitted and encapsulated in IP telegrams. In this way, LAN networks as well as Internet can be used to route KNX telegrams.

The MAC access method for KNX is Carrier Sensor Multiple Access with Collision Avoidance (CSMA/CA°). Collisions are avoided by writing and listening to the bus at the same time.

In KNX, the routing table is being defined at the installation step of the network. Its entries are static and are not self-modified during runtime. Unicast, multicast and broadcast are supported in the network layer.

The KNX transport layer offers two methods of transferring data to the application layer: connection oriented communication and connectionless communication.

At last, the application layer implements services such as process data communication, device management and network management.

In conclusion, KNX is a worldwide standard for home monitoring applications, ranging from lighting and shutter control to various security systems, heating, metering as well as household appliances. Over 200 member companies have almost 7000 KNX certified product groups. However, the protocol specifications are not free for non members. In another hand, KNX standard focuses on low-power low-rate monitoring. QoS requirement is not considered in the standard. Also, network flexibility is restricted by the installation of wired devices and KNX static communication mechanisms.

2.1.4.HART

HART protocol is the global standard for sending and receiving digital information across analog wires between smart devices and control or monitoring system. This protocol implements physical layer, data link layer, network layer, transport layer and application layer [1.24].

The physical layer uses Frequency Shift Keying (FSK°) to communicate at 1200 bps. The signal frequencies representing bit values of 0 and 1 are 2200 and 1200 Hz respectively. This signal is superimposed at a low level on the 4 to 20 mA analog measurement signal without causing any interference with the analog signal.

The data link layer defines a master-slave protocol. In normal case, a field device only replies when it is spoken to. There can be two masters, for example, a control system as a primary master and a handheld HART communicator as a secondary master. Time Division Multiple Access (TDMA°) is employed and the timing rules are defined when each master initiates a communication transaction. Up to 15 slave devices can be connected to a single multidrop cable pair.

The network layer manages sessions for end-to-end communication by static routing. Then the transport layer can be used to ensure that end-to-end communication is successful.

The application layer defines the commands, responses, data types and status reporting. The public commands of the protocol are divided into four major groups: universal commands, common practice commands, device specific commands and device family commands.

HART technology is widely used in industry automation field and has 990 types of devices from 238 companies, such as actuator, isolators, loop monitor and pressure transmitters. Home sensor devices and household appliances are not available yet. However, the wireless versions of HART fieldbus protocol, which may be more appropriate for home monitoring, will be studied in the following section.

In part 2.1, four wired technologies were researched. Most of them are designed for high rate and unlimited power applications. However, our work concentrates on low rate and low cost sub-system of home monitoring. Generally speaking, the global cost including the installation and the deployment of wired devices at home is more important and complex than that of wireless devices. In same cases, wired devices are not even accepted because it's not always possible to install cables in the habitat or the elderly/disabled can not be free with the attached cables.

2.2. Wireless technology/standard

We begin this section with some examples of standards which define low layer protocol model (2.2.1 and 2.2.2). Afterwards several standards with network layer are discussed (2.2.3, 2.2.4 and 2.2.5). At last, Z-Wave and WirelessHART, implementing physical layer until application layer, are presented (2.2.6 and 2.2.7).

2.2.1.IEEE 802.15.4

2.2.1.1. General description

As shown in Figure 1.3, IEEE 802.15.4 specifies the MAC layer and Physical layer (PHY °) for low-cost low-power wireless network [1.25].

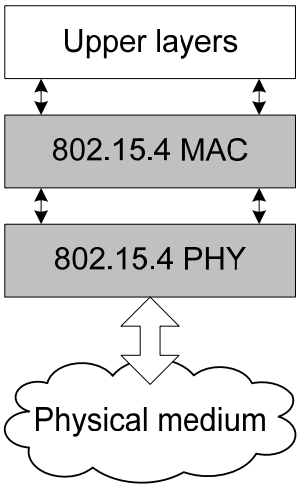

Figure 1.3 Node architecture of IEEE 802.15.4

- IEEE 802.15.4 operates in the 868 MHz, 915 MHz and 2.4 GHz ISM bands. IEEE 802.15.4a [1.26] adds 3 optional UWB PHY in the 500 MHz and 3.1 GHz to 10.6 GHz bands. UWB waveforms support precision ranging between devices, so the device can provide enhanced resistance to multipath fading for robust performance with very low transmit power.

- There are beacon-enabled mode and nonbeacon-enabled mode at MAC layer. The beacons are used to synchronize the attached devices, to identify the PAN, and to describe the structure of the superframes.

- Two different device types can participate in an IEEE 802.15.4 network: a Full-Function Device (FFD°) and a Reduced-Function Device (RFD°). The FFD can operate in 3 modes serving as a PAN coordinator, a coordinator, or a device.

2.2.1.2. Advantages and features

Nonbeacon-enabled mode has the advantage of lower complexity. However, beacon-enabled mode is particularly interesting for our application as the following characteristics:

- It includes a mechanism called Guaranteed Time Slot (GTS°). The GTSs form the Contention-Free Period (CFP°) which is dedicated to low-latency application or application requiring specific data bandwidth. CSMA/CA is used in Contention Access Period (CAP°). 802.15.4a also adds Aloha mechanism for the UWB device.

- It is possible to achieve variable sleep-awake duty cycles in beacon-enabled mode. The inactive period of superframe allows the nodes going to the sleep mode for energy saving.

In addition, there are a lot of IEEE 802.15.4 products with small size and low cost, such as IRIS, MicaZ, TelosB, and Imote2 [1.27] [1.28] [1.29], etc. This makes our work possible to realize and test in real conditions.

2.2.1.3. Shortcomings for our application

IEEE 802.15.4 may operate in two topologies: a star topology or a peer-to-peer topology. Peer-to-peer topology allows more complex network to be implemented such as cluster-tree topology. However, how to construct a mesh networking topology as required by our application is missing. In fact, while the current standard supports multi-hop networking using peer-to-peer topology, it restricts its use to non beacon-enabled mode. This contradiction makes the interesting advantages, such as GTS and energy saving thanks to sleep mode, disappear.

2.2.2.IEEE 802.15.6

2.2.2.1. General description

The IEEE 802.15 Task Group (TG°) 6 [1.30] is developing a communication standard optimized for low-power in-body/on-body devices to serve a variety of medical and non-medical applications for BAN. As shown in Figure 1.4, the standard defines a MAC layer supporting several PHY layers [1.31].

The current IEEE 802.15.6 standard defines three independent PHY layers [1.32]: Narrowband (NB°), Ultra-WideBand (UWB°) and Human Body Communications (HBC°) layers. The selection of each PHY depends on the application requirements.

- NB: this physical is responsible for activation/deactivation of the radio transceiver, Clear Channel Assessment (CCA °) within the current channel and data transmission/reception. Depending on different modulations used by this physical layer, data rates range from 57.5 Kbps to 485.6 Kbps.

- UWB: this physical layer operates in two frequency bands. Both low band and high band are characterized by a bandwidth of 499.3 MHz. The low band consists of 3 channels and the high band consists of 8 channels. Typical data rates range from 0.5 Mbps up to 10 Mbps with 0.4882 Mbps as the mandatory one.

- HBC: this physical layer operates in two frequency bands centered at 16 MHz and 27 MHz with the bandwidth of 4 MHz. This physical layer frame structure contains a preamble. The preamble sequence is transmitted 4 times in order to ensure packet synchronization. When the packet is received by the receiver, it finds the start of the packet by detecting this preamble sequence.

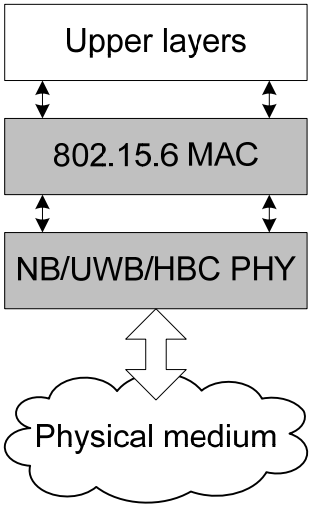

Figure 1.4 Node architecture of IEEE 802.15.6

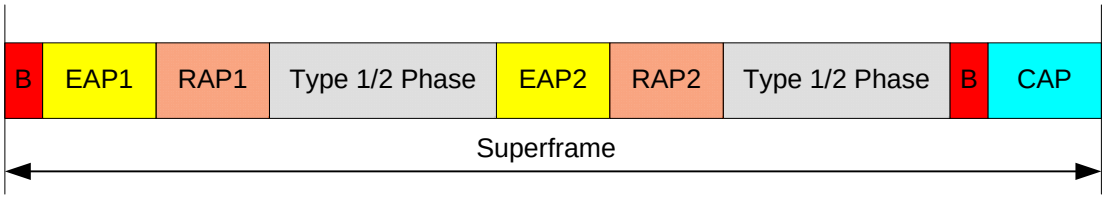

Figure 1.5 Superframe structure of IEEE 802.15.6

The standard defines a sophisticated MAC protocol on the top of PHY. The network can operate in three modes and several medium access methods and their combinations are

provided by these modes. Figure 1.5 shows the superframe structure which is bounded by a beacon period of equal length. A supernode selects the boundaries of the beacon period and thereby selects the allocation slots. Supernode may also shift the offsets of the beacon period.

- Beacon mode with beacon period superframe boundaries: in this mode, the beacons are transmitted by the supernode in each beacon period except in inactive superframes. The superframe is divided into Exclusive Access Phase 1 (EAP1˚), Random Access Phase 1 (RAP1˚), Type 1/2 phase, Exclusive Access Phase 2 (EAP2˚), Random Access Phase 2 (RAP2˚), Type 1/2 phase, and a CAP. In EAP, RAP and CAP periods, nodes contend for the resource allocation using either CSMA/CA or a slotted Aloha access procedure. The EAP1 and EAP2 are used for highest priority traffic such as reporting emergency events. The RAP1, RAP2 and CAP are used for regular traffic only. The Type 1/2 phase are used for uplink allocation intervals, downlink allocation intervals, bilink allocation intervals, and delay bilink allocation intervals. In Type 1/2 phase, polling is used for resource allocation. Depending on the application requirements, the coordinator can disable any of these periods by setting the duration length to zero.

- Non-beacon mode with superframe boundaries: in this mode, the entire superframe duration is covered either by a Type 1 or a Type 2 access phase but not by both phases.

- Non-beacon mode without superframe boundaries: in this mode, the coordinator provides unscheduled Type 2 polled allocation only.

2.2.2.2. Advantages and features

- IEEE 802.15.6 standard supports both low and high rate applications. The data rate can be at most 10 Mbps thanks to the UWB technology.

- A variety of mechanisms are provided by the MAC layer to improve QoS. Random access uses either CSMA/CA or a slotted Aloha procedure for resource allocation. Unscheduled polling/posting is used for connectionless contention-free access. Connection-oriented contention-free access schedules the allocation of slots in one or multiple superframes.

2.2.2.3. Shortcomings for our application

- As IEEE 802.15.6 is typically designed for BAN, a supernode (or coordinator) selects the allocation slots and organizes a star or tree topology. The network flexibility as well as scalability is limited.
- Implementation is an important challenge for this complex standard. Up to now, there are no IEEE 802.15.6 products on the market.

2.2.3.ZigBee

2.2.3.1. General description

IEEE 802.15.4 is commonly known as ZigBee [1.33] because the ZigBee alliance and the IEEE committee decided to join forces to propose a low data rate, low power consumption and low cost wireless networking protocol stack.

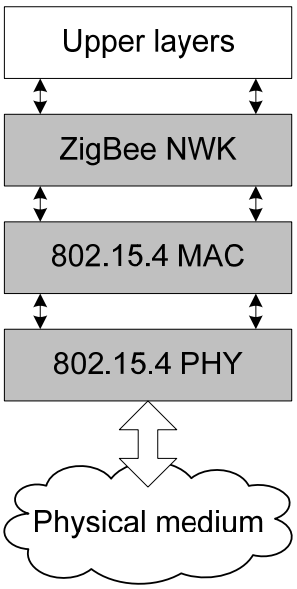

Figure 1.6 Node architecture of ZigBee

As shown in Figure 1.6, IEEE 802.15.4 focuses on the specification of the lower two layers. On the other hand, ZigBee aims to provide the upper layers of the protocol stack (e.g. routing protocol) for interoperable data networking, security services and marketing of the standard. This will assure consumers to buy products from different manufacturers with confidence that the products will work well together.

2.2.3.2. Advantages and features

- ZigBee routing layer start with two well-studied protocols: Ad-hoc On-demand Distance Vector (AODV °) and Motorola's cluster-tree algorithm [1.34]. Therefore, multi-hop communication is possible thanks to these protocols.
- Particularly, how to construct a cluster-tree topology is fully specified and that is exactly one lack of IEEE 802.15.4.
- We can find a large number of ZigBee products from different companies. For example, 13192 SARD (Freescale) [1.35] will be chosen for our prototype implementation.

2.2.3.3. Shortcomings for our application

- Cluster-tree routing algorithm is a hierarchical strategy. Single point of failure costs much time and energy to self-repair. The direct communication between neighbor nodes may not be possible. Therefore, the network performances are not as good enough as our application requirements.
- AODV-based routing algorithm works on non beacon-enabled mode. So there is neither sleep mode for energy saving nor GTS for different QoS capabilities.

2.2.4.IEEE 802.15.5

2.2.4.1. General description

IEEE 802.15.5 [1.36] specifies mesh topology capability in Wireless Personal Area Network (WPAN°). This standard defines recommended practices for low-rate WPAN mesh and high-rate WPAN mesh respectively. Only low-rate WPAN mesh is studied here as it is more suitable for our application.

As shown in Figure 1.7, IEEE 802.15.5 designs a mesh sublayer which allows devices to be compatible with IEEE 802.15.4 MAC/PHY. The objective is to provide an architectural framework that enables low-power, low-rate WPAN devices to promote interoperable, stable, and scalable wireless mesh topologies.

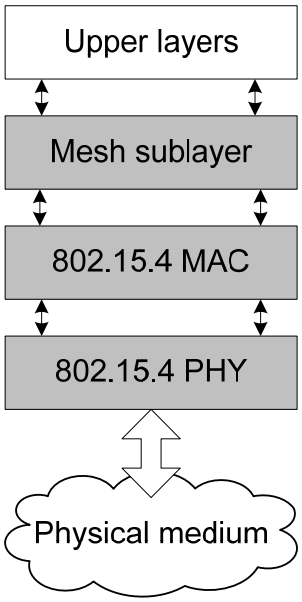

Figure 1.7 Node architecture of IEEE 802.15.5

2.2.4.2. Advantages and features

IEEE 802.15.5 includes the following strong points as the supported features:

- IEEE 802.15.5 provides Asynchronous Energy Saving (AES°) and Synchronous Energy Saving (SES°) algorithms to support mesh communication. Both methods are designed based on IEEE 802.15.4 nonbeacon-enabled mode. Instead of beacon frame at MAC layer, a hello command frame at mesh sublayer is broadcasted to synchronize the network. Hence, sleep mode and dedicated slot communication can be realized.

- A mesh WPAN could be built by this standard. So the network coverage is extended without increasing the transmission power or the receiver sensitivity. Obviously, mesh sublayer with route redundancy enhances flexibility and reliability of the network.

2.2.4.3. Shortcomings for our application

An adaptive robust tree and its meshed form are proposed by IEEE 802.15.5 in order to build the mesh network.

At first, three phases are defined for tree construction: initialization or configuration phase, normal phase and recovery phase. A tree is formed during initialization phase which

begins from the root, normally designated manually to be a topology server. Then node gradually joins the network with address assigned by topology server. Each branch is assigned a block of consecutive addresses according to their capability and other factors. During normal phase, new nodes are still allowed to join the network, but the number of new nodes should be small compared with the number of nodes already in the network. If the tree is broken, the recovery phase is triggered. Control commands such as RREQ (route request), RREP (route reply) and RRER (route error) are broadcasted for route repair.

A mesh topology is formed by keeping additional local links in the route table of each node. From each individual node's point of view, the network is still a tree. But the brother nodes connected through mesh link will treat each other as a child and add this link entry in each other's route table. So tree link provides a simple data forwarding, mesh link provides alternative paths and optimized data forwarding.

Therefore the following points of IEEE 802.15.5 remain to be seen:

- From our point of view, the special role of some set of nodes (e.g. topology server) should be more capable than normal nodes. A shorter path may be missing as tree link has priority when data forwarding. This mesh topology is actually a multi-path tree topology.
- The protocol cost, such as control overhead for the tree formation and the route repair, seems to be expensive and should be further studied.
- At last, there are seldom products of IEEE 802.15.5 on the market. The only implementation as far as our knowledge is from advanced wireless networking lab of the City University of New York [1.37] [1.38].

2.2.5.6LoWPAN

2.2.5.1. General description

IPv6 over Low power WPAN (6LoWPAN) is under development by IETF working group. The objective of this standard is to utilize IPv6 as addressing, routing and security mechanisms for low power WSN [1.39].

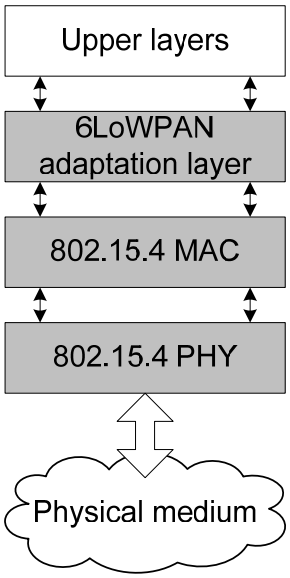

Figure 1.8 Node architecture of 6LoWAPN

As shown in Figure 1.8, 6LoWPAN is based on IEEE 802.15.4 PHY and MAC with nonbeacon-enabled mode. One of the key operations of 6LoWPAN, header compression, is carried out in the newly introduced adaptation layer. There are four basic header types defined in 6LoWPAN: dispatch header, mesh header, fragmentation header and IPv6 header compression header. Additional routing header is needed to be encapsulated as 6LoWPAN also proposes the routing mechanisms at adaptation layer.

In order to achieve a more lightweight protocol that maximizes bandwidth efficiency, 6LoWPAN develops two reactive routing protocols. LOAD [1.40] which is a simplified version of AODV and DYMO-low based on Dynamic MANET On-demand (DYMO°) routing protocol [1.41]. The significant feature in DYMO-low is to support either 16-bit link layer short address or IEEE 64-bit extended address. At last, hierarchical routing (HiLow) that use dynamically assigned 16-bit short address is proposed in [1.42] to save memory for larger scalability.

2.2.5.2. Advantages and features

The major advantages of adopting IPv6 for low power WPAN, as claimed in [1.43], are:

- The IP-based devices can be connected readily to other IP-based networks, without the need for intermediate entities like translation gateways or proxies. Also, the pervasive nature of IP networks allows use of existing infrastructure.

- With contiki [1.44], an open source operating system which allows battery-operated systems to communicate with the Internet, users could be interested in the available products such as IPsensor (Arch Roch) [1.45] and Jennic [1.46].

2.2.5.3. Shortcomings for our application

Although promising, we don't retain this technology for the following reasons:

- 6LoWPAN is focused on implementing IP technology in WSN. IEEE 802.15.4 nonbeacon-enabled MAC is utilized. The current routing layer also do not concern much about QoS and energy saving.
- How mesh topology could be obtained and maintained is not discussed in 6LoWPAN.
- At last, even though the header compression mechanisms are present in 6LoWPAN, the big overhead implies heavy precautions when considering the involved cost and benefit of an IP WSN.

2.2.6.Z-Wave

2.2.6.1. General description

The Z-Wave protocol is a low bandwidth half duplex protocol designed for reliable wireless communication in a low cost control network [1.47]. The protocol is not designed to transfer large amounts of data or to transfer any kind of streaming or timing critical data.

The Z-Wave protocol has 2 basic kinds of devices: controllers and slave nodes. In the case where the controller is used to create a network, it automatically becomes the primary controller. Controllers added to the network using the primary controller are called secondary controllers and don’t have the capability to include/exclude nodes in the network. Although every node in the network is capable of being controller, controllers do have additional functions that make them special in the mesh network. For example, primary controller has a full routing table and is therefore able to communicate with all nodes in the network. As shown in Figure 1.9, Z-Wave protocol consists of 5 layers [1.48].

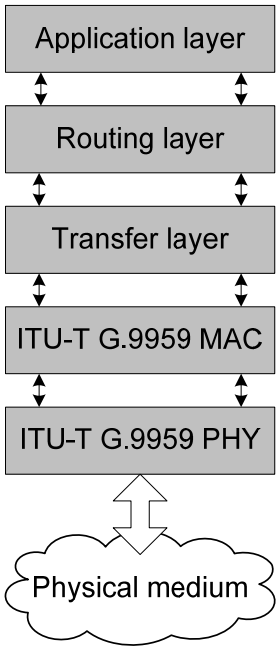

Figure 1.9 Node architecture of Z-Wave

The Z-Wave radio operates in the sub-gigahertz frequency range, around 900 MHz. the data rate can be up to 100 Kb/s depending on different modulations.

The Z-Wave MAC layer controls the radio frequency medium. Both layers are described by ITU-T G.9959 specification. The MAC layer has a collision avoidance mechanism. It is achieved by letting nodes be in receive mode when they are not transmitting, and then delay the transmission if the MAC layer is currently busy. The collision avoidance is active on all types of nodes when they have the radio activated.

Then, transfer layer controls the transfer of data between two nodes including retransmission, checksum check and acknowledgements.

The Z-Wave routing layer is responsible for routing of frames, scanning the network topology and maintaining a full routing table in the primary controller. As a source routed static network, Z-Wave assumes that all nodes in the network remain in their original detected position. Mobile devices, such as remote controls, are therefore excluded from routing.

The Z-Wave application layer is used for decoding and executing commands in a Z-Wave network. The important part of the application layer is the assignment of home ID and node ID and the replication of controllers. The rest of the application layer is implementation specific, and can be different from one implementation to another.

2.2.6.2. Advantages and features

Standing on the technical point of view, Z-Wave has no interference from Wi-Fi or other 2.4 GHz wireless technologies in similar band. In Europe, the 868 MHz band has 1% duty cycle limitation, thus a Z-Wave units can be in power-save mode and only be active 0.1% of the time. Also Z-Wave control is easily added to almost any device in few minutes. Therefore, a mesh network could be built for control, monitoring and status operations.

In market facts, there are 12 million Z-Wave products worldwide and over 700 interoperable products available. For example, 65000 devices are installed in the flagship Wynn Hotel in Las Vegas.

2.2.6.3. Shortcomings for our application

- Like IEEE 802.15.5, the mesh network built by Z-Wave contains a special node. Primary controller must have the topology knowledge of the whole network.
- Z-Wave can not provide QoS for timing critical traffic.
- Mobile devices are not acceptable as a source routed static network is assumed by Z-Wave. The advantages of mesh topology are not fully utilized.
- Z-Wave standard itself is not open and is available only to customers under non-disclosure agreement. This also affects our final decision.

2.2.7.WirelessHART

2.2.7.1. General description

WirelessHART [1.24] is a wireless mesh network communication protocol designed to meet the needs for process automation applications. For example control systems, maintenance tools and asset management applications [1.49] [1.50]. It is the wireless version of HART studied in part 2.1.4.

As shown in Figure 1.10, the specification of physical layer, data link layer, network layer, transport layer and application layer is defined.

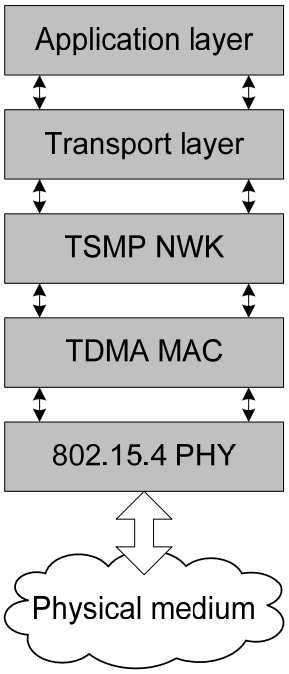

Figure 1.10 Node architecture of wirelessHART

WirelessHART has an IEEE 802.15.4 PHY with added channel hopping. The communication uses Frequency Hopping Spread Spectrum (FHSS°). Direct Sequence Spread Spectrum (DSSS°) provides coding against interference. Upon joining a network, a WirelessHART node, called node *C*, will discover available neighbors and establish communication with at least two nodes already in the network, called parent *A* and parent *B*. During this process, node *C* will receive synchronization information and a frequency hopping sequence from both parents. IEEE 802.15.4 specifies 16 channels within the 2.4 GHz ISM band, WirelessHART uses 15 of those. The hopping sequence is a pseudo-random sequence of all available channels. For example, the sequence may be 4, 15, 9, 7, 13, 2, 16, 8, 1, etc. Node *C* receives a distinct start point in the sequence from each parent, and when a new node joins it, it will give in turn a distinct start point to this new child node. In this way, each pair-wise connection is ensured to be on a different channel during each timeslot.

Two types of time slots are available in MAC layer. Shared time slots with CSMA mechanism are not commonly used. Dedicated time slots are formed by TDMA mechanism. Even though TDMA slot assignment in multi-hop networks is an NP-complete problem, WirelessHART provides a number of constraints such as slot priority and slot frequency.

The network layer employs Time Synchronized Mesh Protocol (TSMP°) which is a fully redundant mesh routing [1.51]. Fully redundant routing requires both spatial diversity

and temporal diversity. TSMP covers spatial diversity by enabling each node to discover multiple possible parent nodes and then establish links with two or more. Temporal diversity is handled by retry and failover mechanisms.

At last, transport layer and application are similar with HART technology and do not be repeated here.

2.2.7.2. Advantages and features

The merits of WirelessHART for our application can be concluded as:

- It provides QoS message delivery. Dedicated bandwidth is used for high priority and periodic communications and shared bandwidth offers elasticity for event traffic and ad hoc request/response maintenance.
- An automatic reconfiguration network could be realized and the redundant pathways in this mesh network eliminate single point of failure.
- A lot of products have been validated by HART communication foundation, such as Sitrans (Siemens) [1.52], Fisher (Emerson) [1.53] and Dust (Dust Networks) [1.54], etc.

2.2.7.3. Shortcomings for our application

On the contrary, the following points affect our decision:

- Channel hopping technology needs the communication of frequency hopping sequence information. This increases radio frequency requirements and consumes extra energy.
- In contrast to beaconing strategies, TSMP dose not begin each frame with a synchronization beacon. ACK messages which contain the offset information are exchanged to ensure alignment. In our application of low data rate, critical messages are delivered only when there are abnormal cases. Periodic beacon synchronization is therefore much more trusted as there may be no ACK exchange within a very long period of time.

- Last but not least, all the actual cases as far as we know are multi-path tree topologies even though fully redundant mesh topology is supported by the standard. The parents allocate slots for children during association phase. How to allocate the slots in a distributed and dynamic manner is not clear.

2.3.Discussion and choice

In part 2, the existing technology/standard on control and monitoring was investigated. Wired technologies are abandoned as our application focus is on low rate low mobility network and flexible infrastructure. The cost of wired network deployment is another reason. For wireless technologies, they are mainly compared according to the concerned metrics in our application. They are QoS, energy saving, mesh topology and some practical factors such as available products. Table 1.1 shows the comparison:

Table 1.1 Technologies comparison

Technology	Support QoS traffic	Energy consumption	Support mesh topology	Available products
IEEE 802.15.4	Yes	Low	No	Yes
IEEE 802.15.6	Yes	Low	No	No
ZigBee	No	Average	No	Yes
IEEE 802.15.5	Yes	Low	Yes	No
6LoWPAN	No	High	Yes	Yes
Z-Wave	No	Low	Yes	Yes
WirelessHART	Yes	Average	Yes	Yes

In conclusion, ZigBee, 6LoWPAN and Z-Wave technology are given up since they do not provide QoS mechanism for timing critical traffic. However in our application, some alarm messages may be quite important for the monitored people and should be sent in a guaranteed manner. IEEE 802.15.5 and IEEE 802.15.6 are still in development and have little or no available commercial products. So IEEE 802.15.4 and WirelessHART seem to be the good choices.

Finally, we choose IEEE 802.15.4 as the following reasons. Our work concentrates on MAC layer and many technologies such as ZigBee and 6LoWPAN are based on IEEE 802.15.4 MAC or backwards compatible with this standard. Secondly, WirelessHART still leaves many details of slot allocation, especially in a distributed and dynamic mesh network. The last consideration is due to implementation. Many types of IEEE 802.15.4 fully open

sensor application boards are today available and widely used in the scientific and academic community. Therefore, our focus is on adapting IEEE 802.15.4 to the mesh network.

3. IEEE 802.15.4 technology

3.1.Overview of IEEE 802.15.4

IEEE 802.15.4 specifications, especially the technical details related to our application requirements will be briefly introduced in this part. The focus is on MAC layer.

3.1.1.PHY layer

As mentioned in 2.2.1, IEEE 802.15.4 can operate in the 868 MHz, 915 MHz and 2.4 GHz ISM bands. The 2.4 GHz DSSS PHY employing Offset Quadrature Phase-Shift Keying (O-QPSK°) modulation is chosen as this band is standardized for unlicensed operation nearly worldwide. Data rate is therefore 250 Kb/s for this PHY.

In addition, the following tasks provided by this PHY are quite useful for us:

- Activation and deactivation of the radio transceiver.
- Energy Detection (ED°) within the current channel.
- Link Quality Indicator (LQI°) for received packets.
- CCA for CSMA/CA mechanism.

3.1.2.MAC layer

In this standard, MAC layer allows the use of a superframe structure. The different medium access methods could be achieved by the superframe. Then MAC frames are sent within the superframe and their formats are presented.

3.1.2.1. Superframe structure

As shown in Figure 1.11, time is subdivided into superframes in beacon-enabled mode. The beacons are used to synchronize the attached devices, to identify the PAN, and to describe the structure of the superframes.

A superframe is bounded by network beacons sent by the coordinator and is divided into 16 equally sized slots. Optionally, the superframe can have an active and an inactive portion. During the inactive period, the coordinator may enter a low-power mode.

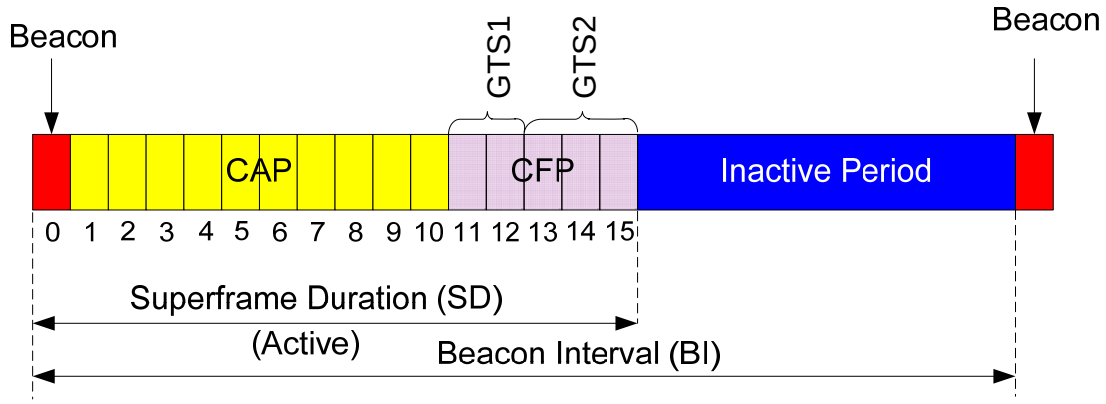

Figure 1.11 IEEE 802.15.4 superframe structure

Any device wishing to communicate during the CAP competes with other devices using a slotted CSMA/CA mechanism. On the other hand, the GTSs form dedicated CFP which always appear at the end of the active period. The PAN coordinator may allocate up to 7 of these GTSs, and a GTS may occupy more than one slot.

In addition, the superframe and its portions are defined by Beacon Interval (BI°) and Superframe Duration (SD°). BI defines the time between two consecutive beacon frames and SD defines the active period in BI. An inactive period is defined if BI > SD.

$$BI = aBaseSuperframeDuration * 2^{BO} \quad (1.1)$$

$$SD = aBaseSuperframeDuration * 2^{SO} \quad (1.2)$$

As shown in equation (1.1) and (1.2), BI and SD depend on Beacon Order (BO°) and Superframe Order (SO°). aBaseSuperframeDuration denotes the number of symbols that form a superframe when SO is 0, and $0 \leq SO \leq BO \leq 14$.

3.1.2.2. Frame format

The standard defines 4 frame structures and 3 of them will be used in our application and therefore presented in this part.

- A beacon frame used by a coordinator to transmit beacons.

- A data frame used for all transfers of data.
- An acknowledgment frame used for confirming successful frame reception.

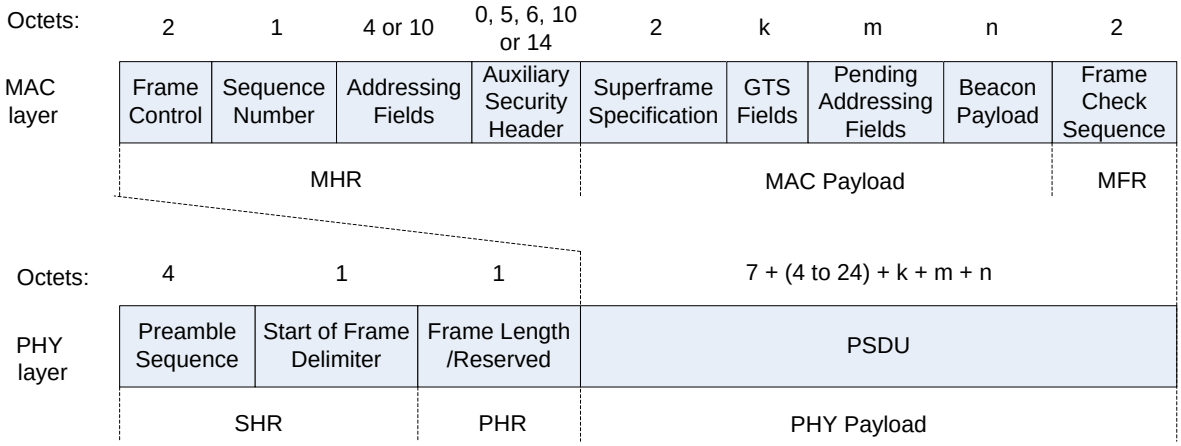

Figure 1.12 IEEE 802.15.4 beacon frame format

Figure 1.12 shows the structure of beacon frame. The MAC payload is prefixed with a MAC header (MHR°) and appended with MAC footer (MFR°). These 3 parts together form the MAC beacon frame (i.e. MPDU°) and it is then passed to PHY as PHY service data unit (PSDU°).

In MAC payload, superframe specification contains superframe organization information such as BO, SO, Final CAP Slot, etc. GTS fields and pending address fields could be used for GTS demand and management.

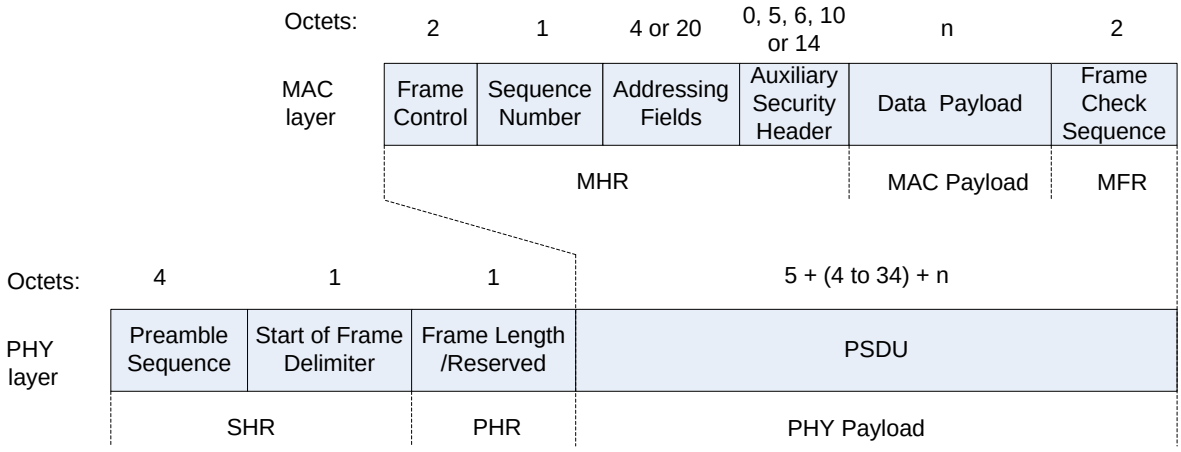

Figure 1.13 IEEE 802.15.4 data frame format

Figure 1.13 shows the structure of data frame. Destination PAN Identifier and Destination Address are added to Addressing Fields for data frame.

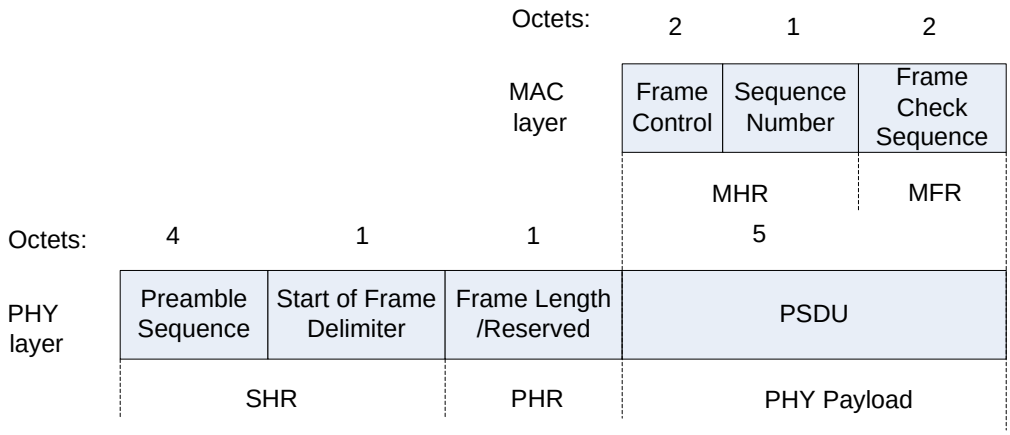

Figure 1.14 IEEE 802.15.4 acknowledgment frame format

At last, the structure of acknowledgment frame is shown in Figure 1.14. The MAC acknowledgment frame is constructed from a MHR and a MFR. It has no MAC payload.

The mesh network with 802.15.4 MAC can be achieved by some technologies such as 6LoWPAN. In which BO must be 15, then superframe specification contained in beacon frame will broadcast this information. All the received devices therefore work in non beacon-enabled mode. In this mode coordinators do not emit regular beacons. Nodes may lose their synchronization and do not know when to sleep and when to wake. GTS fields contained in beacon frame also can not arrive in time. So the QoS performance is weakened.

3.2.Challenges at MAC layer

As mentioned before, our focus is on adapting IEEE 802.15.4 to the mesh network. So the first issue is how to construct and manage a mesh network. Then, some problems followed by mesh networking are further studied and the related works are investigated.

3.2.1.Network construction and management

3.2.1.1. Mesh topology

Our application expects a mesh network which has the ability of self-organization and auto-reparation. The mesh topology also provides an easy way to build scalable network as its non-hierarchical approach. In addition, mesh topology enables route diversity, which will

make transmission more robust if an adapted routing protocol is used. However, how to construct a mesh topology is missing in IEEE 802.15.4 beacon-enabled mode.

3.2.1.2. Difficulties with mesh topology

In a mesh topology, each node is capable of communicating with any other node within its radio sphere of influence. Further network formation may depend on neighbor discovery mechanisms such as beacon exchange in IEEE 802.15.4. In particular, network management such as link failure or link establishment is quite challenging for changing mesh topology.

IEEE 802.15.4 provides two kinds of medium access control methods: CSMA/CA and GTS. However, some problems may be aroused in a mesh network

Firstly, beacons, commands and some application frames are delivered by CSMA/CA in IEEE 802.15.4. Collision probability of these frames is obviously much larger in a mesh topology. As in a star or tree topology, the supernode could allocate and manage the time offset for each branch. Network performance may be degraded by these collisions. Therefore, how to avoid collision should be considered when designing MAC layer.

The main strength of CSMA/CA is that it does not require a hierarchy in the topology such as GTS. However, it cannot offers something else than best-effort service. If GTS enables collision-free MAC, it may require a hierarchy in the MAC, typically with a star or tree topology. Therefore, how to realize GTS under mesh topology is an interesting problem.

Last but not the least, both GTS mechanism and sleep-wake mechanism, for energy saving, need time synchronization of network. In IEEE 802.15.4, a centralized PAN coordinator initiates the propagation of a synchronization beacon which is propagated along a star or tree topology to reach all the associated nodes. Therefore, new synchronization mechanism for mesh network is urgently needed.

3.2.2.Beacon collisions

Beacons are so critical in the standard. They are used to build network, to synchronize the attached nodes, to identify the PAN, to describe the superframe structure, and to require GTSs. Unfortunately, beacon collisions may occur if there is no special care on timing issues when sending beacon periodically. As shown in Figure 1.15, the node *D* in the common transmission range of coordinator *C1* and *C2* may receive beacons at approximately the same

time. Similarly, collision between data and beacon may also happen when a node sends its beacon during the active periods of its neighbors. As we mentioned before, there are more chances of collision in a mesh network.

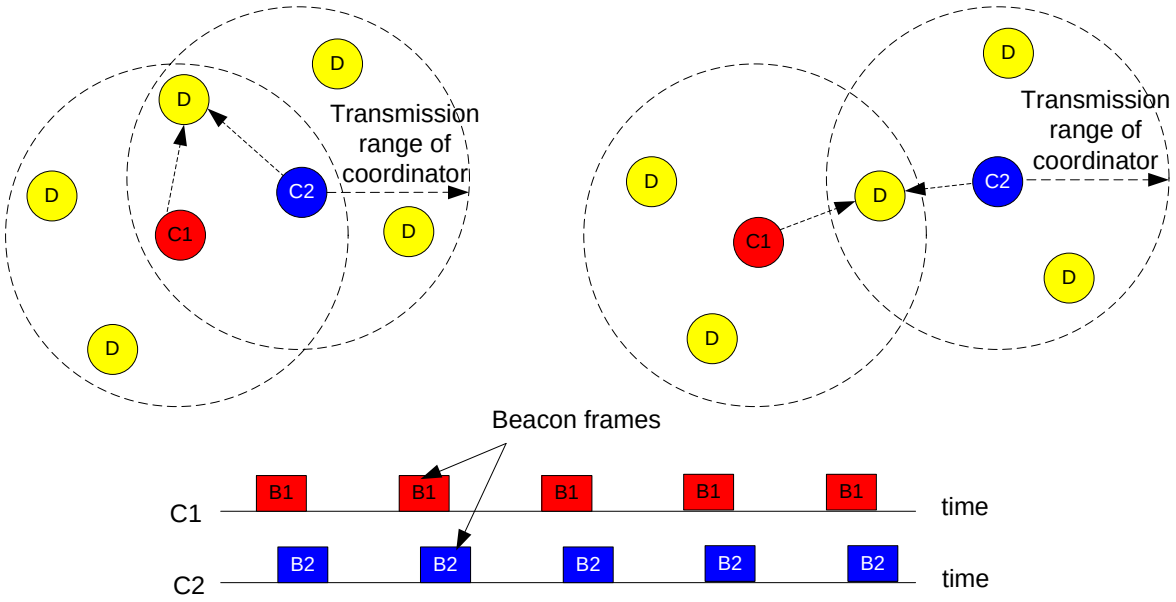

Figure 1.15 Beacon collision

3.2.2.1. Approaches to avoid beacon collision

Since there is no mechanism of avoiding beacon collision in the current IEEE 802.15.4 standard, two approaches were proposed by Task Group 15.4b [1.55]: Time Division approach (TD°) and Beacon Only Period (BOP°) approach.

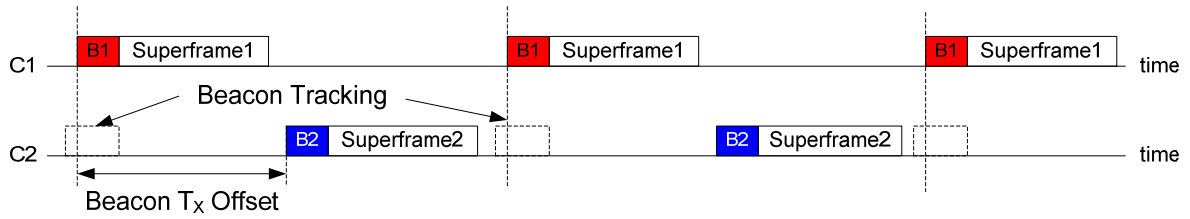

Figure 1.16 TD approach

In TD approach, time is divided such that a given coordinator sends its beacon during the inactive periods of its neighbors, as shown in Figure 1.16. The idea is that each coordinator selects a starting time, Beacon T_x Offset, to transmit its beacon. This value must be different from the starting times of its neighbors. The limitations of this approach are:

- It strictly imposes very low duty cycles.

- The direct communication between neighbors is not possible since each node operates in a time window different from its neighbors.
- Beacon T_x Offset is difficult to choose, especially for large scale network or mobile sensor network.

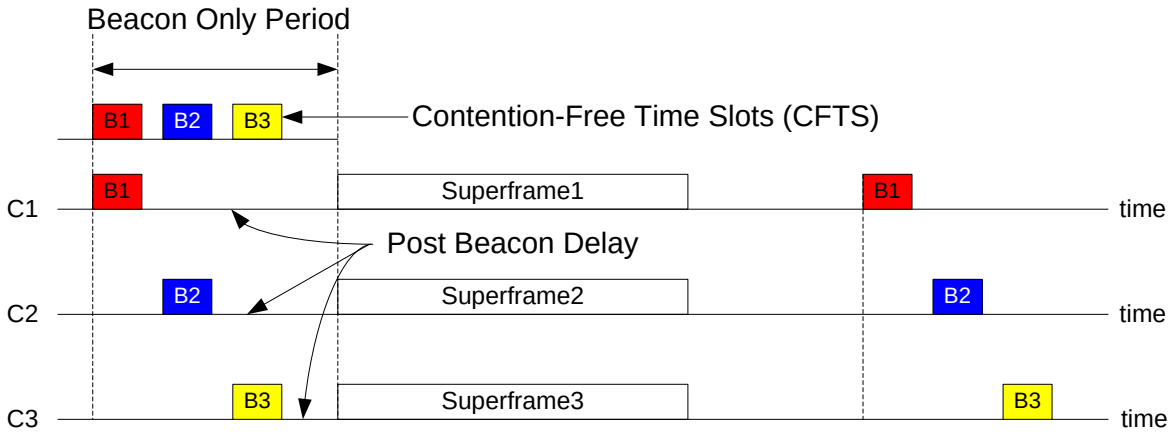

Figure 1.17 BOP approach

In BOP approach, the transmission of beacon is done in a contention-free mode, as shown in Figure 1.17. A time window, denoted as BOP, is considered at the beginning of each superframe. Each node chooses a Contention-Free Time Slot (CFTS°) such that there is no beacon collision between neighbors. The advantage of this approach compared with TD approach is that the active periods of different nodes start at the same time, thus direct communication between neighbors is possible. In addition, there is no constraint on duty cycle. On the other hand, the following works should be done and improved:

- The main complexity of this approach is CFTS allocation method and BOP dimensioning, especially when topology is changing. However, these details are missing in the proposal of Task Group 15.4b.
- As beacon collision exists not only between direct neighbors (exposed terminal problem) but also between 2-hop-away neighbors (hidden terminal problem), a hierarchical organization of CFTS is not possible in the mesh network. Unfortunately, how to choose CFTS in a distributed fashion is not involved at all.

3.2.2.2. Related works

There are some other solutions for solving beacon collision problem and they are all based on the two above approaches of Task Group 15.4b. In this part, these solutions are briefly introduced and the focus is on their limitations for our application requirements.

ZigBee specifications clear the ambiguities of IEEE 802.15.4 in a cluster-tree topology. The centralized PAN coordinator calculates and assigns a Beacon T_x Offset for each node when it wants to associate the PAN. Therefore, the network scalability and flexibility are both limited by this TD approach.

Anis Koubâa [1.56] [1.57] focuses his work in the field of cluster-tree topology. In TDBS [44], the requirement of different BI and SD for each node is calculated in advance. However, these weaken the flexibility and robustness as well as restrict the scalability of network.

Another example has been proposed in OCARI project [1.58] [1.59] [1.60]. A PAN coordinator is the destination of all association requests and allows a beacon slot for each associated node. The main drawback of this solution is also the lack of flexibility, especially regarding a changing topology and the inconstancy of wireless medium.

P. S. Muthukumaran proposed MeshMAC protocol [1.61]. This protocol enables mesh networking through a distributed TD approach in which each node calculates its schedule to transmit beacon based only on locally available information. The limitations of MeshMAC are: it imposes very low duty cycles for a large scale network; the direct communication between neighbors is not possible.

B. Carballido Villaverde proposed DBOP MAC protocol [1.62]. It creates a BOP where beacons are transmitted at different time slots among neighbors and neighbors' neighbors. However, DBOP introduces an overhead into the network. Another drawback is the inefficient management of BOP length. In addition, how to realize a distributed GTS mechanism for different QoS demands is not involved in this protocol.

4. Conclusion

In this chapter, we give an overview from application requirements to the main technologies which could be used in the home monitoring field. The motivation of our work was therefore presented.

Firstly, our application on habitat monitoring had been introduced and our work is focuses on the WSN part of the global system. This network is expected to be energy saving, flexible, robust, scalable, and to have QoS capacity. These metrics are so important factors that they will be carefully considered in protocol design and gradually tested in simulation and prototyping.

Secondly, the main wired and wireless network standards were investigated. The emphasis is on comparison of advantages and shortcomings of these technologies related to our application. IEEE 802.15.4 supports very interesting mechanisms for QoS and energy saving and has a lot of available commercial products. Therefore, we choose this technology for our application and the focus is on adapting IEEE 802.15.4 to the mesh network.

At last, IEEE 802.15.4 standard was studied. The challenges at MAC layer such as beacon collisions, changing link states and multi-hop synchronization in the mesh network were discussed. Some related works and their limitations were surveyed in order to highlight the necessity of new adapted protocols.

The following manuscript is organized as shown in Figure 1.18. Chapter 2 presents our proposition. The adapted MAC protocol tries to solve the difficulties explained in this chapter. The simulation work is presented in chapter 3 and the simulation results help us to ameliorate the MAC protocol. Chapter 4 presents our prototype work. The deployment of our sensor devices in a smart home may further verify our proposition. The three key performance criteria QoS, energy and mesh topology emphasized in this chapter will last throughout the whole work, from protocol to prototype.

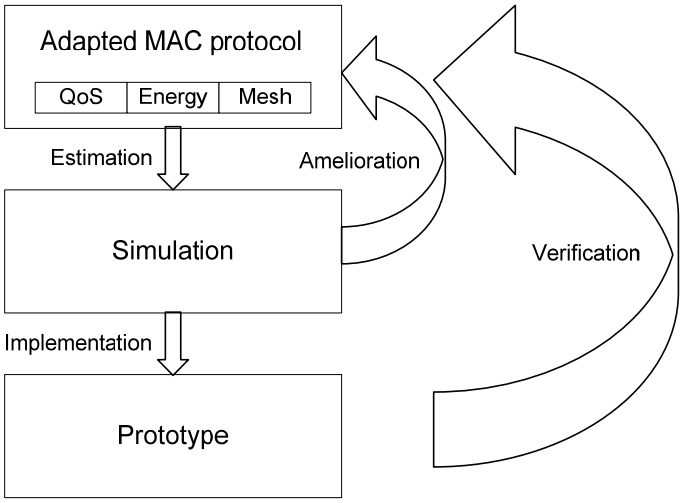

Figure 1.18 Work method and thesis framework

Reference

[1.1] M. Chan, D. Estève, C. Escriba, and E. Campo, "A Review of Smart Homes—Present State and Future Challenges", Computer Methods and Programs in Biomedicine, Volume 91, No. 1, pp. 55-81, July 2008

[1.2] S. Nourizadeh, C. Deroussent, Y.Q. Song and J.P. Thomesse, "A Distributed Elderly healthcare System", MobiHealth 2009, http://hal.inria.fr/inria-00431202

[1.3] S. Kiefer, "Implementation of a Telematic Homecare Platform in Cooperative Health Care Provider Networks", www.topcare-network.com

[1.4] V. Rialle, F. Duchene, N. Noury, L. Bajolle and J. Demongeot, "Health Smart Home: Information Technology for Patients at Home", Telemedicine Journal and e-Health, Volume 8, Issue 4, pp. 395-409, July 2004

[1.5] A. Anfosso, S. Rebaudo, "Gérontechnologies et Contrôle de L'environnement au Service du Maintien à Domicile: le projet GERHOME", Gérontologie et Société, No.136, pp. 119-131, 2011

[1.6] Y. Charlon, W. Bourennane, E. Campo, "Mise en Oeuvre d'une Plateforme de Suivi de L'actimétrie Associée à un Système D'identification", SMS 2010

[1.7] Y. Zatout, E. Campo and J.F. Llibre, "WSN-HM: Energy-Efficient Wireless Sensor Network for Home Monitoring", 5th International Conference on Intelligent Sensor, Sensor Networks and Information Processing (ISSNIP), pp. 367-372, December 2009

[1.8] Wi-Fi Alliance, "IEEE 802.11 Specification", www.wi-fi.org

[1.9] Bluetooth Special Interest Group, "IEEE 802.15.1" www.bluetooth.org

[1.10] J.R. Boulanger, C. Deroussent, "Preliminary Based Service Evaluation for Elderly People and Healthcare Professionals in Residential Home Care Units", 2nd International Conference on Digital Society (ICDS), pp. 93-101, September 2008

[1.11] I.F. Akyildiz, T. Melodia, K.R. Chowdhury, "A survey on wireless multimedia sensor networks", International Journal of Computer and Telecommunications Networking, Volume 51, Issue 4, pp. 921-960, March 2007

[1.12] I.F. Akyildiz, W.L. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: A Survey", IEEE Computer Networks, Volume 40, Issue 8, pp. 393-422, August 2002

[1.13] H.M. Ammari and S.K. Das, "Integrated Coverage and Connectivity in Wireless Sensor Networks", IEEE Transactions on Computers, Volume 57, Issue 10, pp.1423-1434, October 2008

[1.14] G.Z. Zheng, Q.M. Liu, "A Survey on the Topology of Wireless Sensor Networks Based on Small World Network Model", 2nd International Conference on Future Computer and Communication (ICFCC), pp. 67-71, May 2010

[1.15] A. Vogel, B. Kerherve, G.V. Bochmann, J. Gecsei, "Distributed Multimedia and QoS: A Survey", IEEE Multimedia Journal, Volume 2, Issue 2, pp. 10-19, August 2002

[1.16] H. Fattah, C. Leung, "An Overview of Scheduling Algorithms in Wireless Multimedia Networks", IEEE Wireless Communications, Volume 9, Issue 5, pp. 76-83, October 2002

[1.17] M.A. Yigitel, O.D. Incel, C. Ersoy, "QoS-Aware MAC Protocols for Wireless Sensor Networks: A Survey", International Journal of Computer and Telecommunications Networking, Volume 55, Issue 8, June 2011

[1.18] S. Mahfoudh, P. Minet, "Maximization of Energy Efficiency in Wireless Ad hoc and Sensor Networks with SERENA", Mobile Information Systems-Advances in Wireless Networks, Volume 5, Issue 1, pp.32-53, Juanuary 2009

[1.19] W. Allcock, J. Bresnahan, K. Kettimuthu, J. Link, "The Globus Extensible Input/Output System (XIO): A Protocol Independent IO System for the Grid", 19th International Parallel and Distributed Processing Symposium (IPDPS), pp. 8-16, April 2005

[1.20] J. Laudon, D. Lenoski, "System Overview of the SGI Origin 200/2000 Product Line", IEEE Compcon Proceedings, pp. 150-156, February 1997

[1.21] IEEE-SA Standards Board, "IEEE 802.3 Local Area Network Protocols", IEEE Standard for Information Technology

[1.22] KNX Association, "KNX Specification" http://knx.org

[1.23] Wolfgang Kohler, "Simulation of a KNX Network with EIBsec Protocol Extensions", Chapter 2, April 2010

[1.24] HART communication foundation, "WirelessHART Technical Data Sheet", http://www.hartcomm.org

[1.25] IEEE-SA Standards Board, "IEEE 802.15.4 Standard (2006) Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)", IEEE Standard for Information Technology

[1.26] IEEE-SA Standards Board, "802.15.4a (2007)", IEEE Standard for Information Technology

[1.27] Crossbow, "MicaZ Datasheet", www.xbow.com

[1.28] Crossbow, "TelosB Datasheet", www.xbow.com

[1.29] Crossbow, "Imote2 Datasheet", www.xbow.com

[1.30] http://www.ieee802.org/15/pub/TG6.html

[1.31] K.S. Kwak, S. Ullah, and N. Ullah, "An Overview of IEEE 802.15.6 Standard", 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL), February 2011

[1.32] IEEE-SA Standards Board, "IEEE P802.15.6/D06: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs) Used in or around A Body, IEEE Draft Standard for Information Technology

[1.33] S.C. Ergen, "ZigBee/IEEE 802.15.4 Summary" http://citeseerx.ist.psu.edu

[1.34] ZigBee Alliance, "ZigBee Specification", Document 053474r17, http://zigbee.org

[1.35] Freescale, "13192 Sensor Applications Reference Design", www.freescale.com

[1.36] IEEE-SA Standards Board, "Part 15.5: Mesh Topology Capability in Wireless Personal Area Networks (WPANs)", IEEE Recommended Practice for Information Technology

[1.37] M.J. Lee, R. Zhang, J.L. Zheng, G.S. Ahn, C.H. Zhu, T.R. Park, S.R. Cho, C.S. Shin, J.S. Ryu, "IEEE 802.15.5 WPAN Mesh Standard-Low Rate Part: Meshing the Wireless Sensor Networks" IEEE Journal on Selected Areas in Communications, Volume 28, Issue 7, pp. 973-983, Septembre 2010

[1.38] C.H. Zhu, J.L. Zheng, C. Ngo, T. Park, R. Zhang, M. Lee, "Low-Rate WPAN Mesh Network – An Enabling Technology for Ubiquitous Networks", IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6, April 2009

[1.39] IETF 6LoWPAN Working Group, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," IETF RFC 4944

[1.40] K. Kim, S. Park, G. Montenegro, S. Yoo, N. Kushalnagar, "6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD)", draft-daniel-6lowpan-load-adhoc-routing-03

[1.41] K. Kim, S. Park, I. Chakeres, C. Perkins, "Dynamic MANET On-demand for 6LoWPAN (DYMO-low) Routing", draftmontenegro-6lowpan-dymo-low-routing-03

[1.42] K. Kim, S. Yoo, S. Park, J. Lee, "Hierarchical Routing over 6LoWPAN (HiLow)", draft-deniel-6lowpanhilow-hierarchical-routing-00

[1.43] IETF 6LoWPAN Working Group, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", IETF RFC 4919

[1.44] http://www.contiki-os.org/

[1.45] Arch Rock, "IPsensor Node", www.archrock.com

[1.46] Jennic, "Jennic's 6LoWPAN", www.jennic.com

[1.47] Z-Wave Alliance, http://www.z-wavealliance.org/

[1.48] Niels Thybo Johansen, "Software Design Specification: Z-Wave Protocol Overview", April 2006

[1.49] P. Soldati, H.B. Zhang and M. Johansson, "Deadline-Constrained Transmission Scheduling and Data Evacuation in WirelessHART Networks", Technical Report, https://eeweb01.ee.kth.se

[1.50] A. Lehto, "WirelessHART Smart Wireless Solutions", www.hartcomm.org

[1.51] Dust Networks, "Technical Overview of Time Synchronized Mesh Protocol (TSMP)", www.dustnetworks.com

[1.52] Siemens, "Sitrans Data Sheet", www.siemens.com

[1.53] Emerson, "Fisher Manual", www.emerson.com

[1.54] Dust Networks, "SmartMesh WirelessHART", www.dustnetworks.com

[1.55] Task Group 15.4b, http://grouper.ieee.org/groups/802/15/pub/TG4.html

[1.56] A. Koubaa, A. Cunha, M. Alves, "A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks", 19th Euromicro Conference on Real-Time Systems (ECRTS), pp. 125-135, July 2007

[1.57] A. Koubâa, M. Alves, M. Attia, A. Van Nieuwenhuyse, "Collision-Free Beacon Scheduling Mechanisms for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks", 7th International Workshop on Applications and Services in Wireless Networks (ASWN), May 2007

[1.58] K. Alagha, G. Chalhoub, A. Guitton, E. Livolant, S. Mahfoudh, P. Minet, M. Misson, J. Rahme, T. Val, A. van den Bossche, "Cross-Layering in An Industrial Wireless Sensor Network: Case Study of OCARI", Journal of Networks, Volume 4, Issue 6, pp. 411-420, August 2009

[1.59] T. Dang, C. Devic, E. Livolant, A. Van Den Bossche, T. Val, "OCARI: Optimization of Communication for Ad Hoc Reliable Industrial Networks", 6th IEEE International Conference on Industrial Informatics (INDIN), pp. 688-693, July 2008

[1.60] E. Livolant, A. Van Den Bossche, T. Val, "MAC Specificaitons for A WPAN Allowing Both Energy Saving and Guaranteed Delay Part B: Optimization of the Inter-Star Exchanges for MaCARI", Computer Science-Wireless Sensor and Actor Networks, Volume 264, pp. 233-244

[1.61] P.S. Muthukumaran, R. Alberola, R. Spinar and D. Pesch, "MeshMAC: Enabling Mesh Networking over IEEE802.15.4 through Distributed Beacon Scheduling", AD HOC Networks, Volume 28, Issue1, pp. 561–575, January 2010

[1.62] B.C. Villaverde, R. Alberola, S. Rea, D. Pesch, "Experimental Evaluation of Beacon Scheduling Mechanisms for Multihop IEEE 802.15.4 Wireless Sensor Networks", 4th Conference on Sensor Technologies and Applications (SENSORCOMM), pp. 226-231, July 2010

Chapter 2

Improving Robustness and Flexibility of MAC Layer

In order to improve robustness and flexibility of IEEE 802.15.4 MAC layer in a mesh WSN, we propose in this chapter an adaptive and distributed collision free MAC protocol. The objective of this protocol is to build a beacon-enabled MAC over IEEE 802.15.4 PHY which supports mesh topology and enables guaranteed service with low energy consumption. The network formation, the node architecture, the protocol function and its operation details will be described in this chapter.

1. Adaptive and Distributed Collision Free MAC

1.1.General description

This part presents an original protocol named as Adaptive and Distributed Collision Free (ADCF°) MAC. ADCF aims to improve robustness and flexibility of IEEE 802.15.4 MAC, which means the capacity of self-organization and auto-reparation. Simultaneously, ADCF should enable energy efficiency and guaranteed slots negotiation [2.1] [2.2]. Firstly, basic characteristics, mesh network formation and the architecture of ADCF are illustrated. In functional overview, mechanisms are proposed with ADCF superframe and beacon frame.

1.1.1.Basic characteristics

As explained in chapter 1, ADCF is designed for applications with limited energy power and relaxed throughput requirements. The network is expected to be easy to install, to allow reliable data transfer, low cost and reasonable power consumption. So the basic characteristics of this WSN are:

- IEEE 802.15.4 PHY works in 2.4 GHz band with data rate of 250 kb/s. As seen in chapter 1, this PHY layer offers ED, LQI, CCA and activation/deactivation of radio transceiver.
- Only FFDs are applied for mesh operation of this WSN. Each FFD has the same function and can talk to any other FFDs within its communication range. In other words, all the devices of this network are supposed to have the capacity of both sensor and router.
- Each FFD has an allocated 16-bit short address. We can preliminary set the node address during installation.
- There are CSMA/CA channel access and optional allocation of contention-free slots. Additional mechanisms such as beacon scheduling and acknowledgement are provided for transfer reliability and effectiveness.

- The devices in this WSN are supposed to be synchronized. This concern is out of the scope of this work but a clock synchronization algorithm will be considered in perspectives.

- Low power consumption is a fundamental issue of this WSN. As the largest energy consumption of nodes is due to the non efficient transceiver activities such as idle listening [2.3], so timeslot allocation mechanisms are prudently considered in this mesh WSN.

ADCF is based on the IEEE 802.15.4 2.4 GHz DSSS physical layer and classical superframe structure. On the one hand, ADCF proposes a distributed beacon scheduling mechanism in a beacon only period which spatially reuses the timeslots over 2-hop. On the other hand, the contribution of ADCF lies in a data slot allocation mechanism which makes GTS possible in a mesh topology.

1.1.2.Network topology

1.1.2.1. Mesh network formation

A mesh network can be ad hoc, self-organizing, and self-healing. It may also allow multiple hops to route messages from any device to any other device in the network.

In our WSN, each ADCF node will listen to the channel when starting up. Then the node could begin to talk with any other node within its communication range. Beacon frames containing a list of neighbors are exchanged between nodes so that each node has a partial knowledge of the 2-hop neighborhood. If a node joins the network or if a node fails, ADCF could repair the network automatically. Therefore, a mesh network is built.

In reality, node failure can be caused by energy exhaustion, unexpected damage or even the poor quality of wireless link. So a node considers the failure of its neighbor node only when the neighbor's beacon loss is above a threshold. Similarly, new node is regarded as neighbor only when its beacon loss is below a threshold. In our proposal, beacon loss threshold is a predefined parameter which depends on different wireless environments during the specific implementation. In other words, we estimate the wireless link quality by beacon loss. Only the nodes connected by high quality link are considered as 1-hop neighbors.

Asymmetric links are possible and acceptable. More details and parameters will be illustrated in 2.2.6.

1.1.2.2. Initiator

As IEEE 802.15.4, ADCF allows using a superframe structure. In our WSN, the first node which begins to schedule the superframe is called as initiator. Initiator is not a supernode as each node in the network may be selected as initiator; initiator is not a supernode as it only has a partial knowledge of the 2-hop neighborhood, like the other nodes; initiator is not a supernode as the network could work properly when initiator fails. Therefore initiator is just a node chosen for indicating the beginning of each superframe.

1.1.3.Architecture

As shown in Figure 2.1, an ADCF node comprises a PHY layer, which contains the RF transceiver along with its low-level control mechanism, and a MAC layer that provides access to the physical channel for all types of transfer. Static or dynamic routing mechanisms may be added in the upper layer.

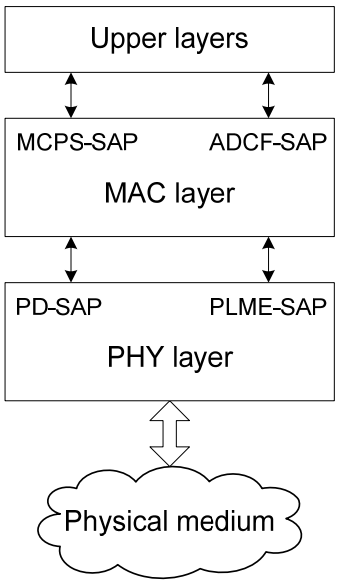

Figure 2.1 ADCF node architecture

In particular, PHY layer provides two services through Service Access Point (SAP°): PHY Data (PD°) SAP and Physical Layer Management Entity (PLME°) SAP. MAC layer conceptually includes a management entity called ADCF-SAP and a MAC data service accessed through the MAC Common Part Sublayer (MCPS°) data SAP. These service

interfaces serve to define the logical links between different layers and will be further described in part 2.3 of this chapter.

1.2.Functional overview

An overview of the general functions of ADCF is given in this part. It includes information on the superframe structure, the beacon frame format and the proposed medium access control mechanisms.

1.2.1.Superframe structure

As shown in Figure 2.2, the time is divided into superframes and each superframe includes three parts {BOP, Active Period, Inactive Period}.

In the BOP, each node sends its beacon in a guaranteed slot called Collision Free Beacon Slot (CFBS°). Active period is divided into 16 equally sized slots as a classical IEEE 802.15.4 superframe. It starts with the CAP where medium accesses are done by using the classical CSMA/CA protocol. It ends with the CFP where medium accesses are done by using an original protocol which provides GTS-equivalent for the mesh topology. This original protocol is based on the Collision Free Data Slot (CFDS°). After active period, the optional inactive period allows all the nodes to go into sleep mode to save energy. Our contributions concern CFBS and CFDS as shown in Figure 2.2.

To schedule a superframe, the basic parameters of ADCF such as BO and SO are consistent with IEEE 802.15.4. The BOP length is variable according to the network parameters such as network density. Therefore, two options are available for organizing an ADCF superframe.

- Option 1: active period and inactive period are fixed by BO and SO. With the dynamic BOP, we obtain therefore the dynamic superframe.
- Option 2: the superframe is set as a constant. With the dynamic BOP and the fixed active period, ADCF nodes could keep a flexible inactive period.

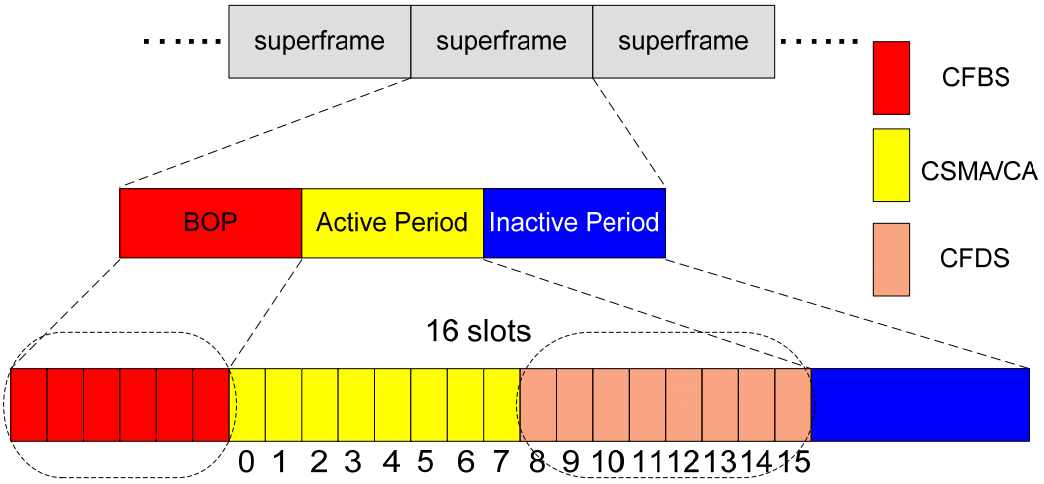

Figure 2.2 ADCF superframe structure

In chapters 3 and 4 dedicated to simulation and prototype, specific details and suitable parameters of both options will be given.

1.2.1.1. CFBS mechanism

The importance of beacon has been explained in chapter 1. To achieve the objectives of ADCF, the CFBS mechanism is proposed. Each ADCF node sends its beacon in its own CFBS. This beacon announces the presence of the node and the presence of its 1-hop neighbors, maintains network synchronization and is used to request/reply neighbor data slot negotiations, i.e. CFDS. As there is scarcely interference at distance of more than 2 hops [2.4], the nodes far away more than 2 hops could reuse the same CFBS to enhance channel reutilization. In other words, nodes must determine their CFBSs by taking into account CFBSs used by their 2-hop neighborhoods with a certain priority, as detailed in section 2.2.2.

1.2.1.2. CSMA/CA mechanism

The contention-based CSMA/CA protocol remains in active period for best-effort traffics or new nodes wishing to join the network. ADCF uses two types of channel access mechanism depending on the network state (*initialization* or *working stage*).

In the *initialization stage*, the superframe is not formed yet, so the classical unslotted CSMA/CA mechanism is used for beacon delivery. Each time a node wishes to transmit a beacon frame, it waits for a random period. If the channel is found to be idle following the random backoff, the node transmits its beacon. If the channel is found to be busy following

the random backoff, the node waits for another random period before trying to access the channel again, until 3 retries.

When a particular node could schedule the superframe, i.e. it has a CFBS, it enters in the *working stage* and the number of converged nodes in the network grows. The classical slotted CSMA/CA in which the backoff slots are aligned with the start of the superframe is used at this moment. Each time a node wishes to transmit beacon frame or data frame, it locates the boundary of the next backoff slot and then waits for a random number of backoff slots. If the channel is busy, following this random backoff, the node waits for another random number of backoff slots before trying to access the channel again. If the channel is idle, the node begins transmitting on the next available backoff slot boundary. Acknowledgment, if requested by data frame, is sent without using CSMA/CA.

1.2.1.3. CFDS mechanism

Because of the imposed hierarchy between end devices and coordinator in the star and tree topology, GTS mechanism of IEEE 802.15.4 has been replaced by CFDS mechanism in ADCF. For the time-bounded traffic or the traffic of zero-tolerance packet loss, CFDS enable these traffics to be sent in some dedicated slots. Medium access can be done directly, without backoff delays or medium sensing.

Thanks to CFBS, the CFDS negotiations can be achieved between the source node and the destination node using beacon frames. Similarly, nodes must determine their own CFDSs by taking into account CFDSs used by their 2-hop neighborhoods. CFDS mechanism allows point-to-point bidirectional communications in the mesh topology. In addition, a node can request several consequent CFDSs to a neighbor node or several neighbors, as detailed in section 2.2.5.

Especially, CFDS mechanism enables contention-free communications for multiple hops traffics. In this work, the peer-to-peer CFDS negotiation protocol is fully described and evaluated. The multi-hop end-to-end reservation method may depend on routing level. This point is not considered here.

1.2.2.Beacon frame format

There are three types of frame in ADCF. Data frame and acknowledgment frame are the same with that in IEEE 802.15.4 as illustrated in chapter 1. In order to achieve the objectives of ADCF, beacon frame structure is modified as given in Figure 2.3.

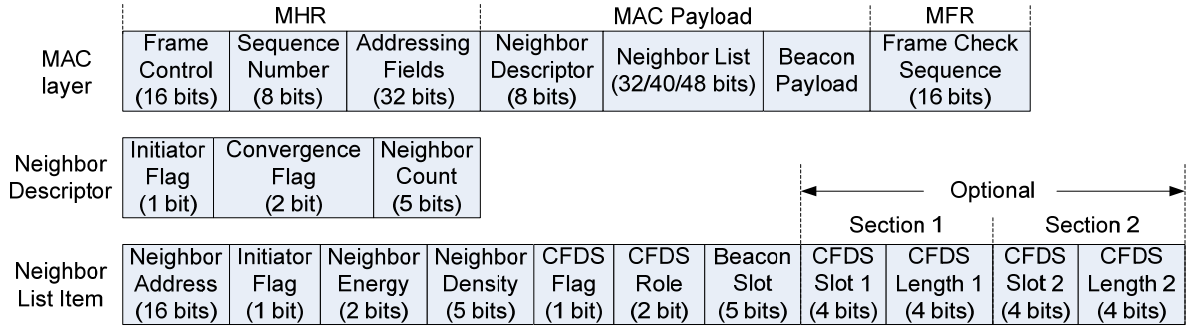

Figure 2.3 Beacon frame format

The modifications take place in MAC payload. *Superframe Specification*, *GTS Fields* and *Pending Addressing Fields* have been replaced by *Neighbor Descriptor* and *Neighbor List*. The two new fields shall be formatted as follows:

- Initiator Flag (IF°, 1 bit): IF in Neighbor Descriptor indicates the role of the source node. IF in one Neighbor List item indicates the role of this neighbor node. IF is cleared by default and set if the node becomes initiator.

- Convergence Flag (CF°, 2 bits): it indicates the state of the source node. If CF is 0, the source node is in the *initialization stage* and sends its beacon by using unslotted CSMA/CA. If CF is 1, the source node knows the BOP information and begins to choose its beacon slot. In this case, the node sends its beacon by slotted CSMA/CA in active period. If CF is 2, the source node is in the *working stage* and sends its beacon in the organized BOP.

- Neighbor Count (NC°, 5 bits): it indicates the number of items in Neighbor List.

- Neighbor Address (NA°, 16 bits): it indicates the 16-bit address of the neighbor node.

- Neighbor Energy (NE°, 2 bits): it indicates the residual energy level of the neighbor node.

- Neighbor Density (ND°, 5 bits): it indicates the number of neighbors within 2 hops, including the node itself.

- CFDS Flag (1 bit): it indicates the presence of a CFDS request for this neighbor node: if it is cleared, there is no CFDS field (section 1 and section 2 as shown in Figure 2.3) for the neighbor node. If this flag equals 1, the neighbor node uses CFDS so that the corresponding CFDS field could be extracted.

- CFDS Role (2 bits): it indicates the role of the neighbor node when using CFDS. If it is 0, the neighbor node does not use CFDS for communicating with the source node of this beacon. Section 1 may present in order to record the CFDS used for the communications between other 2-hop neighbors. If it is 1, the neighbor node is a source for the CFDS request. Optional section 1 is present in the beacon frame and will be extracted when receiving by the destination. If it is 2, the neighbor node is a destination for the CFDS response. Section 1 also exists in the beacon frame. If it is 3, the neighbor node has a two-way communication with the source node of the beacon. Both section 1 and section 2 will be analyzed in this situation.

- Beacon Slot (5 bits): it indicates the beacon slot number of the neighbor node.

- CFDS Slot (4 bits): it indicates the first data slot number negotiated between the nodes.

- CFDS Length (4 bits): it indicates the amount of data slots for the neighbor node, which depends on the application demands.

Generally, beacon frames are exchanged between nodes so that they can collect the interesting information about their neighbors. All the information is stored and updated in the node's Neighbor Table (NT°) and is critical for making decisions in the proposed mechanisms.

In addition, complying with the maximum packet length (127 bytes) in IEEE 802.15.4, ADCF allows at most about 27 neighbors for a node or 18 neighbors if the node has a bidirectional communication with all its neighbors at the same time. We think that it is realistic and acceptable.

2. Operation of ADCF

How to realize the functions of ADCF along with the proposed mechanisms? In this part, we will illustrate the operation of ADCF by a set of slight protocols/algorithms.

2.1. General description

2.1.1. Basic definitions

Before to detail the protocols/algorithms, some standardized basic definitions on the network are given.

- N: it indicates the expected number of ADCF nodes in the network. This value is a parameter of the network configuration and can be decided by the application layer during installation.

- H_{max}: it indicates the maximum number of hops in the network. The value is decided by the physical parameters such as communication range of transceiver.

- D_{max}: it indicates the maximum *Neighbor Density* in the network and can be obtained by the beacon communications between nodes. In fact, D_{max} is related to H_{max} and N. The three parameters could be adaptive and optimized by the management of the network configuration, i.e. suitable topology control algorithm may be considered in perspective.

- T_B: the time duration of a CFBS. Based on the evaluation of beacon transmission time and switching time of transceiver, 10 ms seems to be enough and will be used.

- T_D: the time duration of a CFDS. We can determine this value by using the MAC-layer parameters such as SO.

- T_{cycle} (Time Step): the time interval between two beacons. It is a constant in the *initialization stage* and must be calculated in the *working stage*, by using the organized superframe structure.

- T_{sample}: the number of *Time Step* to listen the medium before talking, i.e. sending the first beacon in the *initialization stage*. 3 * T_{cycle} may be set by default.
- L: the length of a beacon frame. We can obtain the length of each beacon frame by the communications between nodes.

2.1.2.Operational processes

ADCF is divided into several slight protocols and associated algorithms in order to simplify the comprehension of the whole process. Specifically, they are *Beacon Exchange Protocol* (BEP°), *Initiator Selection Protocol* (ISP°), *Beacon Slot Allocation Protocol* (BSAP°), *Data Slot Allocation Protocol* (DSAP°) and *Smart Repair Protocol* (SRP°). In addition, *Simple Priority Algorithm* (SPA°) is used repeatedly both in ISP and BSAP for not only deciding the relative priority but also minimizing the protocol cost.

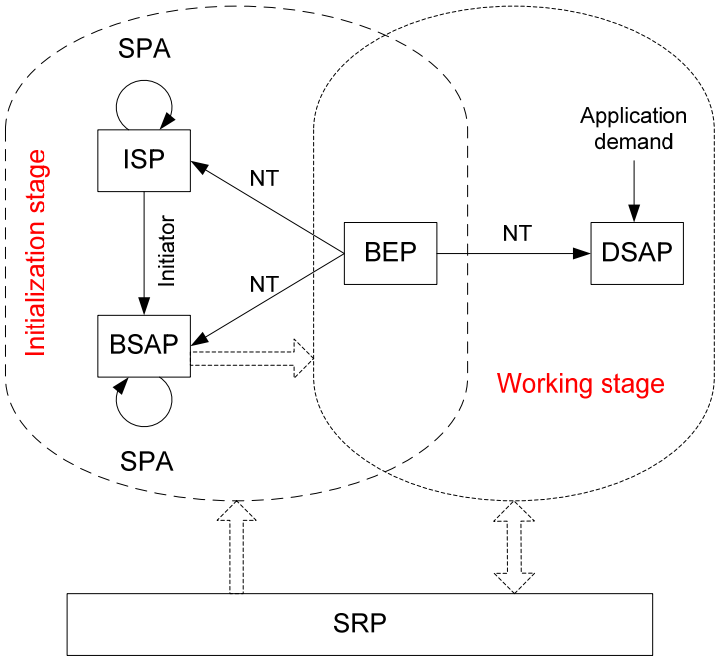

Figure 2.4 ADCF operation diagram

As shown in Figure 2.4, SRP allows ADCF node to switch between two stages: *initialization stage* and *working stage*. The beginning and the core of ADCF is BEP which sets up and updates NT in both stages. With the information in NT, ISP is executed. Then BSAP is triggered when the initiator is decided. As each node knows BOP length (D_{max}), it can calculate the beginning of a superframe by the time of a received neighbor beacon and its slot number. At this moment, the ADCF node enters in the *working stage*. The number of

converged nodes in the network grows gradually, until the entire network converges. At last, DSAP is triggered by a request from a higher layer.

The details of each protocol and associated algorithm are illustrated in the following part 2.2. In addition, as ADCF is a distributed protocol, it is essential to investigate the protocol cost. So *Convergence Time* (T) and *Message Overhead* (M) are theoretically studied in each stage. Typically, T indicates the time duration from a topological change to a valid working stage; M indicates the messages exchanged between nodes from a topological change to a valid working stage.

2.2.Proposed protocols/algorithms

2.2.1.Beacon Exchange Protocol

The main concern of BEP is to build and update NT by beacon exchange. Some interesting information, such as CF, ND and beacon slot number etc., is extracted from the received beacons.

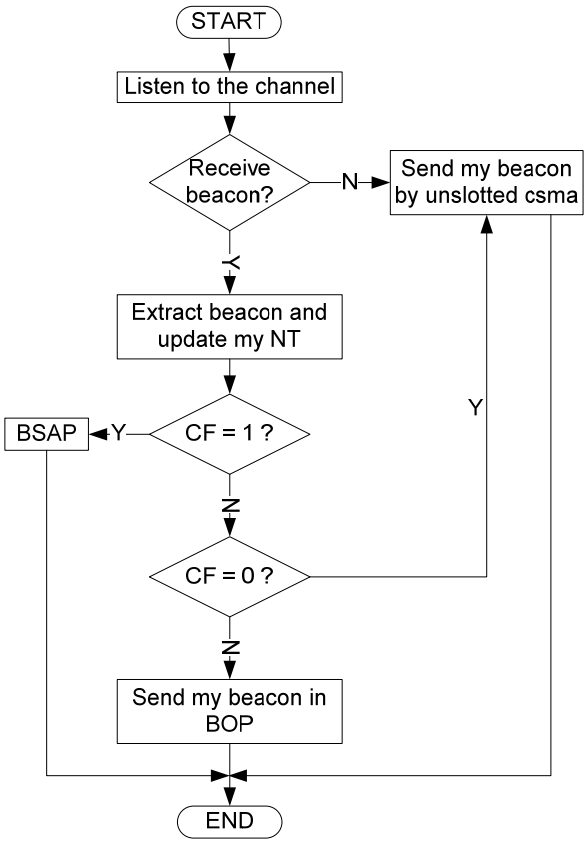

Figure 2.5 BEP flowchart

In the *initialization stage*, a new node will firstly listen to the channel for a fixed period T_{sample}. If there are no received beacons after this listening, the new node sends its beacons by unslotted CSMA/CA. Depending on different CF values in received beacons, the node may enter different states and so send its beacons by different mechanisms. If CF is 0, the node sends its beacons by unslotted CSMA/CA. If CF is 1, the node enters BSAP to choose a beacon slot. If CF is 2, the node sends its beacons directly in its beacon slot. Figure 2.5 shows the protocol flowchart.

In BEP, each node broadcasts its beacons within 1 hop and records direct neighbors' information in its NT. The following beacons include the node's information and its direct neighbors' information (transmission items in beacon). At last, all the 2-hop neighbors' information is obtained by the node; this information is stored in the NT. Ideally, the nodes need $2T_{cycle}$ to collect the information of 2-hop neighbors.

2.2.2.Simple Priority Algorithm

SPA enables to decide between two nodes *u* and *v*. The winner among the two nodes is noted *w*.

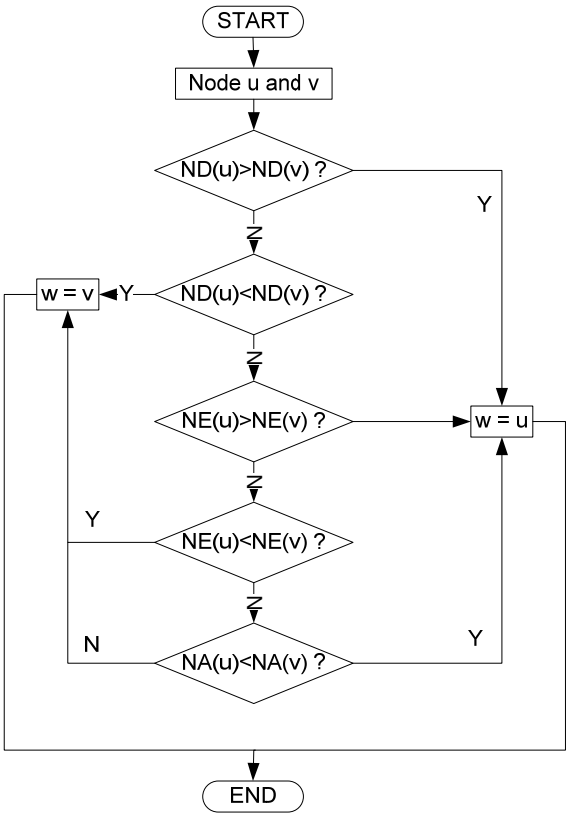

Figure 2.6 SPA flowchart

As shown in Figure 2.6, SPA is implemented by comparing 3 parameters of the nodes. The comparison order is ND, NE and NA. At first, the node with maximum ND which can reduce the protocol cost as much as possible [2.3] is considered as the one with highest priority. If the nodes have the same ND, SPA chooses the one with maximum NE. Finally, the node with minimum address has the highest priority if the first two values are equal.

2.2.3.Initiator Selection Protocol

The objective of ISP is to select an initiator which specifies the beginning of BOP/superframe. When the initiator is decided, its information is broadcasted through the network. In other words, the initiator's information is both a transmission item in all the nodes' beacons and a storage item in all the nodes' NTs.

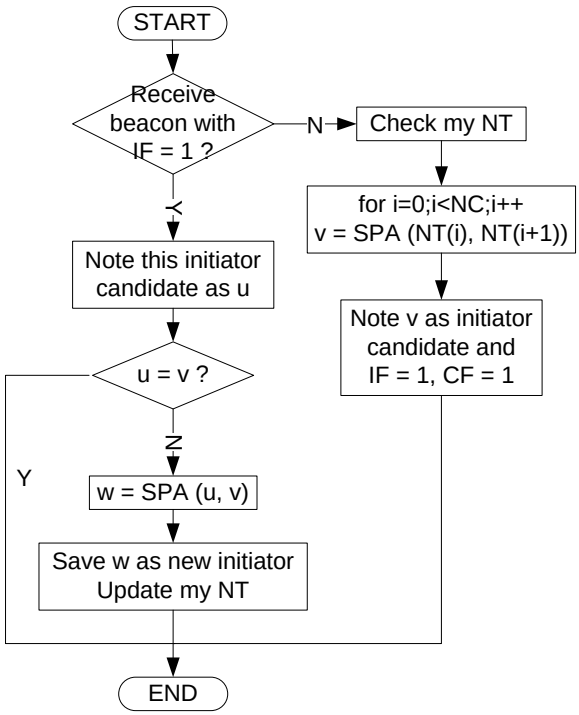

Figure 2.7 ISP flowchart

As shown in Figure 2.7, when the node does not receive the beacon with its IF as 1, it locally selects an initiator candidate (e.g. node *v*) by SPA from its NT. So the node *v*'s IF and CF are set to 1 at this moment. These changes will be broadcasted in the following beacons. When other nodes receive the beacons which indicate the different initiator candidates (e.g. node *u*), SPA is repeatedly used to decide a unique initiator for the network. This procedure takes at most H_{max} time steps.

In addition, the BOP length is measured by the initiator's ND (D_{max}). As initiator's information is broadcasted without limitation on hops, each node knows the BOP length and updates this length dynamically.

2.2.4.Beacon Slot Allocation Protocol

This protocol makes each node choose a CFBS in the BOP. As shown in Figure 2.8, when a node's CF is not 0, the node knows the BOP length and begins the beacon slot selection. The node executes SPA locally to check its priority. The node with higher priority will choose its beacon slot earlier. Obviously, initiator has the highest priority and therefore occupies the first beacon slot. The following node will choose a free beacon slot which is not used by its 2-hop neighbors. When the beacon slot is decided, the node sets CF as 2. This procedure takes at most H_{max} * D_{max} time steps.

Each node runs BSAP with the information (e.g. beacon slot number) provided by its NT. So a node can not know the information of more than 2-hop. Alternatively, the nodes at distance of more than 2-hop could choose the same beacon slot. For example, a node, far away from initiator, may reuse the first beacon slot.

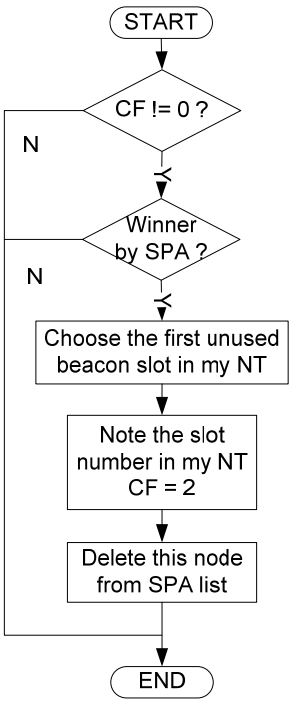

Figure 2.8 BSAP flowchart

At the end of BSAP, the network goes into working stage. Therefore, convergence time and message overhead of initialization stage are shown as (2.1) and (2.2).

$$T \le (T_{sample} + 2 + H_{\max} + H_{\max} * D_{\max}) * T_{cycle} \quad (2.1)$$

$$M \le ((2 + H_{\max} + H_{\max} * D_{\max}) * N * L) / T \quad (2.2)$$

In many cases, ADCF may be in working stage without being disturbed by topological changes such as node joining or node failure (details in part 2.2.6). However, we also consider the worst case with a network rebuilding. So the nodes restart ISP and BSAP, like the same procedure in initialization stage. In this case, convergence time and message overhead are shown as (2.3) and (2.4).

$$T \le (H_{\max} + H_{\max} * D_{\max}) * T_{cycle} \quad (2.3)$$

$$M \le ((H_{\max} + H_{\max} * D_{\max}) * N * L) / T \quad (2.4)$$

2.2.5.Data Slot Allocation Protocol

This protocol allows the CFDS negotiations between source node and destination node by beacon exchanges. Figure 2.9 illustrates DSAP flowchart for both of them.

For source node, DSAP begins when there is application traffic to be transmitted by CFDS. This upper layer traffic contains the destination address and the length of requested CFDS. Source node set its CFDS Flag and CFDS Role as 1. At this moment, NT should be updated. CFDS Flag and CFDS Role of source node are certainly equal to 1. CFDS Flag of destination node is set to 1 and CFDS Role of destination node is set to 2 for distinguishing destination node from all the broadcasted neighbors. Then source node sets its beacon with NT and sends it by BEP. When source node receives a beacon from destination node, CFDS Slot and CFDS Length are extracted. Source node will note the two attributes if they are valid values. At last, source node updates its NT with the allocated CFDS slot number and CFDS length. The new time schedule is also calculated in order to achieve this direct collision-free transmission. As there is no extra overhead for the CFDS negotiation, source node can continue requesting until a valid CFDS is found by the destination node.

So for destination node, the objective of DSAP is to search the available CFDS and respond to source node. The slot number must be decided by the receiver (destination) of the

traffic to avoid frame collision. Figure 2.10 gives an example about message sequence between a source and a destination. Firstly, CFDS Flag is set as 1 and CFDS Role is set as 2 when destination node receives the indicated beacon. The first CFDS (e.g. slot 9) which is not used by the 2-hop neighbors may be used. When source node requests several CFDSs, CFDS Slot is noted as 9 if the following CFDSs are also available. Else, destination node searches until the end of active period (slot 15). If the available CFDSs are not enough, destination node also returns to the first available CFDS and provides CFDS services as possible as it can. At last, destination node updates its NT with the new flags and slot number and sends the beacon by BEP.

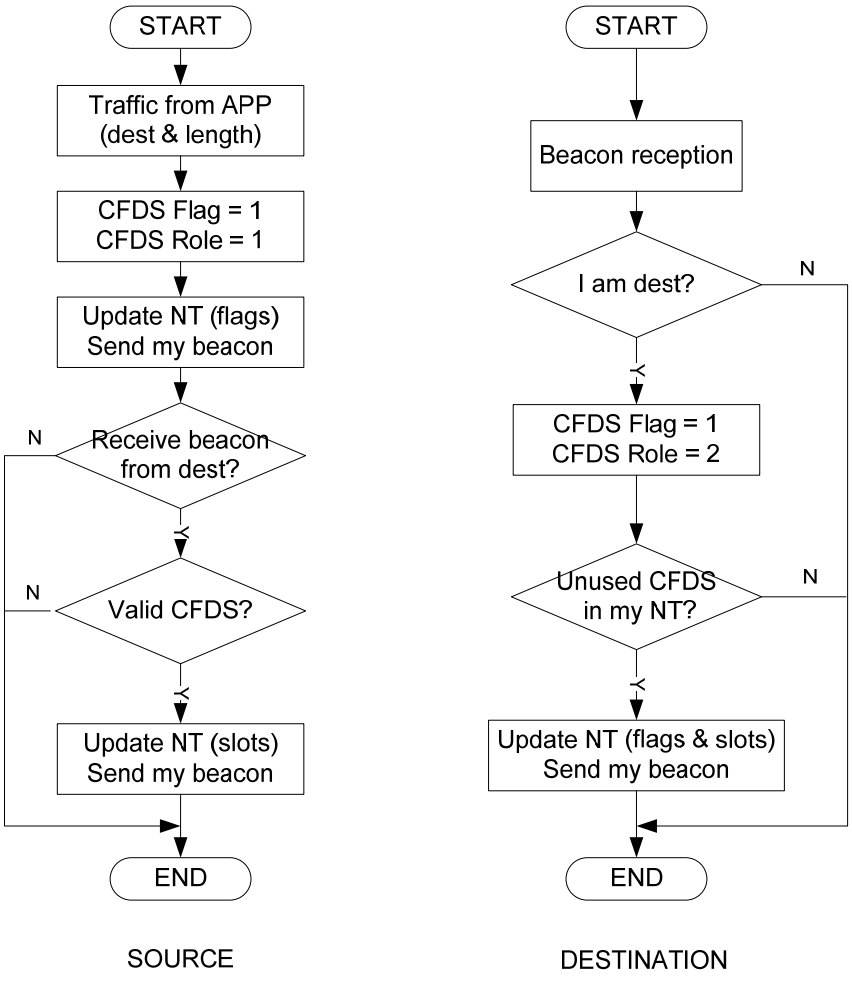

Figure 2.9 DSAP flowchart

The CFDS deallocation is also invoked by source node of this CFDS. Source node clears up all the flags and slot number and sends the beacon by BEP. Destination node also clears up all the CFDS information with this source node from its NT when it receives the indicated beacon. Therefore, the deallocation is complete and the CFDS is free for other nodes.

In addition, DSAP allows a bidirectional communication. When a destination node wishes to reserve CFDS with the corresponding source node, it launches the same procedure as single-direction communication shown in Figure 2.9. CFDS Role becomes 3 to indicate both source and destination of a node.

In summary, DSAP supports the CFDS negotiations in a multi-hop mesh network. An ADCF node can reserve one CFDS or several CFDSs. Then the node could send or receive the data packets without collision in the negotiated slot. If no CFDS used by the node at all, it may sleep during these data slots for further energy saving.

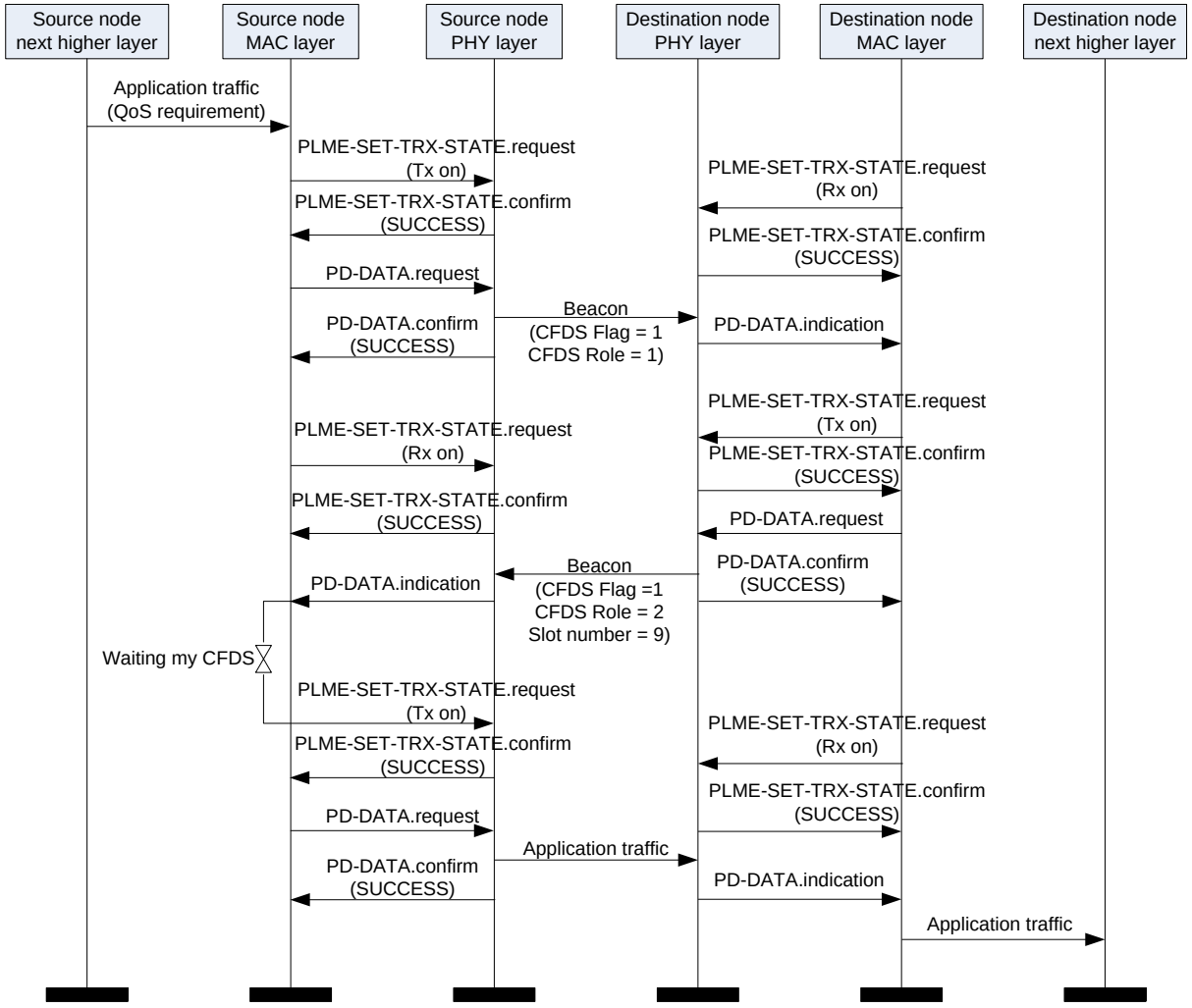

Figure 2.10 DSAP message sequence chart

2.2.6.Smart Repair Protocol

SRP attempts to minimize the impact of a change of topology as much as possible. This protocol is critical in ADCF as it improves network flexibility and robustness. Generally, the

topology changes are classified as four types: node join, node failure, network separation and network integration.

As explained in 1.1.2, we estimate the wireless link quality by beacon loss. For example Figure 2.11 illustrates the link state between two nodes *u* and *v*. Once node *u* receives a beacon from node *v*, it labels node *v* as preliminary in its NT. This label becomes unconfirmed after *k* consequent beacons and then confirmed after *l* consequent beacons. When *m* beacons are loss, node *v* returns to unconfirmed state. At last, node *v* may be considered as confirmed or be deleted after the loss of *n* beacons. The parameters *k*, *l*, *m*, *n* are depending on specific wireless environment and will be studied in the prototype.

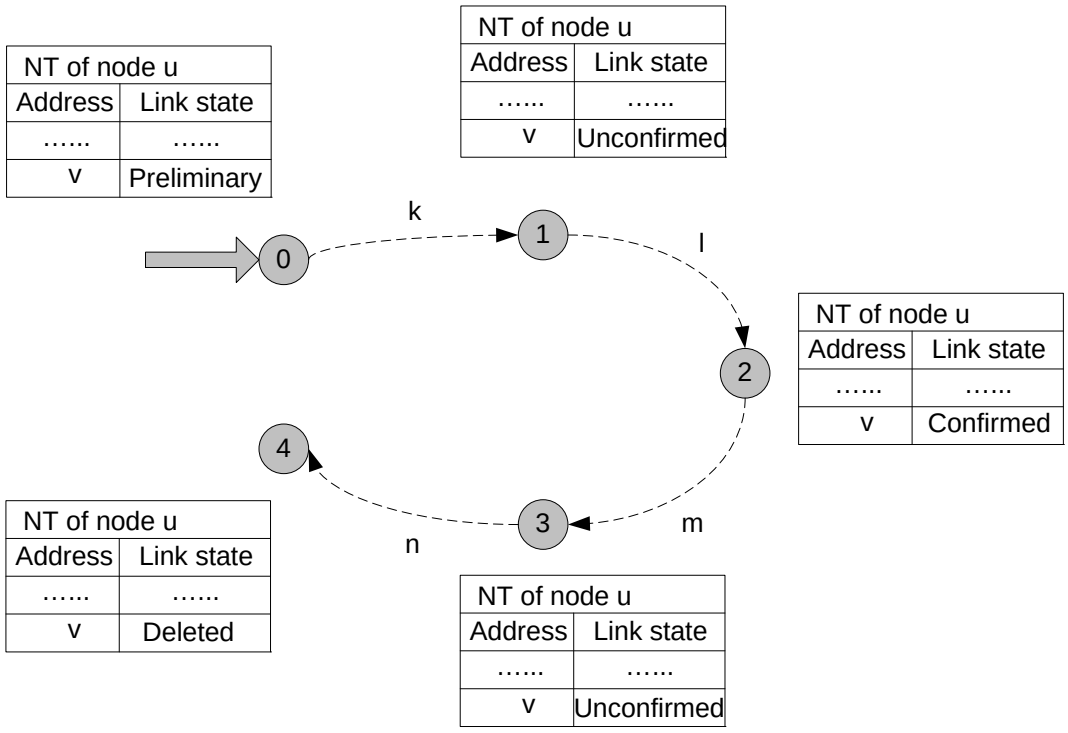

Figure 2.11 Link state transition

Therefore, node join or node failure here means the presence or disappearance of one node in its neighbors' NT. Both unconfirmed nodes and confirmed nodes are stored as 1-hop neighbors in NT. But only confirmed nodes are chosen as 1-hop neighbors when constructing beacon frame. Based on the above definitions, the following parts specify different SRP mechanisms depending on different topology change cases.

2.2.6.1. Node join and BOP augmentation

We start this part with an example shown in Figure 2.12 (a). Table 2.1 gives partial information (e.g. without CFDS information) of node 1's NT in *working stage*. Supposing that it is the beginning of the network, all the ADCF nodes have the same high energy level 3. As node 1 and node 2 can get to each node of the network within 2 hops, so their neighbor densities are 8 (D_{max}). Node 3 can just get to node 1, 2, 6 and 7 within 2 hops, so its neighbor density is 5 and its NT is shorter. Obviously, node 1 has a maximum neighbor density and a smaller address than node 2, it is selected as initiator of the network. Initiator sends its beacon in the first beacon slot 0 of BOP and BOP length is defined as 8 (D_{max}). Then node 2 has the highest priority and it picks up the following beacon slot 1. Node 3 and node 5 are at distance of more than 2 hops, so they can reuse the same beacon slot 2. Each node operates BEP, ISP, SAP and BSAP as explained before. Therefore, the network superframe is organized as shown in Figure 2.13.

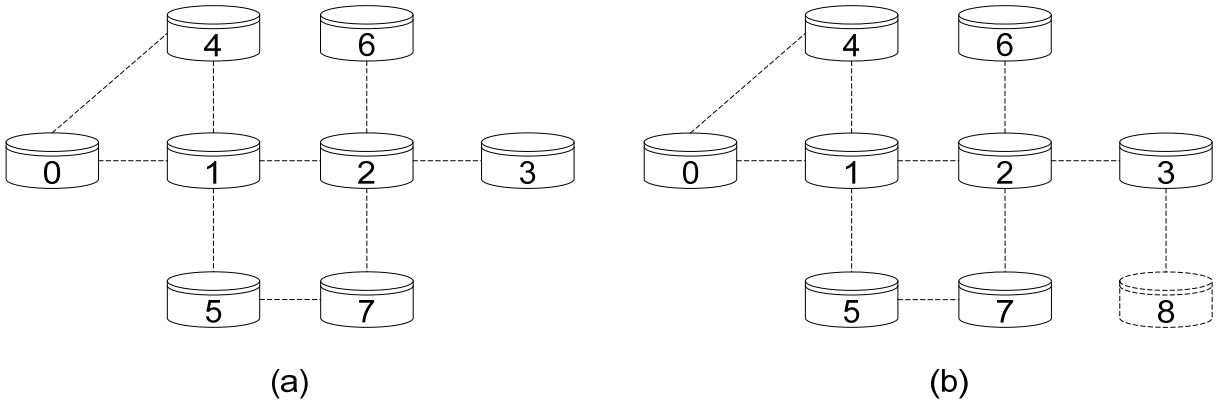

Figure 2.12 Topology example 1

Table 2.1 Node 1's partial NT in working stage

Neighbor Address	Neighbor Density	Neighbor Energy	Beacon Slot	Initiator Flag
0	5	3	3	0
1	8	3	0	1
2	8	3	1	0
3	5	3	2	0
4	5	3	4	0
5	6	3	2	0
6	5	3	4	0
7	6	3	3	0

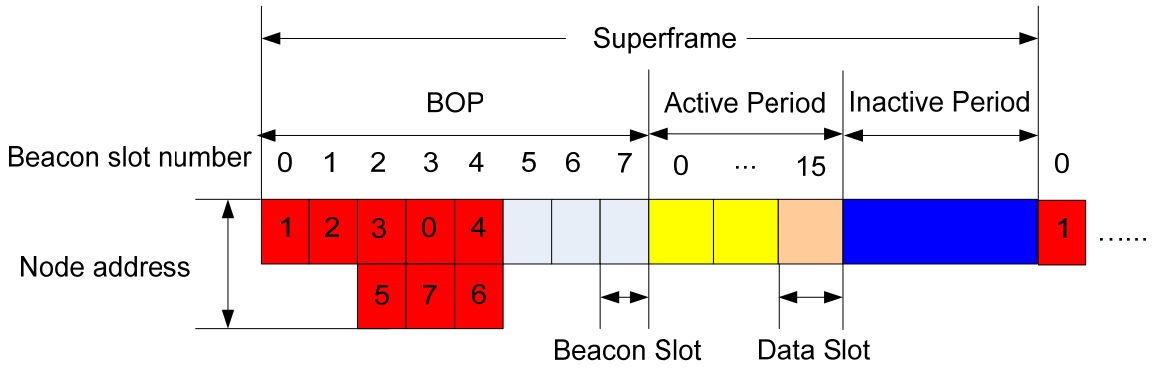

Figure 2.13 Superframe of network shown in Figure 12 (a)

Now let's discuss about node join. BOP length is defined as D_{max}. Obviously, it is enough and usually excess when there are slots reuse. For example in Figure 2.12 (b), node 8 wants to join the network. After listening period, it knows the BOP length with 3 free slots (slot 5, 6 and 7). Node 8 will choose the first free beacon slot 0 as it is a 3-hop neighbor of node 1. Therefore, node 8 can access the medium directly after listening and there remain 3 free beacon slots in BOP for other new nodes. Generally, neighbor density is a proper parameter to define BOP length [2.5] [2.6] as the network can work properly without disturbing by new node join.

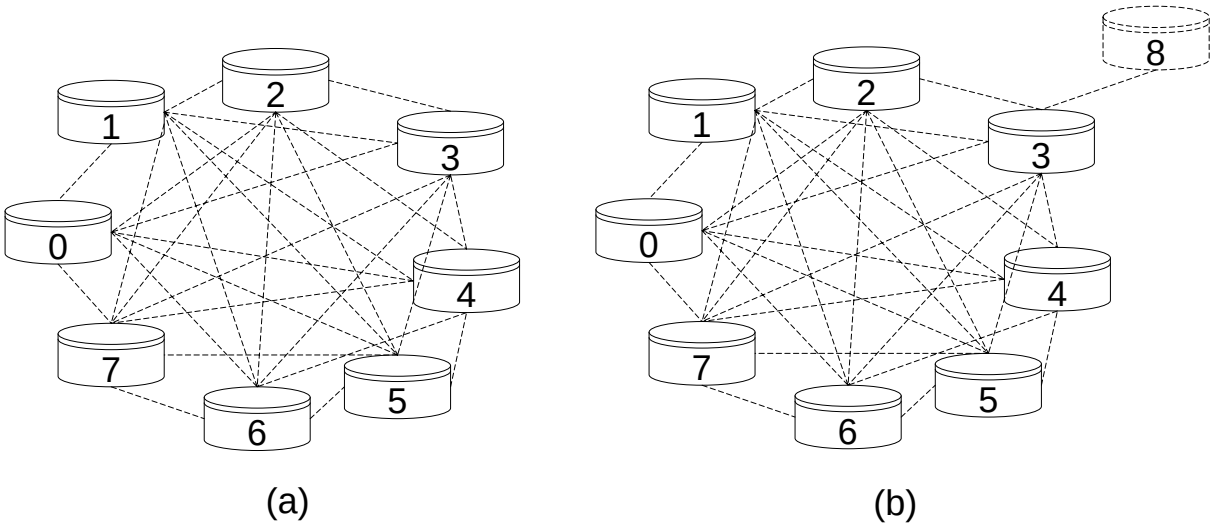

Figure 2.14 Topology example 2

One special case should be studied. There is no free beacon slot in a full mesh network as shown in Figure 2.14 ($D_{max} = 8$). If node 8 wants to join the network, it firstly sends its beacon in active period by CSMA/CA. Node 3 discovers node 8 as a 1-hop neighbor and broadcasts its beacon with a new neighbor density ($D_{max} = 9$). Other nodes should update this new D_{max}

and schedule BOP length as 9. At last, node 8 will choose the last beacon slot when a free beacon slot is shown in BOP. BOP augmentation is automatically organized.

2.2.6.2. Node failure and BOP reduction

Several cases exist in node failure. If a node failure is detected by beacon loss, neighbors simply delete this node from NT. If initiator fails, other nodes re-select an initiator but keep their BOP with the original slots. Therefore, the network could still work without disruption.

Only when the current D_{max} is less than half of the original BOP length, BOP reduction will be launched to improve the time efficiency. In this case, ADCF nodes return to *initialization stage* and restart from ISP.

2.2.6.3. Separation and integration of networks

In this part, we study network separation and network integration. Figure 2.15 shows a network with line topology.

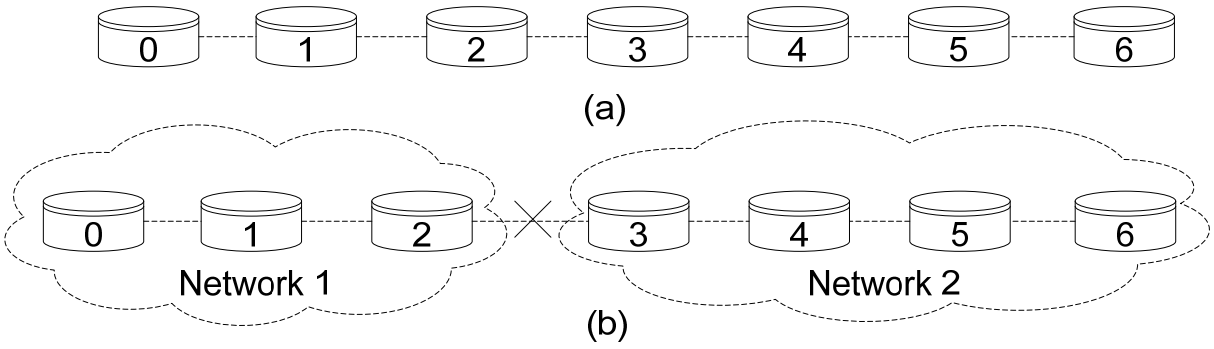

Figure 2.15 Topology example 3

From Figure 2.15 (a) to 2.15 (b), the original network ($D_{max} = 5$) is separated into network 1 ($D_{max1} = 3$) and network 2 ($D_{max2} = 4$) if the indicated link is broken by destruction or mobility. Obviously, network 1 keeps the original initiator node 2 as new initiator and original beacon slots. Therefore network 1 continues *working stage* with a BOP length as 5. On the other hand, when node 3 discovers the failure of initiator node 2, it deletes node 2's information from its NT and broadcasts in the following beacons. Other nodes in network 2 will receive this information gradually and return to *initialization stage*. A recovery procedure is therefore launched for network 2 and the nodes should start from ISP.

From Figure 2.15 (b) to 2.15 (a), when two networks meet, ADCF nodes return to *initialization stage* if the new D_{max} is larger than the two original lengths. BOP augmentation is invoked as explained in 2.2.6.1. But if the new D_{max} is not more than one network's BOP length, the initiator and beacon slots of the network will keep. Nodes in another network will find the corresponding free slots to insert BOP directly.

In part 2.2, a set of protocols/algorithms were expounded. BEP serves as cornerstone of ADCF. An initiator is selected by ISP in order to synchronize the network and schedule the superframe dynamically. Making use of wireless link characteristics, BSAP and DSAP enable each node to choose beacon and data slot in a distributed manner. Last but not least, SRP improve flexibility and robustness of ADCF. Thence all the functions explained in 2.1 could be achieved.

2.3.Service primitives

As explained in ADCF node architecture, each layer provides the services through the associated SAP. A service is specified by describing the service primitives and parameters that characterize it. A service may have one or more related primitives that constitute the activity that is related to that particular service. Each service primitive may have zero or more parameters that convey the information required to provide the service.

As in IEEE 802.15.4, a primitive can be one of four generic types, shown in Figure 2.16.

- Request: the request primitive is passed from the N-user to the N-layer to request that a service is initiated.
- Indication: the indication primitive is passed from the N-layer to the N-user to indicate an internal N-layer event that is significant to the N-user. This event may be logically related to a remote service request, or it may be caused by an N-layer internal event.
- Response: the response primitive is passed from the N-user to the N-layer to complete a procedure previously invoked by an indication primitive.
- Confirm: the confirm primitive is passed from the N-layer to the N-user to convey the results of one or more associated previous service requests.

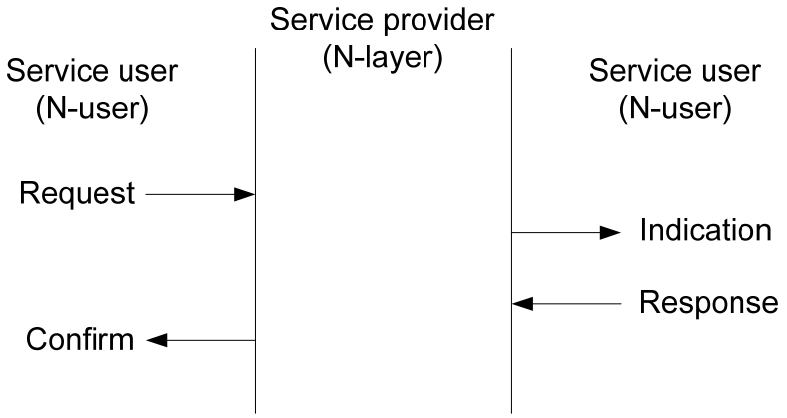

Figure 2.16 Service primitives

In ADCF, the PHY layer provides an interface between the MAC sublayer and the physical radio channel, via the RF firmware and RF hardware. PD-SAP supports the transport of MPDUs between peer MAC sublayer entities. PLME-SAP provides the layer management service interfaces through which layer management functions may be invoked. The PLME is also responsible for maintaining a database of managed objects pertaining to the PHY. This database is referred to as the PHY PAN Information Base (PIB°).

The MAC sublayer provides an interface between the upper layer and the PHY layer. MCPS-SAP supports the transport of upper layer data units between peer entities. MAC layer management functions may be invoked by ADCF-SAP. It is also responsible for maintaining the MAC sublayer database named ADCF PIB.

In addition, an Energy Entity (EE°) provides the services through the associated EE-SAP as shown in Figure 2.17. EE-SAP provides the information on the residual energy level of an ADCF node. This service allows sublayer to evaluate the current energy consumption and therefore achieve the protocol functions.

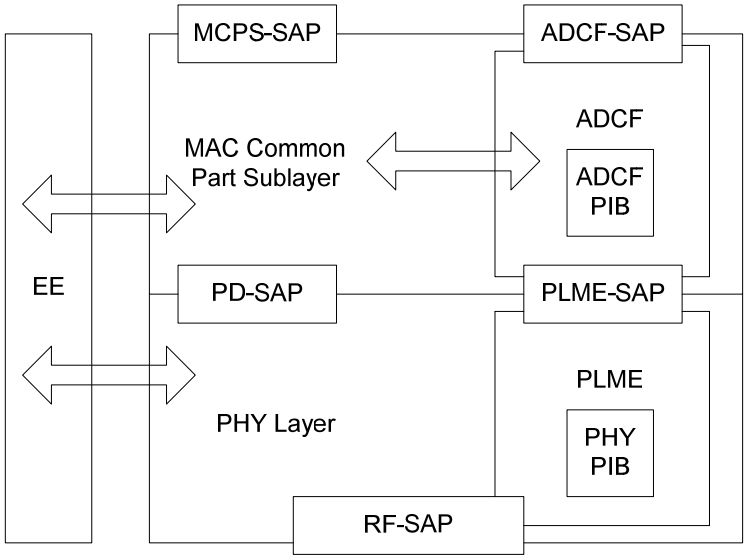

Figure 2.17 Sublayer reference model of ADCF node

The next parts gradually define the service specification of each layer by describing the required primitives and the related parameters.

2.3.1.PHY sublayer service specification

Table 2.2 shows the summary of the available primitives on PD-SAP which supports the transmission and reception of data by the PHY sublayer.

Table 2.2 PD-SAP primitives

	Request	Confirm	Indication	Response
PD-Data	√		√	

The PD-Data.Request primitive requests the transfer of an MPDU from the MAC sublayer to the local PHY entity. The semantics of the PD-Data.Request primitive is as follows: PD-Data.Request (psduLength, psdu).

- psduLength (1 octet): the number of octets contained in the PSDU to be transmitted by the PHY entity.
- psdu: the set of octets forming the PSDU to be transmitted by the PHY entity.

The PD-Data.Indication primitive indicates the transfer of an MPDU from the PHY to the local MAC sublayer entity. The semantics of the PD-Data.Indication primitive is as follows: PD-Data.Indication (psduLength, psdu, timestamp, ppduLinkQuality).

- psduLength (1 octet): the number of octets contained in the PSDU received by the PHY entity.
- psdu: the set of octets forming the PSDU received by the PHY entity.
- timestamp (3 octets): timestamp of the reception time.
- ppduLinkQuality (1 octet): LQI value measured during reception.

2.3.2.MAC sublayer service specification

2.3.2.1. MAC data service

The MCPS-SAP supports the transport of upper layer data units between peer entities. Table 2.3 lists the primitives provided by the MCPS-SAP.

Table 2.3 MCPS-SAP primitives

	Request	Confirm	Indication	Response
MCPS-Data	√	√	√	
MCPS-Purge	√	√		

The MCPS-Data.Request primitive requests the transfer of a MAC Service Data Unit (MSDU°) from a local upper layer entity to a single peer entity. The semantics of the MACP-Data.Request primitive are as follows: MCPS-Data.Request (msduLength, msdu, DstAddr, msduHandle, AckOptions, TxOptions).

- msduLength (1 octet): the number of octets contained in the MSDU to be transmitted by the MAC sublayer entity.
- msdu: the set of octets forming the MSDU to be transmitted by the MAC sublayer entity.
- DstAddr (2 octets): the device address of the entity to which the MSDU is being transferred.
- msduHandle (1 octet): the handle associated with the MSDU to be transmitted by the MAC sublayer entity.
- AckOptions (1 octet): acknowledged transmission or unacknowledged transmission of the MSDU.

- TxOptions (1 octet): the transmission mode of the MSDU. It can be transmitted by CSMA/CA or CFDS.

The MCPS-Data.Confirm primitive reports the results of a request to transfer a MSDU. The semantics of the MCPS-Data.Confirm primitive are as follows: MCPS-Data.Confirm (msduHandle, status).

- msduHandle (1 octet): the handle associated with the MSDU being confirmed.
- status (1 octet): the status of the last MSDU transmission. It is the same as IEEE 802.15.4 standard.

The MCPS-Data.Indication primitive indicates the transfer of a MSDU from the MAC sublayer to the local upper layer entity. The semantics of the MCPS-Data.Indication primitive are as follows: MCPS-Data.Indication (msduLength, msdu, SrcAddr, timestamp, mpduLinkQuality).

- msduLength (1 octet): the number of octets contained in the MSDU being indicated by the MAC sublayer entity.
- msdu: the set of octets forming the MSDU being indicated by the MAC sublyaer entity.
- SrcAddr (2 octets): the device address of the entity from which the MSDU was received.
- timestamp (3 octets): the time at which the data were received.
- mpduLinkQuality (1 octet): LQI value measured during reception of the MPDU.

The MCPS-Purge.Request primitive allows the next higher layer to purge an MSDU from the transaction queue. The semantics of the MCPS-Purge.Request primitive are as follows: MCPS-Purge.Request (msduHandle).

- msduHandle (1 octet): the handle of the MSDU to be purged from the transaction queue.

The MCPS-Purge.Confirm primitive allows the MAC sublayer to notify the next higher layer of the success of its request to purge an MSDU from the transaction queue. The

semantics of the MCPS-Purge.Confirm primitive are as follows: MCPS-Purge.Confirm (msduHandle, status).

- msduHandle (1 octet): the handle of the MSDU requested to be purge from the transaction queue.
- status (1 octet): the status of the request to be purged an MSDU.

2.3.2.2. ADCF management service

The ADCF-SAP allows the transport of management commands between the upper layer and the ADCF MAC layer. Table 2.4 summarizes the primitives supported through ADCF-SAP.

Table 2.4 ADCF-SAP primitives

	Request	Confirm	Indication	Response
ADCF-Get	√	√		
ADCF-Set	√	√		
ADCF-Start	√	√		
ADCF-CFDS	√	√	√	

The ADCF-Get.Request primitive requests information about a given PIB attribute. The semantics of the ADCF-Get.Request primitive are as follows: ADCF-Get.Request (PIBAttribute).

- PIBAttribute (1 octet): the identifier of the PIB attribute to read.

The ADCF-Get.Confirm primitive reports the results of an information request from the PIB. The semantics of the ADCF-Get.Confirm primitive are as follows: ADCF-Get.Confirm (status, PIBAttribute, PIBAttributeValue).

- status (1 octet): the result of the requested PIB attribute information.
- PIBAttribute (1 octet): the identifier of the PIB attribute that was read.
- PIBAttributeValue (1 octet): the value of the indicated PIB attribute.

The ADCF-Set.Request primitive attempts to write the given value to the indicated PIB attribute. The semantics of the ADCF-Set.Request primitive are as follows: ADCF-Set.Request (PIBAttribute, PIBAttributeValue).

- PIBAttribute (1 octet): the identifier of the PIB attribute to write.
- PIBAttributeValue (1 octet): the value to write to the indicated PIB attribute.

The ADCF-Set.Confirm primitive reports the results of an attempt to write a value to a PIB attribute. The semantics of the ADCF-Set.Confirm primitive are as follows: ADCF-Set.Confirm (status, PIBAttribute).

- status (1 octet): the result of the request to write the PIB attribute.
- PIBAttribute (1 octet): the identifier of the PIB attribute that was written.

The ADCF-Start.Request primitive allows the ADCF node to initiate a given superframe configuration. The semantics of the ADCF-Start.Request primitive are as follows: ADCF-Start.Request (BO, SO). Both BO and SO occupy 1 octet.

The ADCF-Start.Confirm primitive reports the result of the attempt to start using a given superframe configuration. The semantics of the ADCF-Start.Confirm primitive are as follows: ADCF-Start.Confirm (status). This 1 octet parameter may be set as SUCCESS or INVALID_PARAMETER.

The ADCF-CFDS.Request primitive allows a node to send a request to its neighbor to allocate a CFDS or to deallocate an existing CFDS. The semantics of the ADCF-CFDS.Request primitive are as follows: ADCF-CFDS.Request (DstAddr, CFDSLength, CFDSDuration).

- DstAddr (2 octets): the device address to which the node requests CFDS.
- CFDSLength (1 octet): the number of CFDSs.
- CFDSDuration (1 octet): the duration of the requested CFDSs.

The ADCF-CFDS.Confirm primitive reports the results of the ADCF-CFDS.Request primitive. The semantics of the ADCF-CFDS.Confirm primitive are as follows: ADCF-CFDS.Confirm (status, CFDSNumber, CFDSDuration).

- status (1 octet): the status of the CFDS request.
- CFDSNumber (1 octet): the slot number of the first allocated CFDS.
- CFDSDuration (1 octet): the duration of the allocated CFDSs.

The ADCF-CFDS.Indication primitive indicates that one or several CFDSs have been allocated. The semantics of the ADCF-CFDS.Indication primitive are as follows: ADCF-CFDS.Indication (SrcAddr, CFDSNumber, CFDSDuration).

- SrcAddr (2 octets): the device address from which CFDS was requested.
- CFDSNumber (1 octet): the slot number of the first allocated CFDS.
- CFDSDuration (1 octet): the duration of the allocated CFDSs.

2.3.3.Hardware service specification

At last, EE-SAP has two primitives, EE-ED.Request and EE-ED.Confirm as shown in Table 2.5, to get the energy information from hardware.

Table 2.5 EE-SAP primitives

	Request	Confirm	Indication	Response
EE-ED	√	√		

The EE-ED.Request primitive requests energy information from one sublayer to hardware. The semantics of the EE-ED.Request primitive are as follows: EE-ED.Request (EnergyAttribute).

- EnergyAttribute (1 octet): the identifier of the energy attribute.

The EE-ED.Confirm primitive reports the results of the EE-ED.Request primitive. The semantics of the EE-ED.Confirm primitive are as follows: EE-ED.Confirm (status, EnergyAttributeValue).

- status (1 octet): the sublayer successfully get the energy attribute or not.
- EnergyAttribute (1 octet): the available energy value of the node. Thence, Neighbor Energy (NE°, 2 bits) in the beacon frame could be coded by this energy value and the previously fixed energy threshold.

Finally, the primitives and the associated parameters of PHY, MAC and hardware entity were presented in part 2.3. These primitives provide critical services between different entities. Thence, an ADCF node can work thanks to the protocol of each layer and the service interfaces between layers.

3. Conclusion

As illustrated in the chapter 1, our application requires an adaptive communication protocol providing QoS-guaranteed service with reasonable energy consumption in a mesh network. The current technologies and associated protocols can not solve the problems together such as beacon collision, dynamic timeslot allocation, energy saving on router nodes, determinism on medium access, etc. Therefore we proposed an original ADCF protocol and presented it in this chapter.

ADCF based on the IEEE 802.15.4 2.4 GHz DSSS physical layer and classical superframe structure was designed for enabling the mesh topology. So the descriptions such as ADCF node architecture and network formation were firstly given in this chapter.

Afterwards, by describing superframe structure and beacon frame format, we specified the contributions of ADCF. Two mechanisms, CFBS and CFDS were fully explained. CFBS enables the nodes far away than 2-hop to reuse the timeslots so that beacon collisions could be avoided. The nodes could join or leave the network freely as BOP dynamically changes according to topological changes. Thanks to CFBS, CFDS enables the nodes to negotiate collision-free data slots in the mesh topology. The wireless medium is dedicated to the nodes that use CFDS to transmit application traffic within a bounded time. In addition, ADCF enables all nodes, including routers, to sleep for energy saving. In order to achieve these protocol functions, ADCF is divided into a set of protocols/algorithms and each of them was fully explained in this chapter.

At last, service primitives and related parameters used in the ADCF node were illustrated. These primitives connect different layers and provide important services. So a complete ADCF node architecture was described and the corresponding functions could be achieved. An efficient multi-hop mesh network is built and maintained with these ADCF nodes.

In the next two chapters, simulation and prototype implementations will be presented to evaluate the performance of ADCF.

Reference

[2.1] Juan Lu, A. van den Bossche, E. Campo, “An Adaptive and Distributed Collision-Free MAC Protocol for Wireless Personal Area Networks”, 6th International Symposium on Intelligent Systems Techniques for Ad hoc and Wireless Sensor Networks (IST-AWSN 11), Volume 5, pp. 798-803, September 2011

[2.2] Juan Lu, A. van den Bossche, E. Campo, “Enabling Guaranteed Beacon and Data Slots in Multi-hop Mesh Sensor Networks for Home Health Monitoring”, 8th International Conference on Wireless and Mobile Communications (ICWMC 12), pp. 98-102, June 2012

[2.3] S. Mahfoudh, P. Minet, “Maximization of Energy Efficiency in Wireless Ad hoc and Sensor Networks with SERENA”, Mobile Information Systems, Advances in Wireless Networks, Volume 5 Issue 1, pp. 33-52, April 2009

[2.4] A. van den Bossche, T. Val, E. Campo, “Prototyping and performance analysis of a QoS MAC layer for industrial wireless network”, 7th International Conference on Fieldbuses and nETworks in industrial and embedded systems (IFAC 07), Volume 7 Part 1, November 2007

[2.5] P. Minet, S. Mahfoudh, “SERENA: SchEduling RoutEr Nodes Activity in wireless ad hoc and sensor networks”, 4th International Wireless Communications and Mobile Computing Conference (IWCMC 08), pp. 511-516, August 2008

[2.6] S. Mahfoudh, P. Minet, “Performance evaluation of the SERENA algorithm to SchEdule RoutEr Nodes Activity in wireless ad hoc and sensor networks”, 22nd International Conference on Advanced Information Networking and Applications (AINA 08), pp. 287-294, March 2008

Chapter 3

Simulation Study

The simulation work is studied in this chapter, to evaluate the scope of our contribution. Firstly, several simulation tools are investigated and we choose OPNET as our simulator. After that, a simulation model which implements ADCF MAC is illustrated. Many experimental scenarios were simulated. The results show the performance of ADCF, such as protocol cost, QoS capability, energy consumption, etc.

1. WSN simulation tools

Running real experiments on a testbed is costly and difficult, so simulation is often needed in the network design phase before actual implementation. There are many different simulation tools for WSN, such as NS-2, TOSSIM, OMNeT++, OPNET, GloMoSim, UWSim, Avrora, SENS, COOJA, Castalia, Shawn, EmStar, J-Sim, SENSE, etc. Papers [3.1] [3.2] [3.3] provide comprehensive survey and comparisons of these popular simulators and may help user to choose the most suitable one.

As explained in chapter 2, ADCF is based on IEEE 802.15.4 standard. In order to be comparable and simplify the simulation work, only 4 main-stream simulators with 802.15.4 MAC protocol are studied in this chapter.

1.1.NS-2

1.1.1.Overview

NS-2 is a discrete event simulator targeted at networking research, both wired and wireless area. It has evolved substantially over the past few years with a lot of users. NS-2 is open source and provides online documents [3.4]. People can run this simulator on Linux or on Cygwin. In addition, NS-2 is built with combination of C++ and OTcl.

1.1.2.Merits and limitations

To the merits, firstly as a non-specific network simulator, NS-2 can support a considerable range of protocols in all layers, including the 802.15.4 MAC protocol. Secondly, the open source model saves the cost of simulation, and online documents allow users to modify and improve the codes.

However, GTS mechanism is not implemented in NS-2. In addition, the codes are relatively difficult to understand. The users have to directly face to text commands as its poor graphical support.

1.2.TOSSIM

1.2.1.Overview

TOSSIM is an emulator specifically designed for WSN running on TinyOS [3.5]. It is a discrete event network emulator built in C++ and Python. TOSSIM captures the behavior and interactions of network not on the packet level but at network bit granularity. People can run this emulator on Linux or on Cygwin. TOSSIM also provides open sources and online documents.

1.2.2.Merits and limitations

TOSSIM is a simple but powerful emulator. It can provide more precise simulation results at component levels because of compiling directly to native codes. TOSSIM has a GUI and can support thousands of nodes simulation. The protocol library contains 802.15.4 MAC.

On the other hand, TOSSIM is specifically designed for TinyOS applications, motes-like nodes are the only thing that it can simulate. Secondly, TOSSIM's run-instantly execution model does not capture CPU time. Since interrupts are discrete events, TOSSIM follows the FIFO run-to-completion model and does not model preemption and the resulting possible data races with different priorities. Compilation steps lose the fine-grained timing and interrupt properties of the code, which can be important when the application runs on the hardware and interacts with other nodes. Moreover, like NS-2, GTS mechanism is not implemented, which disables future comparisons between 802.15.4 GTS and CFDS.

1.3.OMNeT++

1.3.1.Overview

OMNeT++ is a discrete event network simulator built in C++ [3.6]. OMNeT++ provides both a noncommercial license, used at academic institutions or no-profit research organizations, and a commercial license, used at for-profit environments. This simulator supports module programming model. Users can run OMNeT++ on Linux, Unix-like system and Windows. OMNeT++ is a popular non-specific network simulator, which can be used in both wired and wireless area. Most of frameworks and simulation models are open sources.

1.3.2.Merits and limitations

OMNeT++ provides a user-friendly GUI which makes the tracing and debugging much easier than using other simulators. This simulator can support MAC protocols as well as some localized protocols in WSN.

However, 802.15.4 MAC protocol is not fully implemented. Specifically, GTS reservation and allocation are missing. In addition, the compatible problem will rise since individual researching groups developed the models separately, this makes the combination of models difficult and programs may have high probability report bugs.

1.4.OPNET

1.4.1.Overview

OPNET [3.7] is very large and powerful simulator with wide variety of possibilities. It enables to simulate entire heterogeneous networks with various protocols. OPNET is expensive for commercial usage but fortunately there are free licenses for educational purposes. This simulator is constructed from C and C++ source code blocks with a huge library of OPNET specific functions. We can run OPNET on Linux or Windows system.

1.4.2.Merits and limitations

Like all the simulation tools, OPNET contains both merits and limitations. We finally choose OPNET due to the following reasons: Firstly, high-quality programming of OPNET makes the codes simple and clear. Secondly, we can modify and improve the codes to our ADCF MAC as they are all open source. Thirdly, this simulator provides potent capabilities in GUI, data collection as well as data analysis. In fact, the most important reason is the 802.15.4 MAC implementation. The interesting features of this 802.15.4 MAC implementation is further described in the next part.

1.4.3.IEEE 802.15.4 MAC implementation

To the best of our knowledge, there are 2 versions of 802.15.4 MAC in OPNET.

1.4.3.1. WPAN version

802.15.4 WPAN version is provided in OPNET 11.5A. There are two types of nodes in the network. WPAN analyzer node captures global statistical data from the whole PAN. WPAN sensor node includes coordinator and end device.

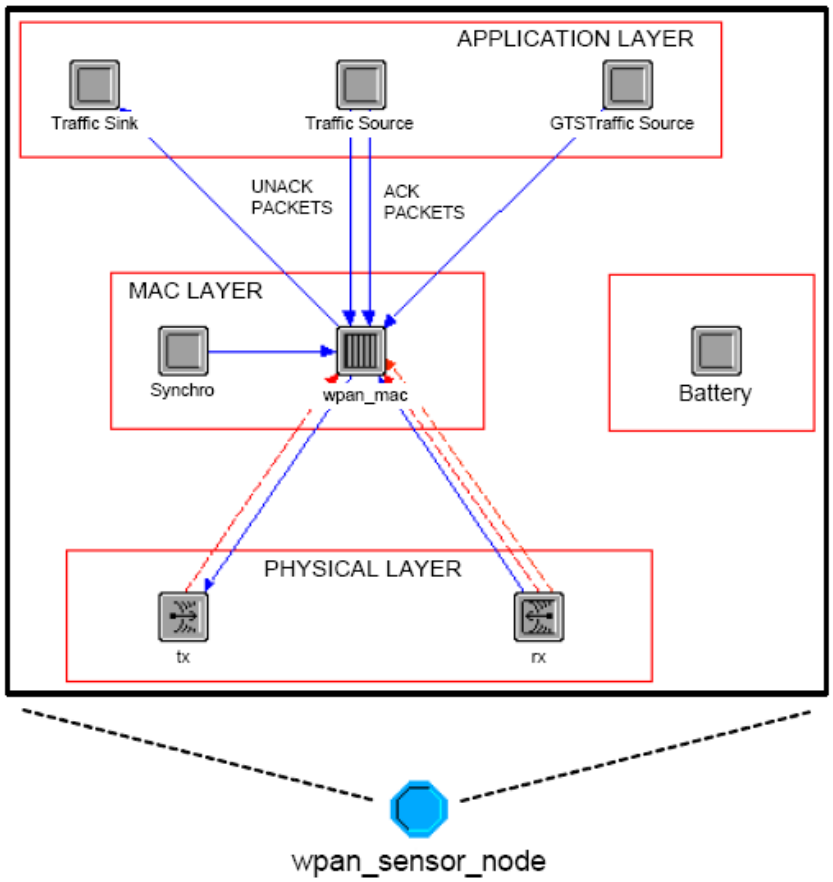

Figure 3.1 WPAN sensor node model under OPNET simulator

The following features are supported by WPAN sensor node, as shown in Figure 3.1.

- It supports star topology.
- Only beacon-enabled mode is implemented.
- There are 802.15.4 standardized frame formats, including beacon, command, acknowledge and MAC packet.
- Physical layer characteristics are consistent with the standard.
- It provides Slotted CSMA/CA MAC protocol.

- It also provides GTS mechanism, including GTS allocation, de-allocation and re-allocation functions.
- The sensor node can generate acknowledged and unacknowledged application data which is transmitted during CAP.
- The sensor node can generate acknowledged and unacknowledged application data which is transmitted during CFP.
- Battery module can compute the power consumption of each sensor node. Two motes, MICAz and TelosB, are supported. Figure 3.2 shows the specific parameters.

Attribute Name	Value	Description
Receive Mode	[mA]	The current draw of the device when the transceiver is in the receiving mode (RX_ON state). There are 2 predefined options default for MICAz and TelosB motes: • **MICAz** = 27.7 mA • **TelosB** = 24.8 mA
Transmission Mode	[mA]	The current draw of the device when the transceiver is in the transmitting mode (TX_ON state). There are 6 predefined options corresponding to the nominal transmitting power of the transceiver: • **MICAz (0 dBm) = TelosB (0 dBm)** = 17.4 mA • **MICAz (-5 dBm) = TelosB (-5 dBm)** = 14 mA • **MICAz (-10 dBm) = TelosB (-10 dBm)** = 11 mA
Idle Mode	[μA]	The current draw of the device when the transceiver is in idle mode and voltage regulator is on. There are 2 predefined options default for MICAz and TelosB motes: • **MICAz** = 35 μA • **TelosB** = 26.1 μA
Sleep Mode	[μA]	The current draw of the device when the transceiver is inactive and voltage regulator is off. There are 2 predefined options default for MICAz and TelosB motes: • **MICAz** = 16 μA • **TelosB** = 6.1 μA

Figure 3.2 Battery module parameters [3.7]

Actually, as seen from Figure 3.2, only the consumption of transceiver is considered with this battery module.

1.4.3.2. ZigBee version

802.15.4 ZigBee version [3.8] [3.9] is based on WPAN version. Some new features are added in OPNET 15.0.

- Cluster-tree topology is supported in this new version.

- Network layer is implemented with ZigBee hierarchical tree routing.
- Verification of nodes' addresses corresponds to the cluster-tree addressing scheme.

In conclusion, we decided to choose OPNET as our simulation platform after investigating the 802.15.4 implementation situations. ZigBee version is adapted in our simulation for comparing the performance of ADCF.

2. ADCF simulation model

How to build ADCF simulation model is explained in this part. As the hierarchical structure of OPNET, modeling is divided to 3 main domains: network domain, node domain and process domain.

2.1.Network domain

Network domain is responsible for geographical coordinates, network topology and mobility, etc. As required by the application, we choose a piece of land of 100 m * 100 m, as shown in Figure 3.3.

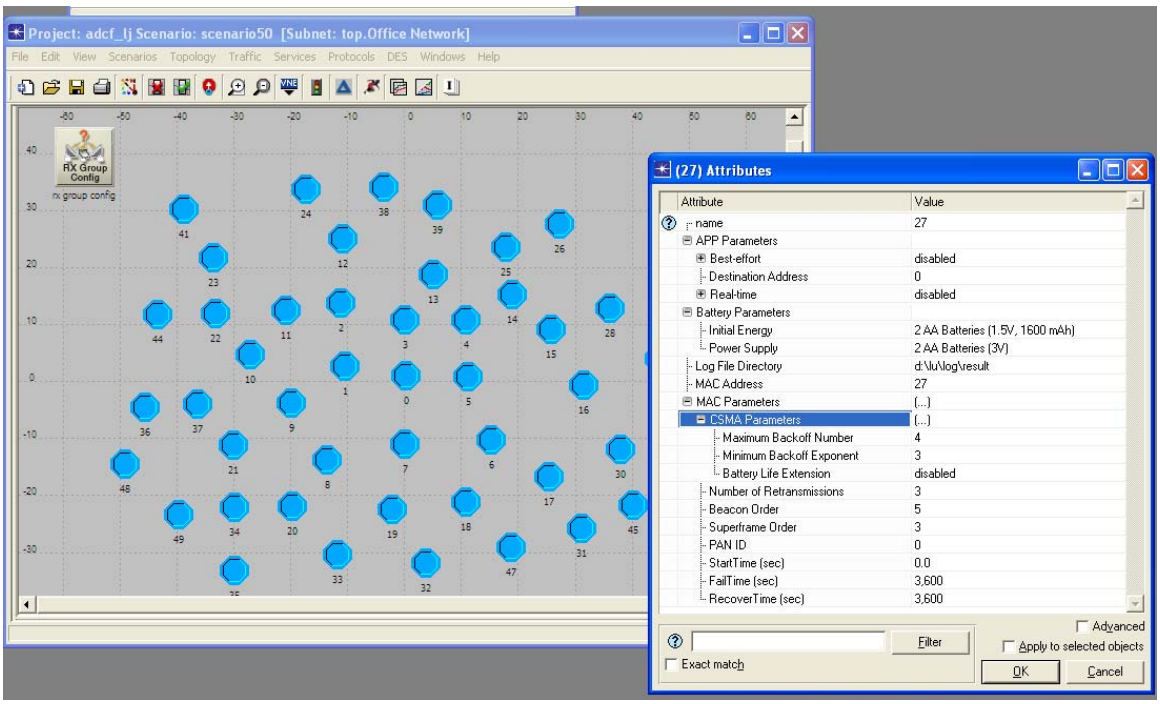

Figure 3.3 ADCF network domain

ADCF nodes are randomly deployed in the network. A device named RX Group Config which does not participate in the network communication can strictly limit the transmission range of each node as 15 meters. Therefore, multi-hop networks (up to 9 hops) could be achieved in this zone of 100 m * 100 m, for investigation of interesting network scenarios in the house or even buildings. The max hop count 9 is obtained on the observations and is not a theoretical bound. Also, we can create and configure many attributes in the network domain, such as MAC address, battery parameters, start time, etc.

2.2.ADCF node domain

In the node domain, Figure 3.4 shows the node architecture of ADCF. Each ADCF node has a wireless transceiver, a MAC layer and an application layer. Also, a battery module is built to calculate the energy consumption of each node.

Even though we haven't network layer, a static routing mechanism achieved by prior manually adding routes was directly implemented in the MAC layer, in order to simulate application traffics over the network and compare the performances with 802.15.4 MAC protocol, without being disturbed by a routing protocol.

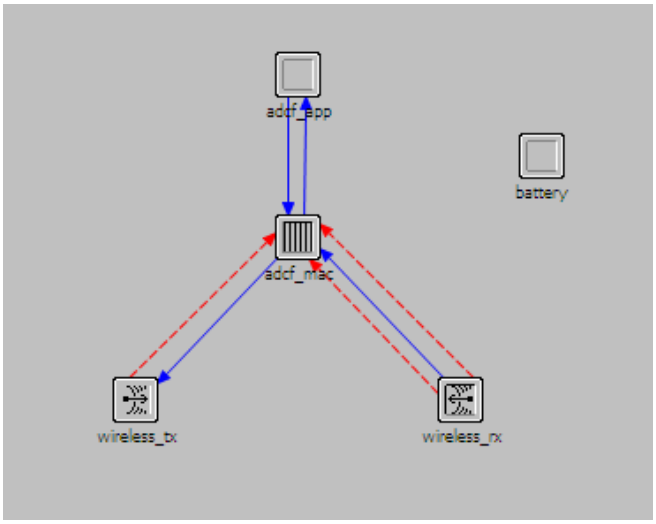

Figure 3.4 ADCF node domain

2.3.Process domain

The basic building blocks of node domain are modules, which include processors, queues, transceivers and generators. Processors are the primary general purpose building blocks and are fully programmable. The following parts illustrate ADCF modules one by one.

2.3.1.PHY layer module

The physical layer module consists of a wireless rx for reception and a wireless tx for transmission. They are compliant to the IEEE 802.15.4 specification operating at the 2.4 GHz frequency range, where each channel has a bandwidth of 2 MHz. The transmission power is set to 1 mW and the modulation scheme is OQPSK.

We use the default wireless models of OPNET library for simulating the background noise, propagation delay, radio interferences, received power, bit error rate, etc. In case of collisions, the reception result depends on the number of collided frames, received power and bit error threshold computed in the default receiver pipelines of the OPNET library.

In fact, it is an ideal physical layer module by default. We use this module for two reasons: Firstly, a suitable wireless link module cannot be found in the OPNET library. We will further study the impact of an imperfect physical layer to ADCF in prototype of chapter 4. Secondly, this module is consistent with 802.15.4 implementation, in order to compare the MAC performances with the same PHY conditions.

2.3.2.MAC layer module

Figure 3.5 shows the state transitions at MAC layer. The initial state is the place where execution begins in this process. The green state is a forced state which does not allow a pause during the process. The red one is an unforced state which allows the pause. So transitions describe the possible movement of the process from state to state and the conditions allowing such a change.

Our MAC layer implements all the ADCF proposals: BEP, ISP, SPA, BSAP, DSAP and SRP. In other words, the following features are achieved in the implementation:

- It provides slotted and unslotted CSMA/CA mechanisms,
- The superframe which contains BOP, active period and inactive period can be organized with the corresponding parameters,
- It provides CFDS mechanism. Each node can reserve dedicated data slots by beacon exchanges,
- An adaptive BOP is achieved, including BOP augmentation and BOP reduction.

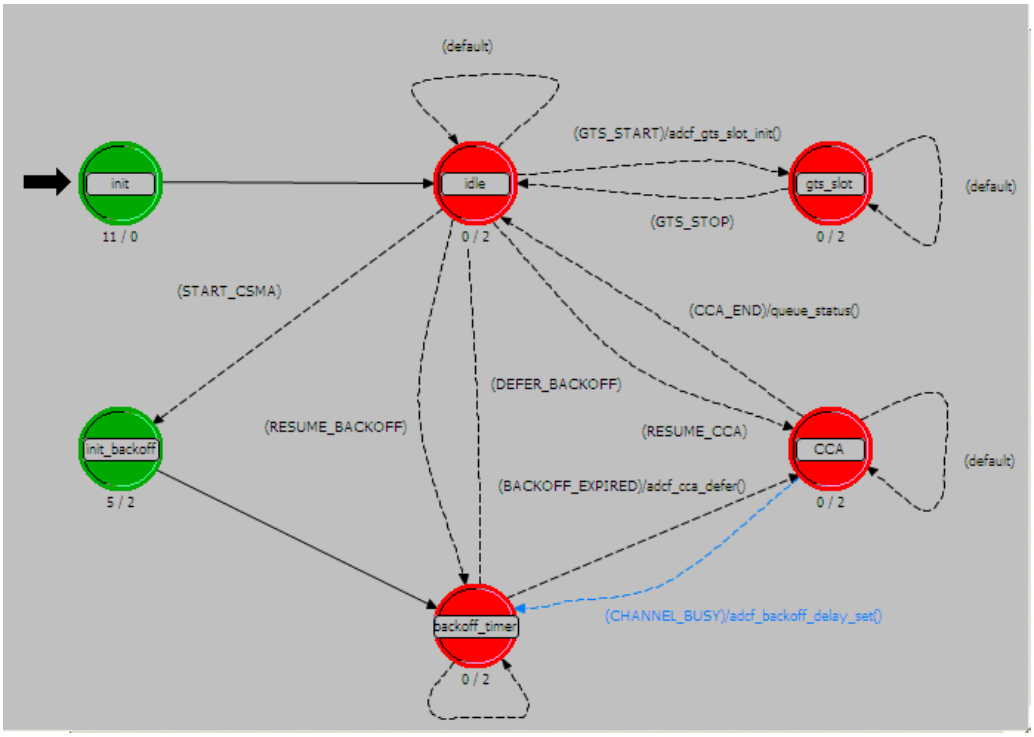

Figure 3.5 MAC layer state transition diagram

2.3.3.APP layer module

As shown in Figure 3.6, the application layer consists of two generators: best-effort traffic generator and real-time traffic generator. Traffic distribution can be constant, with an application payload of 100 bits. The application traffic should include destination address and traffic type and they are all with acknowledgements.

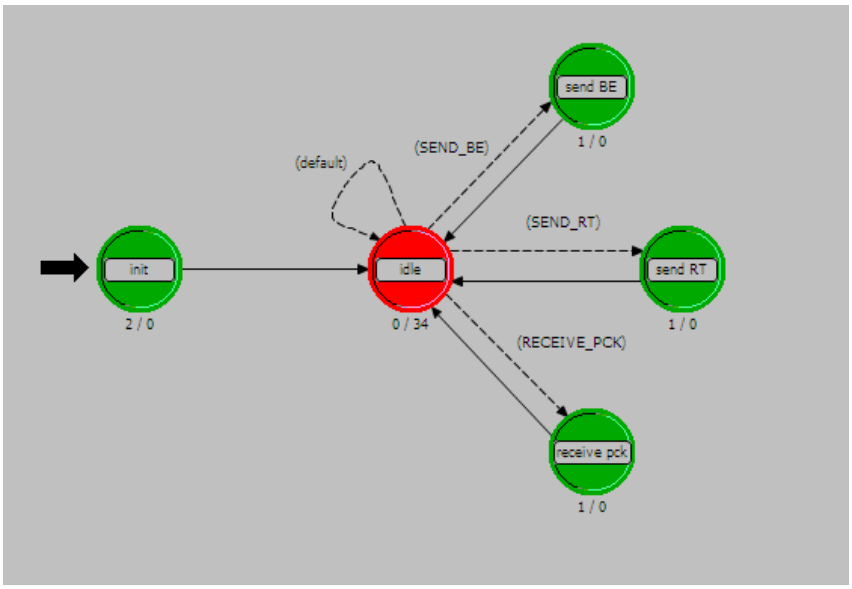

Figure 3.6 APP layer state transition diagram

2.3.4.Battery module

The battery module computes the consumed and remaining energy levels. As the included battery module of OPNET does not consider MCU energy consumption, a more realistic energy model [3.10] is used in the simulation. It includes all components of the node, both MCU and transceiver. Table 3.1 shows the current consumption in each state.

Table 3.1 Current consumption for MCU and transceiver

State	MCU	Transceiver
Tx (0 dBm)	3.5 mA	30 mA
Tx (3.6 dBm)	3.5 mA	37 mA
Rx	3.5 mA	38 mA
Idle	5 mA	1.3 mA
Sleep	140 μA	

In fact, this energy model is based on practical measurement of our sensor board considered in the prototype. Therefore it has been used in simulation for both ADCF and IEEE 802.15.4.

2.4.Basic simulation parameters

Table 3.2 summarizes the basic and important simulation parameters which will be used in all our experiments presented in the following of this chapter.

Table 3.2 Basic simulation parameters

Parameter	Value
Scene area	100 m *100 m
Transmission range	15 m
BO	7
SO	4
Application payload	100 bits
CSMA buffer	0.5 k octets
CFDS buffer	1.5 k octets
Simulation duration	30 min
Simulation times	20

As fixed transmission range, in each scenario the positions of nodes do not be changed except we want to adjust the network density. With the same number of nodes, the network

density is adjusted by increasing or reducing the distance between nodes. In other words, the hop count of a network is certainly changed in this case.

Also, both CSMA buffer and CFDS buffer are parametrized with realistic size of prototype.

Each simulation lasts 30 minutes and each value is the average of 20 simulations.

3. Experimental scenarios and simulation results

Several experiments are studied in this part. The first one aims to evaluate the cost of ADCF. The second experiment focuses on investigation of QoS capability. Node failure and recovery cases are studied in the third experiment. Then we compare ADCF with IEEE 802.15.4 in the same simulation conditions. The last experiment presents the performances of ADCF with large scale and high neighbor density.

3.1. Protocol cost

As ADCF is a distributed protocol, it is essential to investigate its cost: *Convergence Time* and *Message Overhead*. They are decided by three parameters of a network: node number (N), maximum hop count (H_{max}) and beacon interval (T_{cycle}).

3.1.1. Convergence time

Convergence Time indicates the time from a topology change to a valid working stage. Here we evaluate the time consumption of organizing a synchronized mesh network by ADCF. Nodes start working from 0 s to 6 s gradually. Therefore the time from the beginning of the first node to the synchronized state of the last node is convergence time and studied as follows.

As shown in Figure 3.7, when we set N as 30 and T_{cycle} as 1.5 s, *Convergence Time* is larger with the increase of H_{max}. That is because more time is needed for propagation of initiator's information and waiting priority in the multi-hop network. The average convergence time of 30 nodes in 7 hops is about 25 s and the variance is less than 2.46 s.

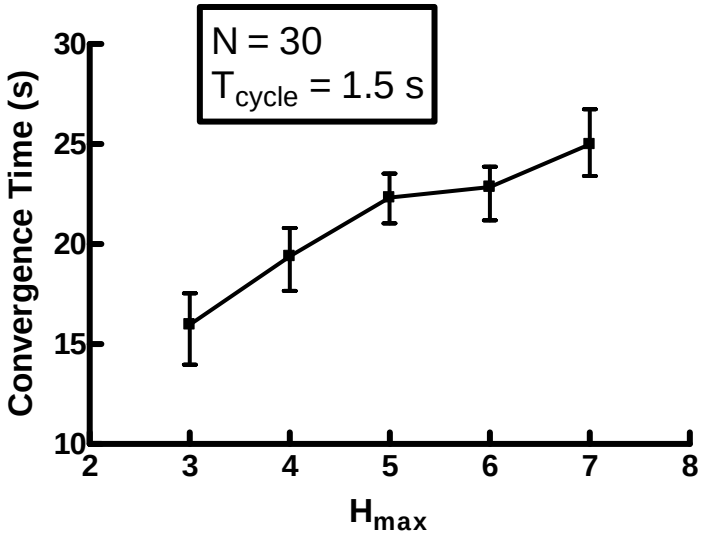

Figure 3.7 Convergence time vs. Hmax

We configure T_{cycle} from 0.5 s to 2.5 s. The nodes exchange their beacons less frequently when T_{cycle} is higher. Therefore *Convergence Time* increases gradually, as shown in Figure 3.8. When T_{cycle} equals 0.5 s, only 5 s could allow 30 nodes to be synchronized and sent their beacons without collision.

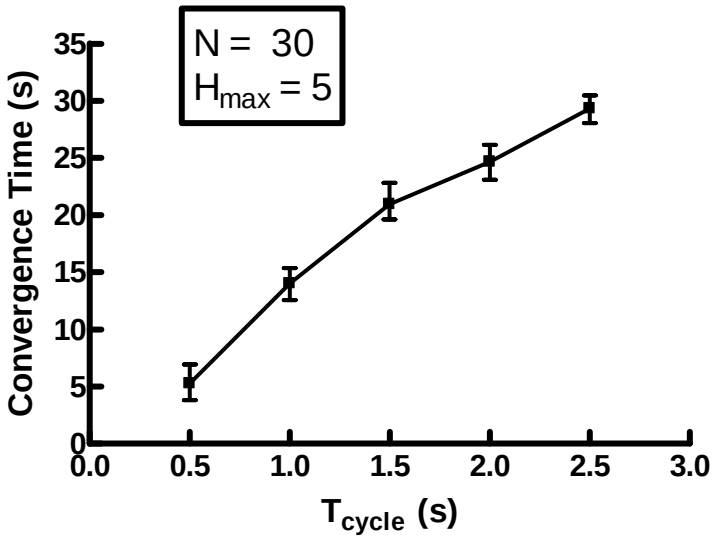

Figure 3.8 Convergence time vs. Tcycle

Obviously, it takes longer *Convergence Time* for a larger scale network. The average convergence time for 50 nodes is approximate 26 s, as shown in Figure 3.9.

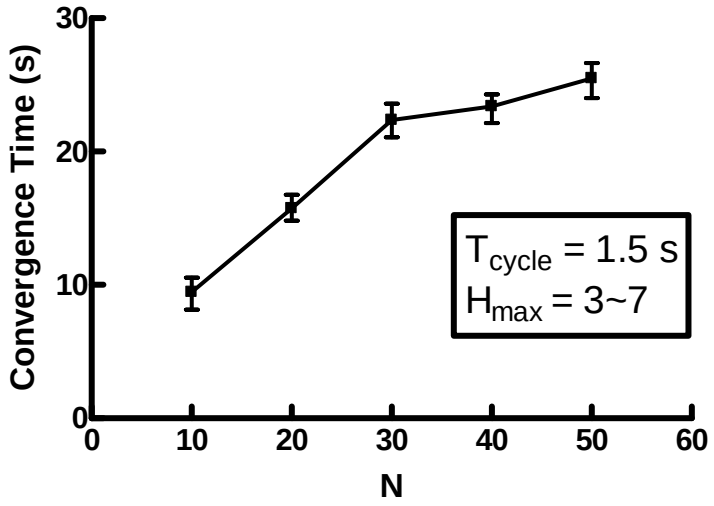

Figure 3.9 Convergence time vs. N

3.1.2.Message overhead

Message Overhead here indicates the messages exchanged between nodes from a topology change to a valid working stage. Here we evaluate the transmitted beacons in the *Convergence Time*. As illustrated in part 3.1.1, *Convergence Time* increases with the augmentation of H_{max} and N. Consequently, more beacons are sent with the augmentation of H_{max} and N. When T_{cycle} increases, *Convergence Time* also increases but the nodes send their beacons more slowly. So we merely discuss the relationship between *Message Overhead* and T_{cycle} as an example.

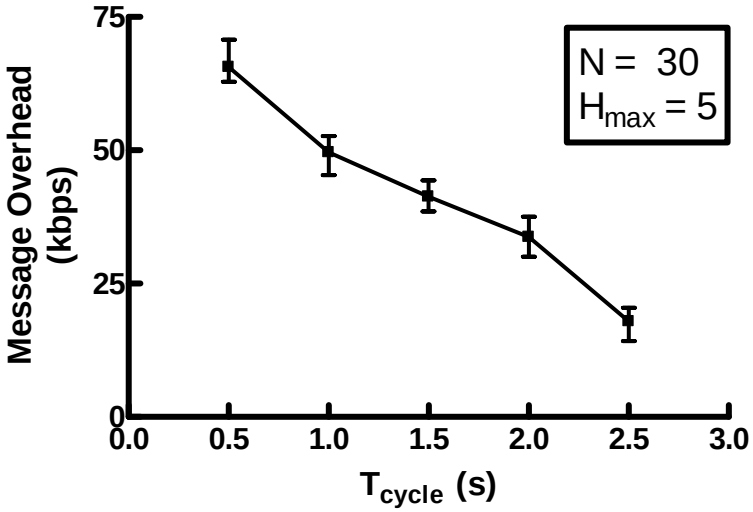

Figure 3.10 Message overhead vs. Tcycle

As shown in Figure 3.10, even though *Convergence Time* is larger with the increase of T_{cycle}, fewer beacons are delivered for organizing a valid working stage network. When T_{cycle} equals to 0.5 s, *Message Overhead* is about 63.225 kbps so much less than data rate of 250 kbps supported by the IEEE 802.15.4 physical layer. That means we can make trade-offs between *Convergence Time* and *Message Overhead*.

In conclusion, we study the cost of building a collision-free mesh network in this part. The worst case in the working stage is to rebuild the network, so the maintenance cost could be considered by this study too. From the simulation results, we can see that *Convergence Time* is rational, for example less than 30 s for a network of 50 nodes. In the application of home monitoring, a working stage usually lasts several months. Compared with this long-term organized network, the cost of *Convergence Time* is worth and acceptable. Also, *Convergence Time* can be further reduced at the price of a higher *Message Overhead*.

3.2.QoS capability

In this experiment, there are 30 nodes in the network. 7 of them have best-effort traffic and another 7 nodes have real-time traffic. Other nodes only exchange beacons and have no application traffic to send. Best-effort traffic is sent by CSMA/CA. Real-time traffic demands CFDS and is sent in the collision-free data slot. Two QoS metrics, *End-to-End Delay* and *Packet Success Ratio*, are simulated and analyzed.

3.2.1.End-to-end delay

On the source node, MAC layer receives a packet from application layer at the time *t1*. Finally, on the destination node, MAC layer receives this packet from its physical layer at the time *t2*. The difference between *t2* and *t1* is defined as *End-to-End Delay* in our work. It should be noted that, in the application layer, traffics are sent randomly so that the packets can fall on anywhere of superframe when arriving at MAC layer.

Figure 3.11 shows *End-to-End Delay* of best-effort traffic. When traffic is too heavy, indicating *Packet Interarrival Time* equals 0.1, CSMA buffer overflows and so *End-to-End Delay* is higher. When CSMA buffer is available, the average *End-to-End Delay* is about 0.91 s and slightly decreases with the augmentation of *Packet Interarrival Time*. The variance drifts from 0.14 s to 1.72 s, depending on random generation time of the packets.

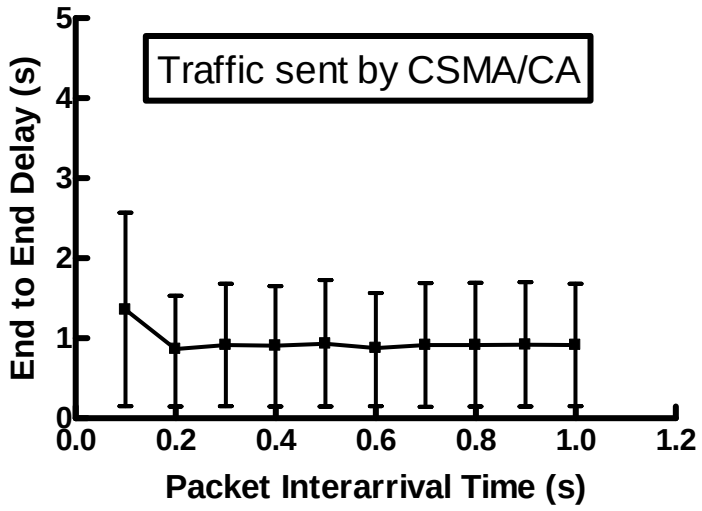

Figure 3.11 End-to-end delay for traffic sent by CSMA/CA

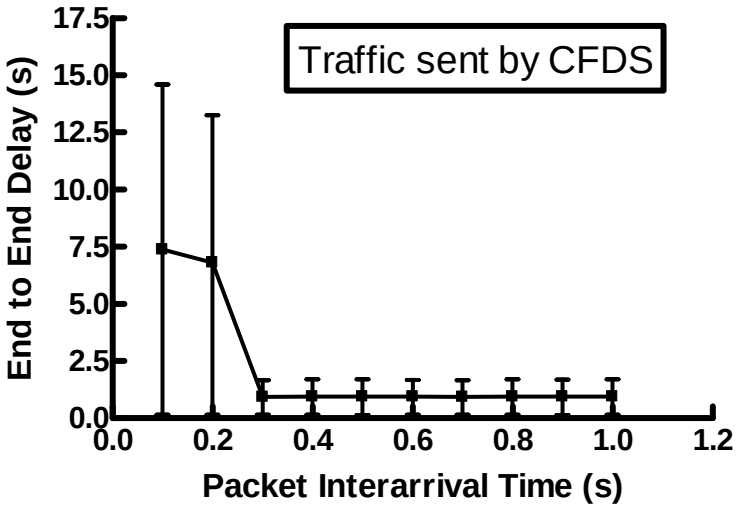

Figure 3.12 End-to-end delay for traffic sent by CFDS

Figure 3.12 shows *End-to-End Delay* of real-time traffic. Compared to Figure 3.11, CFDS buffer overflows more quickly and *End-to-End Delay* is huge in this condition. The reason for this is as follows: we have 8 CSMA/CA slots and 8 CFDS slots. For best-effort traffic, they can share 8 CSMA/CA slots. However, for real-time traffic, each node just requests one CFDS slot because its real-time traffic is light, with the application payload of 100 bits. But when the traffic frequency is more quickly than 0.2 s, CFDS buffer is full and the accumulated traffic has to be sent in the next several superframes. When CFDS buffer is

available, the average *End-to-End Delay* is about 0.97 s. In the following section, we continue to study CFDS delay for better understanding the characteristics of this mechanism.

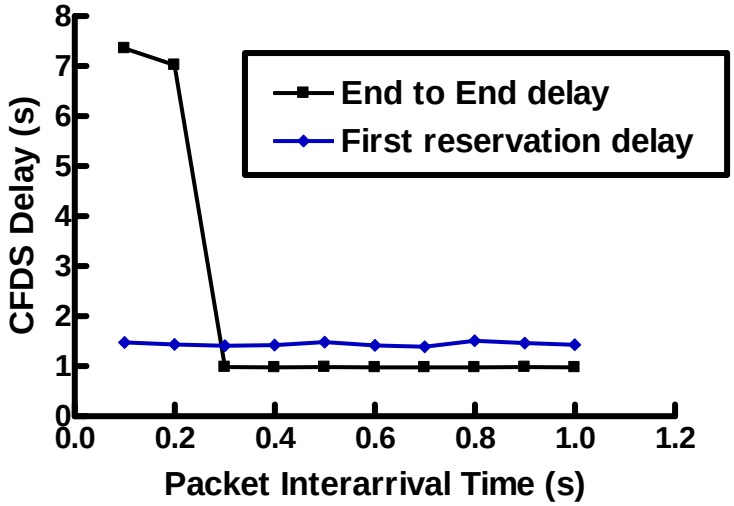

Figure 3.13 CFDS delay composition

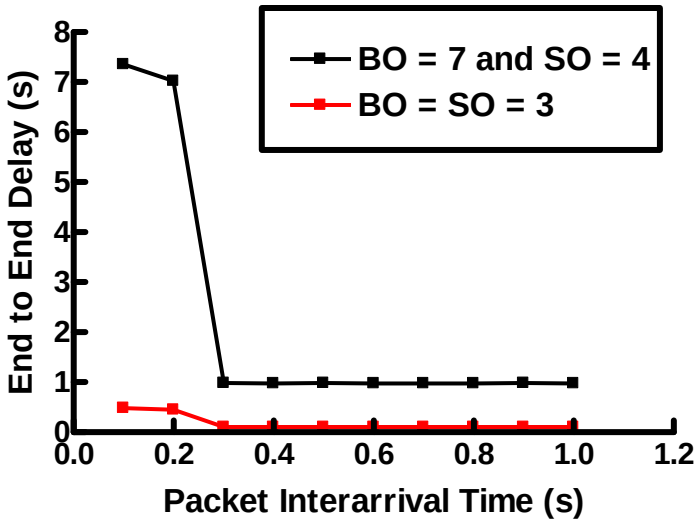

Figure 3.14 CFDS delay with different superframe structures

Based on Figure 3.12, Figure 3.13 shows CFDS delay composition. When the packet arrives at MAC layer, CFDS should be reserved by beacon exchanges. So the time from arrival of the first packet to its allocated slot is called as *First Reservation Delay*. We can see that *First Reservation Delay* keeps the same for all the traffic loads. Then the following packets don't need the slot reservation. They are inserted to CFDS buffer and wait for the corresponding slot.

At last, if we shorten the superframe or remove inactive period, *End-to-End Delay* of both best-effort traffic and real-time traffic will decreases. Figure 3.14 just shows *End-to-End Delay* for traffic sent by CFDS as an example. Obviously, there is a trade-off between *End-to-End Delay* and energy consumption.

3.2.2.Packet success ratio

Packet Success Ratio is the percentage of packets which are received successfully. As shown in Figure 3.15, when buffers are available, traffic sent by CFDS always keeps 100% *Packet Success Ratio*. However, for traffic sent by CSMA/CA, when *Packet Interarrival Time* decreases, much more traffic are transmitted in the network, so *Packet Success Ratio* also decreased as the possible collisions.

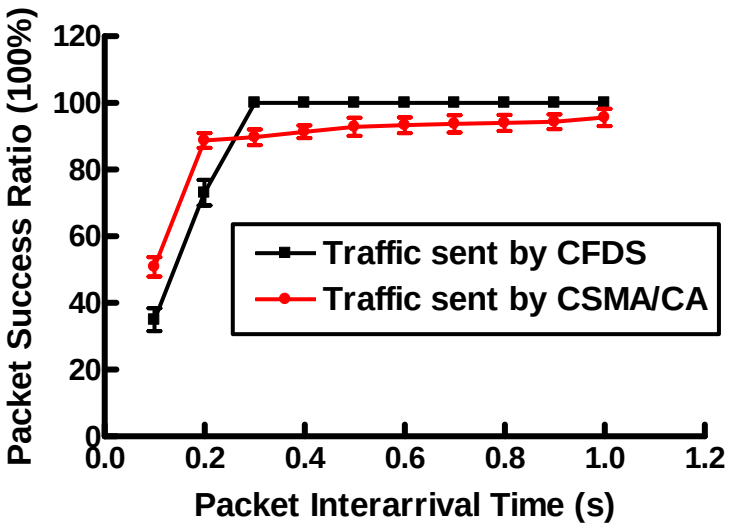

Figure 3.15 Packet success ratio for traffic sent by CSMA/CA and CFDS

Afterwards, we modify the application configuration: there are now 14 nodes with best-effort traffic and 14 nodes with real-time traffic. This change does not affect the traffic sent by CFDS and so we just show the result of best-effort traffic. As shown in Figure 3.16, *Packet Success Ratio* decreases when there are more sources in the network.

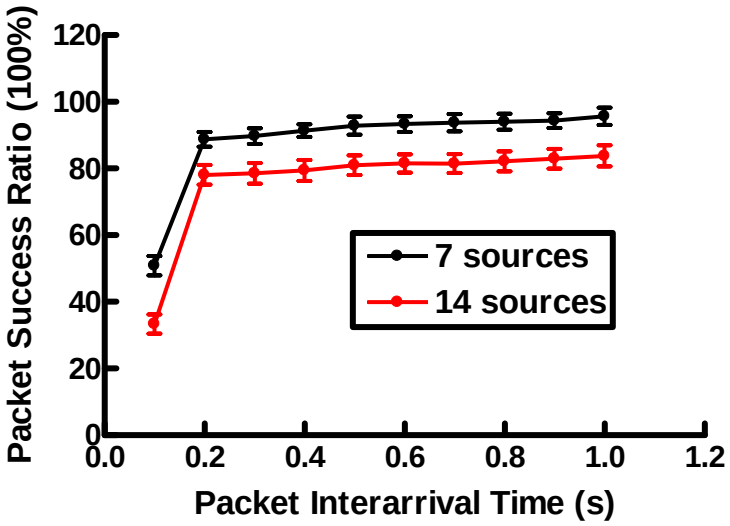

Figure 3.16 CSMA/CA packet success ratio with different sources

In part 3.2, we investigate the QoS capability of ADCF. The simulation results illustrate the different characteristics of CSMA/CA mechanism and CFDS mechanism. CFDS can provide guaranteed delivery, no matter what traffic load conditions. However, *Packet Success Ratio* for traffic sent by CSMA/CA is much affected according to the different traffic loads. For delay, we simulate and analyze its composition. All the *End-to-End Delays* are less than 1 s, with the current parameters. We can also reduce *End-to-End Delay* by changing the superframe structure such as narrowing inactive period, however, this change costs more energy consumption.

3.3.Node join and node failure

In this section, we attempt to evaluate the performance of ADCF with node join or node failure. At the beginning, there are 30 nodes in the network.

In the first experiment, there are 2 sources and 2 destinations. At 200 s, a node that has not application traffic is randomly chosen as failure node. This failure node may be normal node or initiator. Figure 3.17 shows the simulation results. Node density decreases at 200 s as there are fewer nodes in the network. But application throughput is the same. We note that a node failure can not disturb the good running of the network.

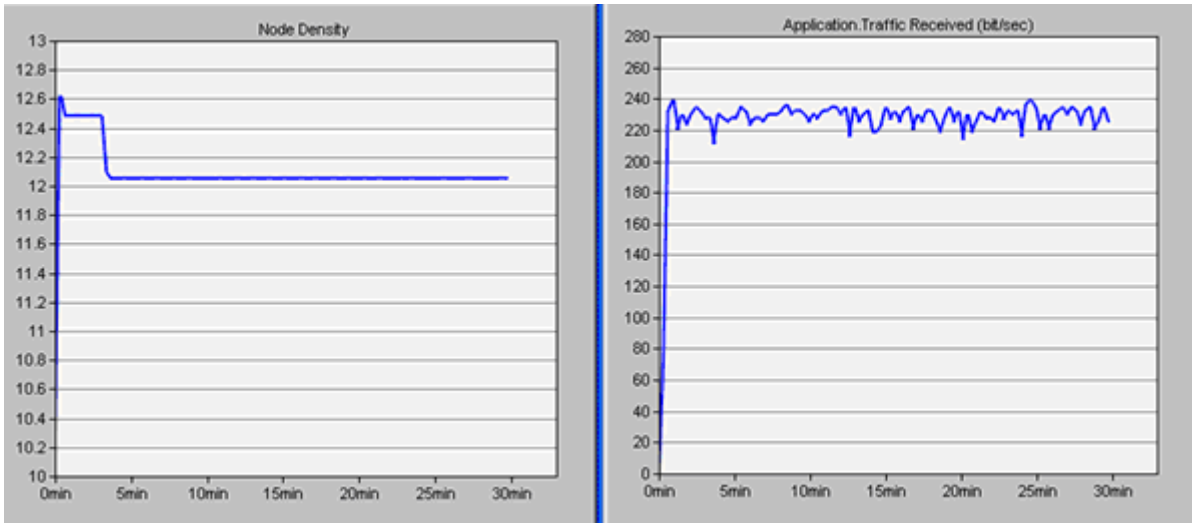

Figure 3.17 Throughput vs. node failure

In the second experiment, 7 new nodes join the network one by one, as shown in Figure 3.18. When the first node, second node, third node gradually add to the network, they could find the free beacon slots and enter working stage immediately. The join of the fourth node causes network rebuilding as there is no available CFBS. Again, BOP augmentation is triggered by the join of the sixth or seventh node. When T_{cycle} is 1.5 s, the average rebuilding time for 37 nodes is less than 27 s.

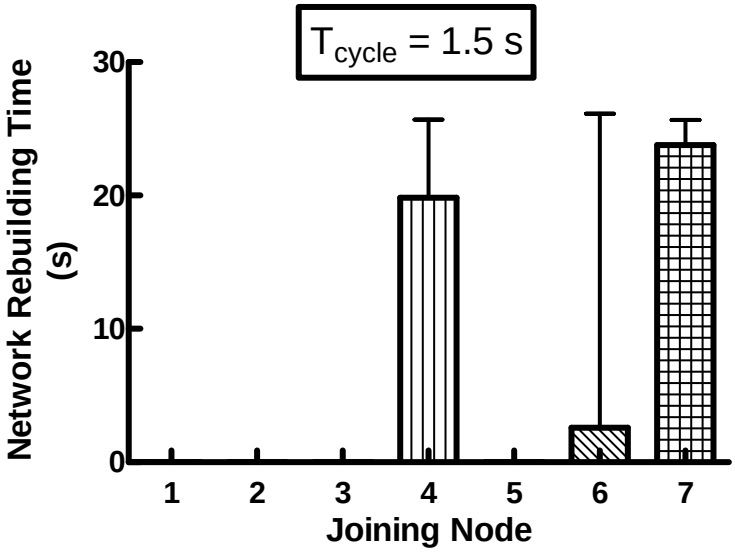

Figure 3.18 Network rebuilding time vs. node join

In fact, the relationship between network performance and topology change is difficult to study because of huge number of complex and specific cases. We simulate the most common cases as examples and only one topology change event is generated each time. Anyway, the

following conclusions can be obtained from the simulation. Firstly, ADCF allows the network to be well performed without single point of failure. In the meantime, the network can work properly with some joining nodes. At last, in the worst case a network rebuilding process is needed and the rebuilding time is acceptable.

3.4.Comparison of ADCF with IEEE 802.15.4

In this section, we compare the MAC performance of ADCF with that of 802.15.4. There are 14 nodes in the network. CSMA/CA performance is investigated at first. Then we compare CFDS with GTS. Last but not least, their energy consumptions are simulated and analyzed.

3.4.1.CSMA/CA performance

In this experiment, there is only best-effort traffic in the network. *Packet Interarrival Time* and number of sources are configured for different scenarios.

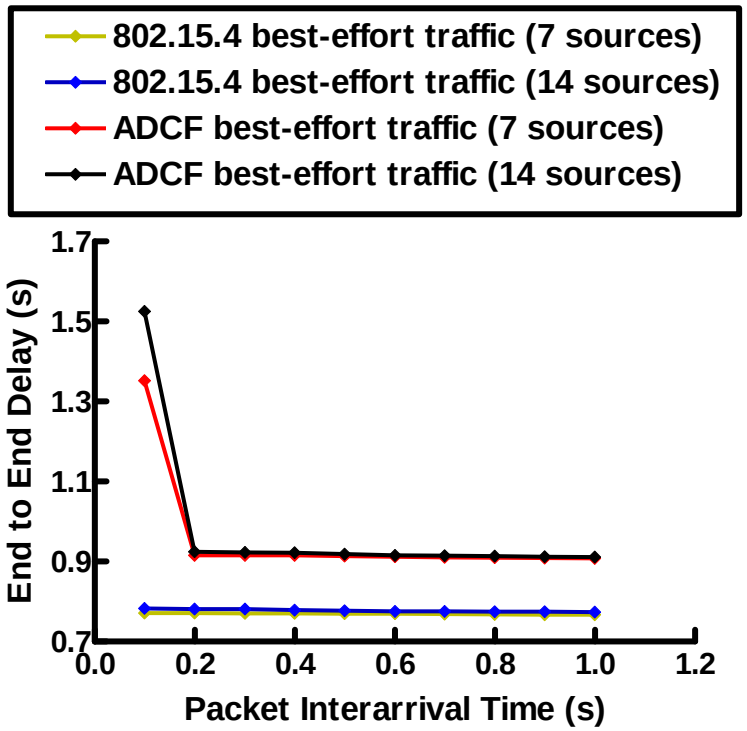

Figure 3.19 End-to-end delay comparison of traffic sent by CSMA/CA

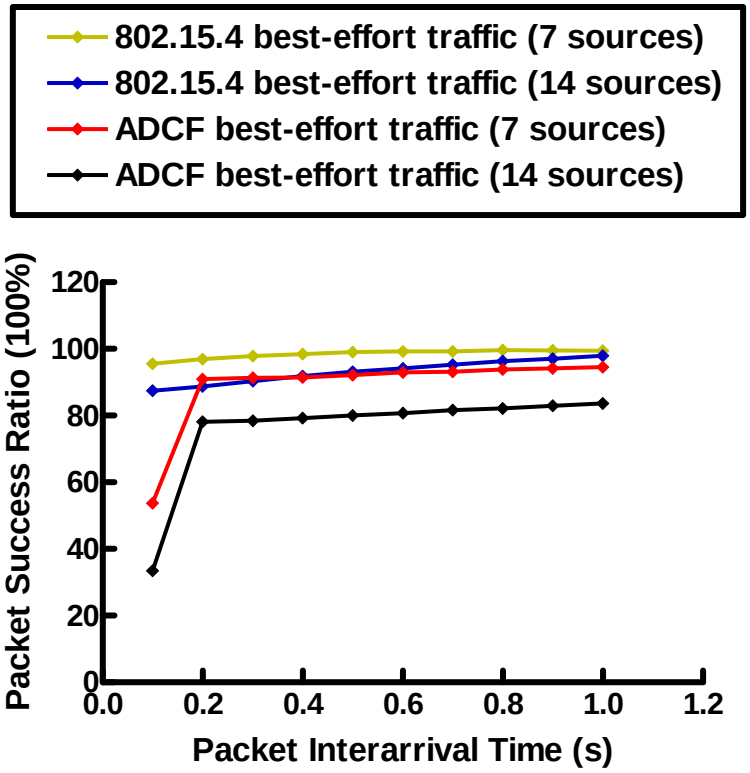

Figure 3.20 Packet success ratio comparison of traffic sent by CSMA/CA

As shown in Figure 3.19 and Figure 3.20, when *Packet Interarrival Time* is less than 0.2 s, CSMA buffer of ADCF is full. Therefore, *End-to-End Delay* sharply increases and *Packet Success Ratio* significantly decreases. At this moment, when there are 7 sources, 38.2% of packet loss is due to buffer overflow and 7.5% of packet loss is due to collisions. When there are 14 sources, 43.1% of packet loss is due to buffer overflow and 24.6% of packet loss is due to collisions. We can see that CSMA/CA of 802.15.4 performs better than ADCF. The reason is even though they have the same traffic loads; 802.15.4 has 2 CAP in a superframe, one for the communication with parents and another for the communication with children. In addition, cluster-tree topology with its active period schedule also limits the collisions.

3.4.2.CFDS and GTS

Only QoS traffic is transmitted in this experiment. There are 7 sources with 1-hop QoS traffic or 3 sources with multi-hop QoS traffic.

As shown in Figure 3.21, there is a small difference for the *End-to-End Delay* of 1-hop traffic. On average, ADCF has 60 ms advantage when *Packet Interarrival Time* is greater than 0.4 s. But for multi-hop traffic, ADCF takes much smaller *End-to-End Delay* than 802.15.4.

Precisely, when buffers are available, *End-to-End Delay* of 802.15.4 is about 1.37 times as large as that of ADCF.

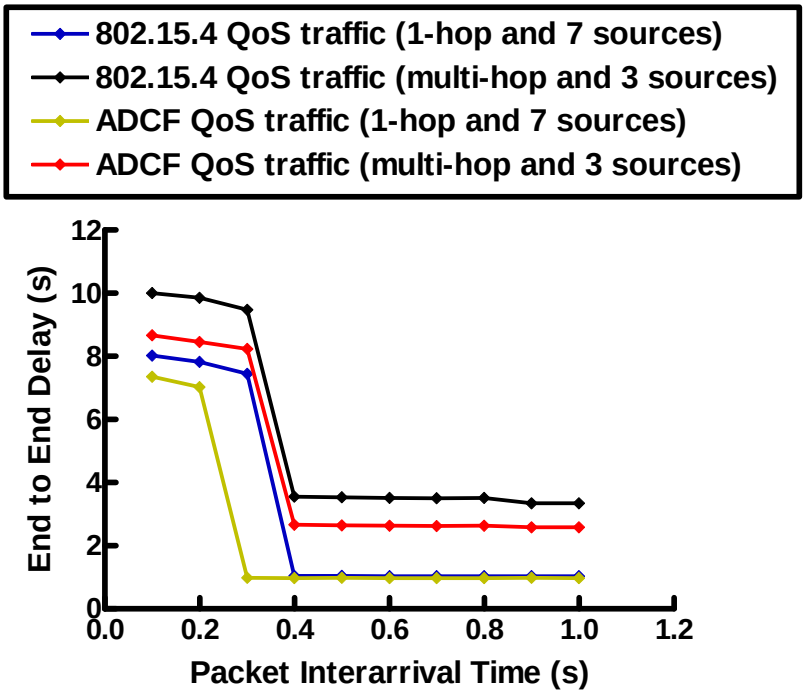

Figure 3.21 End-to-end delay comparison of traffic sent by CFDS

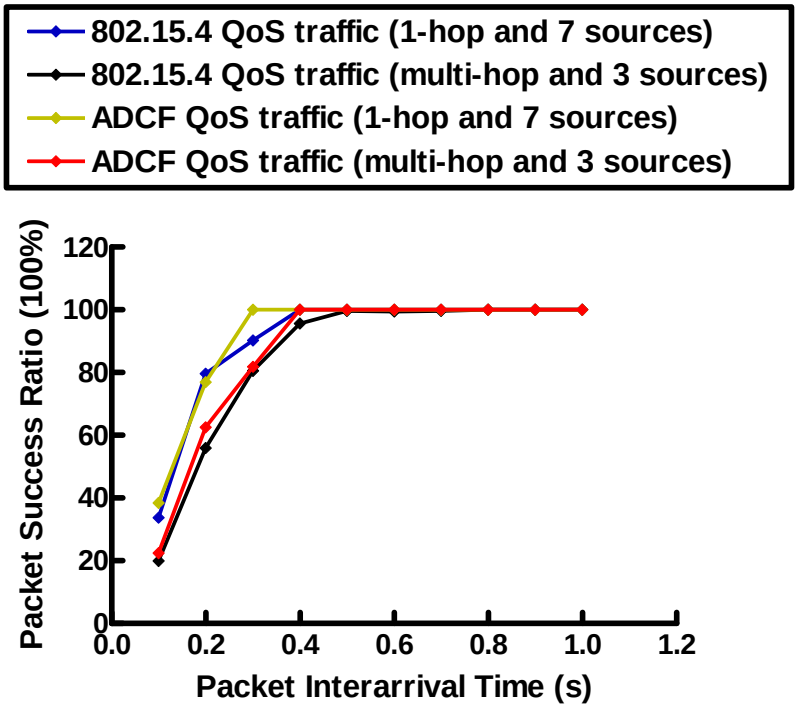

Figure 3.22 Packet success ratio comparison of traffic sent by CFDS

Figure 3.22 shows that all the *Packet Success Ratios* always keep 100% when buffers are available. Otherwise, a lot of packets are dropped because of buffer overflow. The difference between ADCF and 802.15.4 is tiny. Additionally, 1-hop traffic obviously performs better than multi-hop traffic.

In another point of view, hop count has a stronger influence than number of sources and *Packet Interarrival Time* for the QoS traffic. This means the performances of CFDS mechanism are stable, no matter what traffic loads has been chosen.

3.4.3.Energy consumption

In this experiment, no application traffic is delivered, in order to properly evaluate the energy cost of both ADCF and 802.15.4. As explained in 2.3.4, this battery module simulates the energy consumption of radio transceiver and MCU.

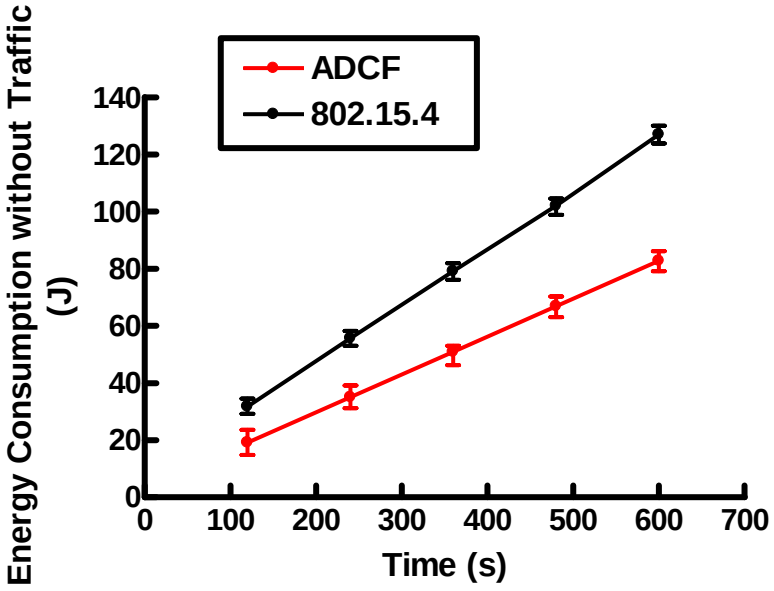

Figure 3.23 Energy consumption comparison

Figure 3.23 shows that ADCF consumes less energy than 802.15.4. This is because of 2 active periods in the cluster-tree network of 802.15.4. Hence more time is spent for idle listening. Typically, about 37.5% of energy can be saved by ADCF.

In summary, we simulate both ADCF and 802.15.4 in the same conditions and compare them with some interesting metrics. For CSMA/CA mechanism, 802.15.4 has a smaller *End-*

to-End Delay and a higher *Packet Success Ratio*. However, ADCF performs better than 802.15.4 for QoS traffic thanks to CFDS mechanism. Most importantly, the simulation result displays the energy consumption advantage of ADCF.

3.5.ADCF performances in large scale and high density network

We focus on studying the *Packet Success Ratio* of ADCF in large scale and high density network in this experiment. Firstly, 30 nodes and 50 nodes are deployed in the network separately. Then the network of 50 nodes with different neighbor density is simulated. For all the scenarios, there are 7 sources with QoS traffic and another 7 sources with best-effort traffic at the same time. All these application traffics are generated to a 1-hop destination.

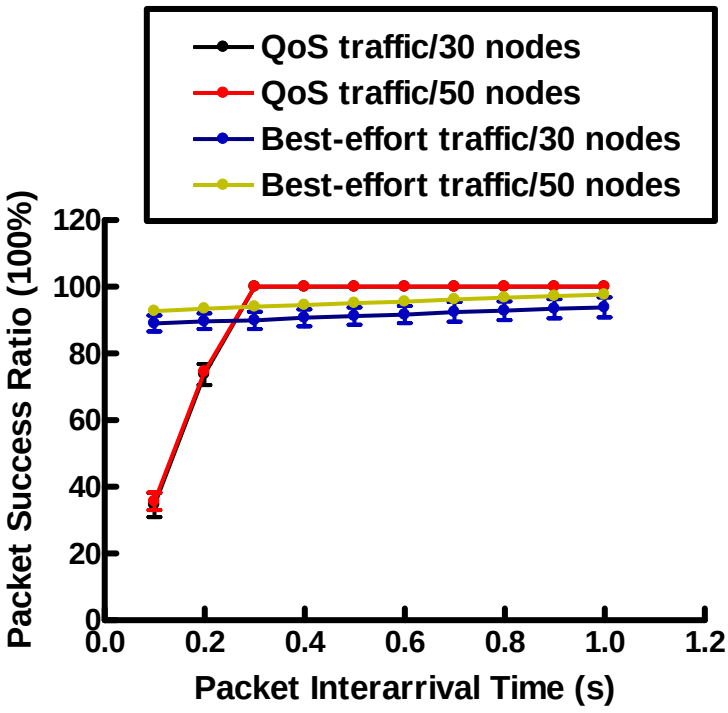

Figure 3.24 Packet success ratio for different scale

As explained before, QoS traffic is transmitted in one CFDS of each superframe while best-effort traffic could be transmitted in 8 shared CSMA/CA slots, so the current traffic load is light for CSMA/CA but relatively heavy for CFDS. As shown in Figure 3.24, when *Packet Interarrival Time* is less than 0.3 s, CFDS buffer becomes full and CSMA buffer is still available. Fortunately, *Packet Success Ratios* always keep 100% for QoS traffic when the

buffers are available, in the network of both 30 nodes and 50 nodes. So the red curve overlaps with black curve. With the same best-effort traffic load, *Packet Success Ratio* becomes higher when the network scale is larger. This is because the risk of collisions is lower in a larger network. Also, *Packet Success Ratio* for best-effort traffic increase with the augmentation of *Packet Interarrival Time*. For example for 30 nodes, *Packet Success Ratio* changes from about 89.04% to 93.85% when *Packet Interarrival Time* increases from 0.1 s to 1 s.

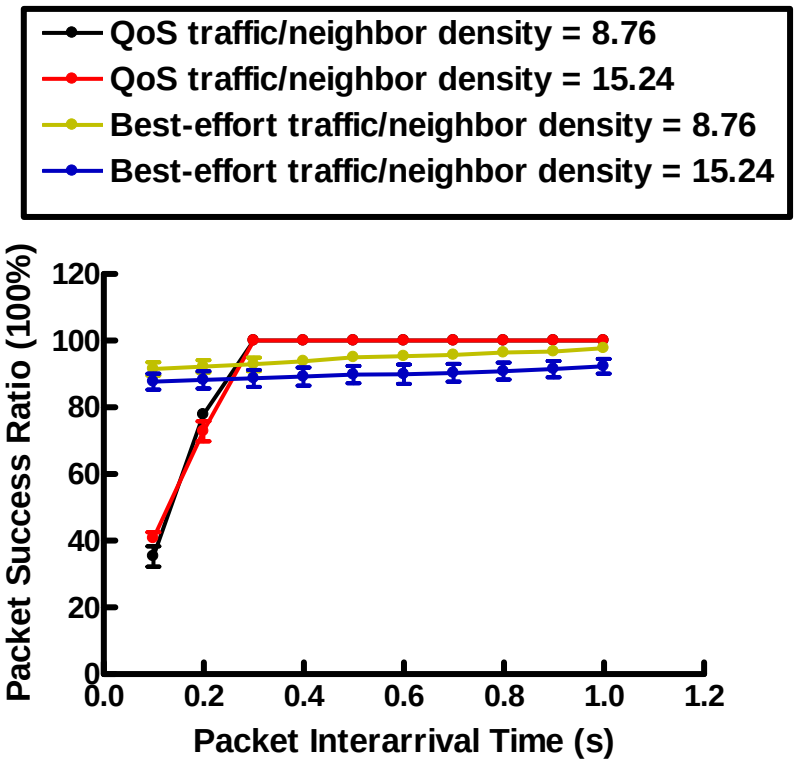

Figure 3.25 Packet success ratio for different density

Neighbor densities are average values obtained by simulation. For example, neighbor density of 15.24 means that each node has about 15 2-hop neighbors in average. As shown in Figure 3.25, network density has no influence on QoS traffic. *Packet Success Ratios* always keep 100% when CFDS buffer are available. While for best-effort traffic, *Packet Success Ratio* becomes higher with the lower neighbor density as there are fewer collisions in the network.

This experiment aims to confirm the performances of our MAC protocol in some interesting scenarios, such as large scale and high density network. The simulation results show the normal running of ADCF. Especially, QoS traffic can be delivered without packet loss, no matter what network configuration is adopted.

4. Conclusion

We began this chapter from an investigation of simulation tools for WSN. Finally OPNET is chosen as its high-quality programming, user-friendly GUI and data processing capability. Most importantly, OPNET contains a complete IEEE 802.15.4 implementation.

Afterwards, we presented our ADCF simulation model which implemented all the proposals of Chapter 2. The network modeling was organized with ADCF node modules. Each layer of the node was illustrated in the process domain. Meanwhile, the simulation configurations and parameters were given in this part.

At last, many experimental scenarios were simulated and some interesting results were shown. In the application challenges of chapter 1, we had classified the performance metrics of MAC layer: QoS, energy saving, flexibility and robustness. So we discussed the simulation results from these 3 metrics:

- ADCF satisfies our application request of delivering QoS traffic without packet loss. End-to-end delay depends on the superframe structure and our simulation results may provide user the reference configurations. We can confirm that ADCF is never worse than 802.15.4 for delivery of QoS traffic. In some cases, ADCF can be even better than 802.15.4 thanks to the availability of mesh topology.

- The cost of ADCF is acceptable. We can build a mesh network of 30 nodes in 25 s and with little overhead. Obviously, the cost of ADCF also includes its energy consumption. Simulation result shows that ADCF consumes less energy, about 37%, than 802.15.4. We can further improve the performances such as convergence time and end-to-end delay at the price of energy consumption. So the trade-offs should be made according to specific application environments.

- For flexibility and robustness, a lot of cases are studied but it is difficult to lead an exhaustive study. However, we can confirm that ADCF could tolerate node failure. Compared with star or tree topology, thanks to ADCF, the network works properly even though there are some failure nodes. Also, new nodes could join the network freely, increasing the flexibility of the network. In some cases such as a multi-hop network with free CFBS, new node can perfectly insert the superframe

and send beacons without collision. Other cases may cause network rebuilding. An example of simulation result shows the cost of this rebuilding.

In terms of scalability, we can consider that 50 nodes are enough for an indoor environment. The simulation results show that both network scale and neighbor density have no influence on QoS traffic which sent by CFDS mechanism. QoS traffic could be sent without packet loss as the stable performance of ADCF.

The current simulation work and simulation results verify the advantages of ADCF. In the next chapter, we will present the implementation of ADCF on material prototypes and the network deployment in our smart home “Maison Intelligente” of Blagnac. Some limitations and deficiencies about simulation will be discussed in the conclusion.

Reference

[3.1] E. Egea Lopez, J. Vales Alonso, A. S. Martinez Sala, P. Pavon Marino, J. Garcia Haro, "Simulation Tools for Wireless Sensor Networks", International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECT 05), pp. 2-9, July 2005

[3.2] Harsh Sundani, Haoyue Li, Vilay Devabhaktuni, Mansoor Alam, Prabir Bhattacharya, "Wireless Sensor Network Simulators – A Survey and Comparisons", International Journal of Computer Networks (IJCN), Volume 2 Issue 5, pp. 249-265, April 2010

[3.3] Fei Yu, Raj Jain, "A Survey of Wireless Sensor Network Simulation Tools", http://www.cse.wustl.edu/~jain/cse567-11/ftp/sensor/index.html#tossim

[3.4] NS-2, http://www.isi.edu/nsnam/ns/

[3.5] TOSSIM, http://docs.tinyos.net/index.php/TOSSIM

[3.6] OMNeT++, http://www.omnetpp.org/home/what-is-omnet

[3.7] OPNET, http://www.opnet.com/

[3.8] P. Jurcik, A. Koubaa, M. Alves, E. Tovar, Z. Hanzalek, "A Simulation Model for the IEEE 802.15.4 Protocol: Delay/Throughput Evaluation of the GTS Mechanism", 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOT 07), pp. 109-116, October 2007

[3.9] IEEE 802.15.4/ZigBee OPNET Simulation Model, http://www.open-zb.net/wpan_simulator.php

[3.10] N. Fourty, A. Vand Den Bossche, T. Val, "An Advanced Study of Energy Consumption in an IEEE 802.15.4 Based Network: Everything but the Truth on 802.15.4 Node Lifetime", Computer Communications, Special Issue: Wireless Green Communications and Networking, Volume 35 Issue 14, pp. 1759-1767, May 2012

Chapter 4

Prototype Implementation

Prototype implementation is a fundamental approach to verify a protocol and its performances. We begin this chapter by introducing a fast prototyping platform named WiNo. The supported sensor boards and some useful tools are also presented. Then we explain the implementation of ADCF, including some improvements such as link state confirmation and node state transition. The following part shows the representative results obtained through practical measurements in a real environment and compares these results with simulation. At last, the deployment of ADCF in a real application context using our smart home of Blagnac will be presented and concrete conclusions will be given.

1. Platforms and tools for prototyping

The objective of this section is to illustrate the platform and the tools used for the prototyping. Firstly, a platform with its important management issues and mechanisms is briefly introduced. Two types of sensor boards came from *Freescale* are adopted and their basic characteristics will be presented. The last of this section introduces some useful tools for debugging the prototype and also for collecting the results.

1.1. WiNo platform

WiNo [4.1] is an open platform ready to accommodate protocols at MAC layer or network layer for wireless sensor networks. It provides an open environment adapted to research projects, including the management of physical layer and the necessary tools to develop a full but very compact protocol stack using ANSI C-language. More accurately, a developer with WiNo can master not only the time access to the medium and the sleep-wake cycle, but also the CPU time and memory resource generally restricted by hardware in a WSN.

WiNo consists of two sub-systems: *WiNoEmu* and *WiNoTB*. WiNoEmu is an emulator under GNU/Linux, and WiNoTB is a test bed targeted real environment. Both of them share the same *WiNoKernel* and its characteristic details will be presented in the following parts. Obviously, protocol implementation process is simplified by the possibilities of jointing emulated nodes with deployment on the final target, without changing codes between the emulator and the test bed.

1.1.1. Physical layer

WiNoKernel controls the unreliable physical layer which meets the 2.4 GHz PHY of IEEE 802.15.4-2006 specifications. The physical layer not only includes the two main primitives, *PD_data_request ()* and *PD_data_indication ()*, for message transmission and reception, but also some management PHY primitives such as enabling/disabling reception, requesting for channel assessment, setting transceiver to *doze* mode, etc.

1.1.2.Queue memory management

As the available memory on the final target is very limited, the memory management dedicated to queues and buffering frames is a delicate task.

For memory management, the creation of each frame, packet or message needs a memory allocation request. Typically, this memory allocation is done by a top layer during transmission or the physical layer at a reception from the medium. So the allocation request is performed by calling one of the functions: *allocTxFrame ()* or *allocRxFrame ()*. The two functions will assign the first free frame in memory. WiNoKernel just uses the allocated frame by an associated pointer during passing through each layer, except the last object that physically contains the data to send, to save memory and CPU time. Once the frame has been treated, the memory must be freed by calling *freeTxFrame ()* or *freeRxFrame ()*.

And for queue management, once a frame from the top layer reaches MAC layer, it will be inserted into the corresponding buffer according to its transmission mode, *Best Effort Mode* or *Quality of Service Mode*.

1.1.3.Clock and interrupt management

1.1.3.1. Local clock and shared clock

In order to be accordance with what is required by the IEEE 802.15.4 standard, a local clock performing a timer symbol (4 μs) of 24-bit resolution (a cycle of 67 s) must be present on the targeted node. WiNoKernel provides two clocks, a local clock to the node and a shared clock to all the nodes, coded on 32-bit. So the synchronized cycle is more than 4 h. For WiNoTB, the shared clock needs time synchronization protocols such as SISP [4.2].

1.1.3.2. Interrupt handler

The interrupt handler allows programming a function call at a given moment. By reference to the local clock, the interrupt may be relative or absolute. An interrupt vector containing the list of interrupts is set and the size of this vector is defined by a constant. Because of the limited processor on the final targeted node, the interrupt is triggered even if the start time is exceeded.

1.1.4.Encapsulation and de-encapsulation mechanism

Once memory allocation is well performed as described in the previous part, the following rules must be observed when a new frame passes through all the protocol layers.

When receiving a frame, the *length* field of the frame contains the length of the received frame. An *offset* field has been added to indicate the first byte to the upper layer when decapsulating. When *offset* and *length* are equal, the entire decapsulation will be performed.

When creating a frame to be transmitted, the frame will always be completed by the *footer*. The higher layer places its data unit to the end of the reserved memory space and uses the *length* field as pointer to indicate the stop of the data to the lower layer, and so forth for the encapsulation at each layer. Unlike the reception, there is no *offset* filed here. At physical layer, the frame is transmitted with the length of *MAX_FRAME_LENGTH*.

In all cases, the *userLevel* field of the frame must be updated with a defined value corresponding to the protocol layer which possesses the frame.

1.1.5.Neighbor table management

Adhoc approach involves storing an important amount of information concerning neighbor nodes. Some classical information such as the sequence number of the last received frame of a neighbor for detecting duplications of the received frames (LLC sub-layer) need to be saved. Some other specific data such as neighbor address or neighbor energy are also recorded. Therefore, a neighbor table management is implemented by WiNoKernel.

Classic access mechanisms including *insert*, *update* and *delete* for the neighbor table are implemented with the constraint of memory/CPU. For example, we can retrieve the index of a node in the neighbor table and modify the corresponding attributes directly. To make possible cross-layer approaches, reading/writing neighbor table is not reserved to the MAC layer.

1.2.Sensor application boards

In the previous section 1.1, the platform WiNo of emulation and rapid prototype for WSN was presented. Currently, the supported nodes are those based on or derived from *Freescale* microprocessor *9S08GT60* and *transceiver MC1319x*: *13192-SARD*, *1321x-SRB*, *1321x-NCB* or even *Freescale ZRD01*.

In fact, WiNoTB is based on *driver open source Simple MAC* [4.3] of *Freescale*, but it optimizes SMAC in several details, such as the utilization of *timers* of the radio component.

The following two parts will briefly introduce two types of sensor nodes available in our laboratory. At present, we have four nodes of 13192-SARD and four nodes of 1321x-SRB, as shown in Figure 4.1.

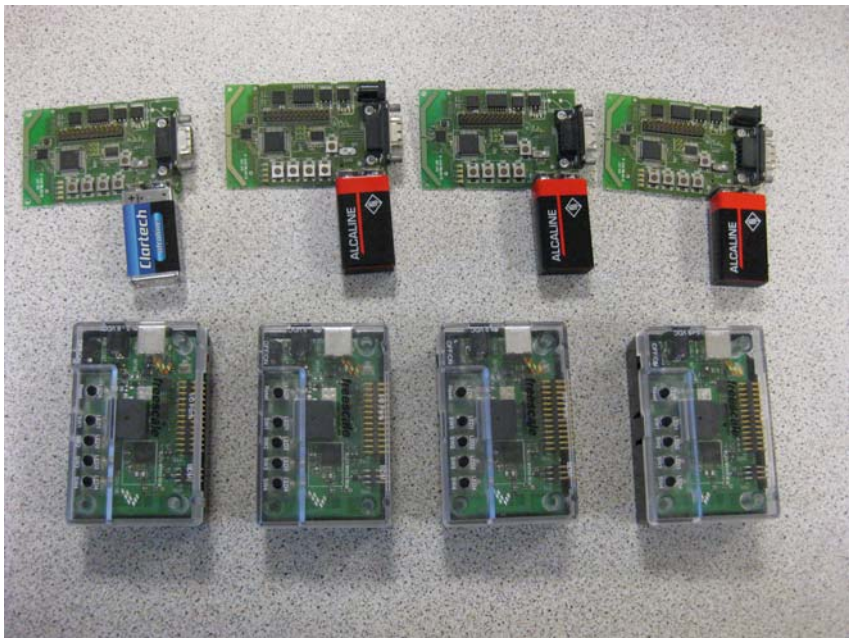

Figure 4.1 Available targeted nodes

1.2.1.Freescale 13192-SARD

13192-SARD indicates the MC13192 Sensor Application Reference Design [4.4]. As shown in Figure 4.2, 13192-SARD includes the following features:

- 1: MC13192 2.4 GHz transceiver RF reference design with printed circuit antenna. It supports IEEE 802.15.4 PHY.
- 2: MC9S08GT60 low-power, low-voltage 8 bits MCU with 4KB of RAM and 60KB of on-chip Flash.
- 3: Background Debug Module (BDM) programming port for support of Metrowerks CodeWarrior [4.5] Development Studio.
- 4: Two Accelerometers MMA6261Q (X and Y axis) and MMA1260D (Z axis).
- 5: RS-232 port for interface with a personal computer.

- 6: Four switches and LEDs for control and monitoring.
- 7: Reset switch for program reset.
- 8: 9 V battery or 2.1 mm power connector which allows a supply of 5.5 to 9 V.

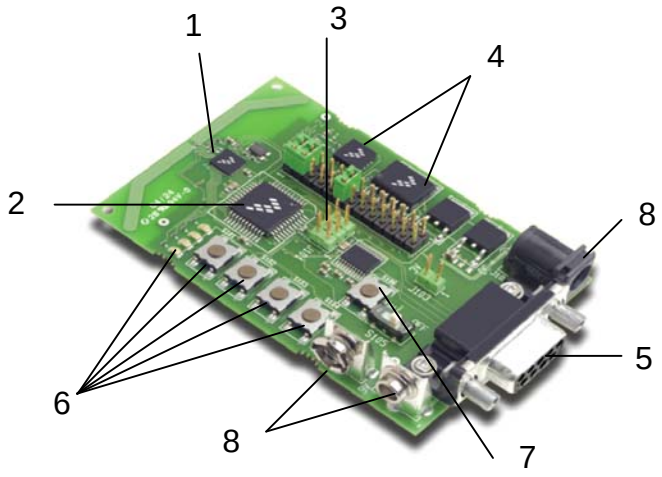

Figure 4.213192-SARD node

1.2.2.Freescale 1321x-SRB

The 1321x-SRB, Sensor Reference Board, is an 802.15.4/ZigBee evaluation board [4.6]. It provides USB connectivity to a PC for easy evaluation. As shown in Figure 4.3, 1321x-SRB contains the following features:

- 1: MC13213 RF transceiver which is an IEEE 802.15.4 compliant radio operating in the 2.4 GHz ISM frequency band.
- 2: The microcontroller unit is based on the HCS08 MCU and provides up to 60KB of flash memory and 4KB of RAM.
- 3: 2.0 USB port.
- 4: 2*3 pin BDM connection allowing flash programming and in-circuit debug via the included USB Multilink Cable.
- 5: The MMA7260Q Acceleration Sensor provides the 1321x-SRB with unique applications to demonstrate wireless sensing solutions.

- 6: Four Push buttons and four LEDs.
- 7: Reset button and one power switch.
- 8: Power connector of 5-9 V or battery holder of 2*AA.

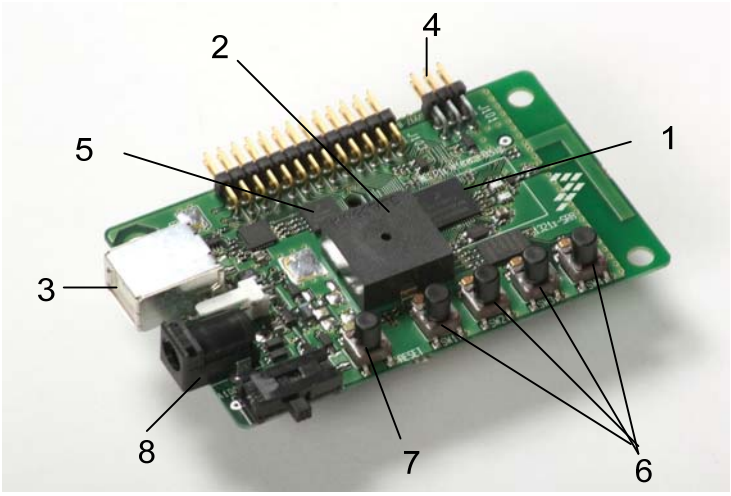

Figure 4.31321x-SRB node

In conclusion, both 13192-SARD and 1321x-SRB nodes are open source and can be programmed with CodeWarrior. For example, we can configure the node's address with the push of one button. All of these helpful features of the nodes make the implementation of a protocol possible and simpler.

1.3.Other useful tools

Once a protocol is implemented on the final targeted nodes, performance analysis tools are commonly needed. Many tools for the evaluation of protocol performance in real conditions can be imagined as the open architecture of the nodes. Actually, two tools are mainly used in our prototyping.

1.3.1.Console and a central server

Both 13192-SARD with the RS-232 port and 1321x-SRB with the USB port can print the protocol details into console. So the console can be a valuable tool for debugging, measuring and collecting. Also, we can send the results of every console to a central server for better analyzing the protocol performances.

For example, as shown in Figure 4.4, one console represents one targeted node. We can open the consoles in the same PC or several different PCs. By a small program *data-logger*, the information of consoles would be written and saved in a same text file of one central server. Therefore the nodes activities and their interactions can be obtained and analyzed with the same timer of the central server.

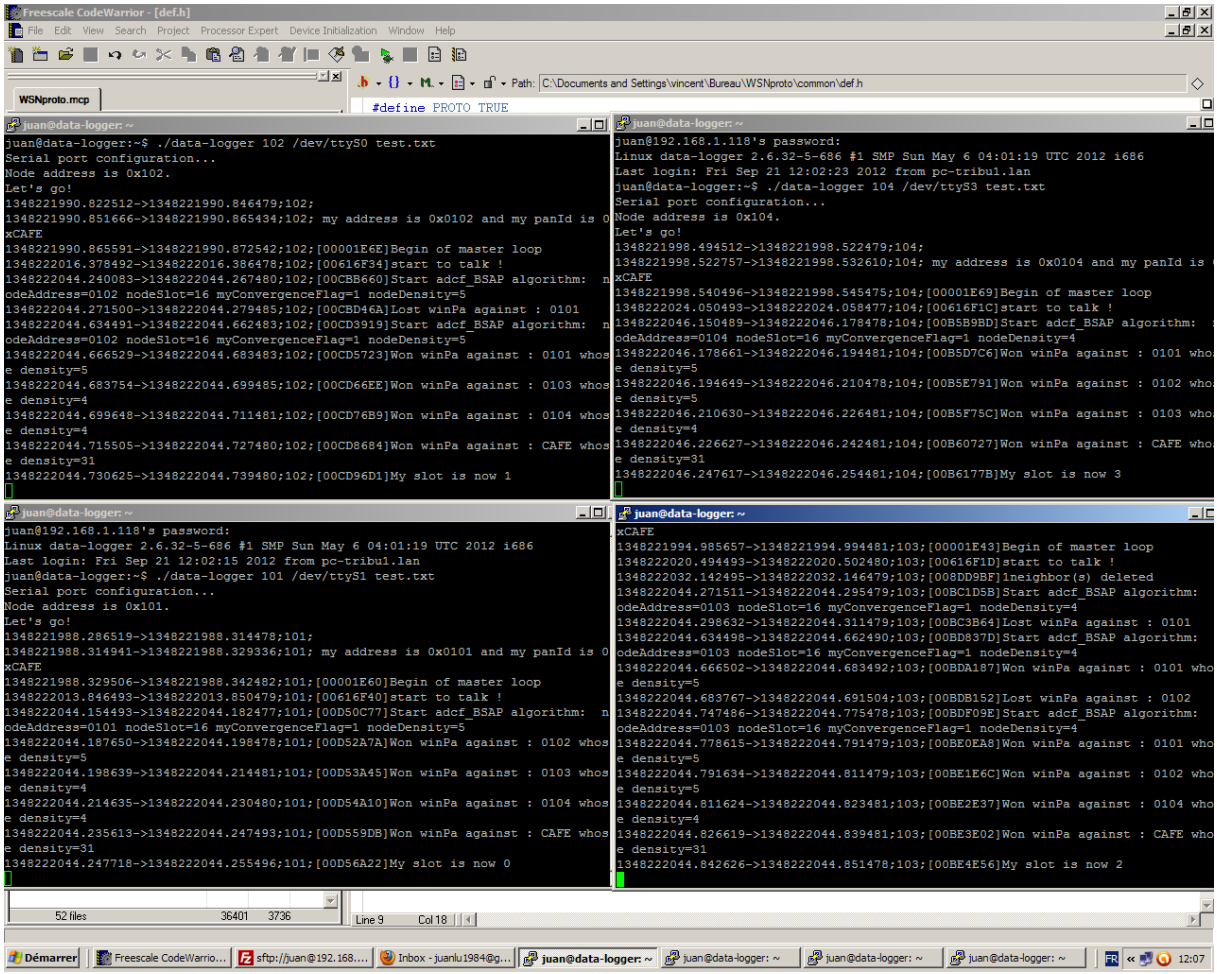

Figure 4.4 Data-logger Console

Unfortunately, we could not write too much to console as the limited speed of 38400 bps. In addition, it brings delays to the real-time application. For example, SARD nodes and SRB nodes have different output time and can not work well together when consoles are open. So another useful tool, protocol analyzer, had been considered.

1.3.2.Daintree's sensor network analyzer

Sensor Network Analyzer (SNA°) includes protocol analyzer software fitted to sensor networks and sensor network adapter hardware, as shown in Figure 4.5. SNA known as an

expert tool for IEEE 802.15.4 and ZigBee provides comprehensive solution for developing, decoding, debugging and deploying wireless embedded networks [4.7].

Sensor network adapter hardware is a powerful node which can capture all the packets of a network but do not participate in this network. By SNA software, then we can observe network operations and analyze protocol performances without disturbing the original network. More precisely, SNA includes a protocol decoder that allows user to drill down to packet, field and byte level. Thanks to its very high sensibility and the unique visualization capabilities, SNA allow user to view all network nodes and interactions simultaneously, even if the nodes are not 1-hop neighbors, which is very important capability to evaluate ADCF performances. The only condition is that two nodes do not send data exactly in the same time.

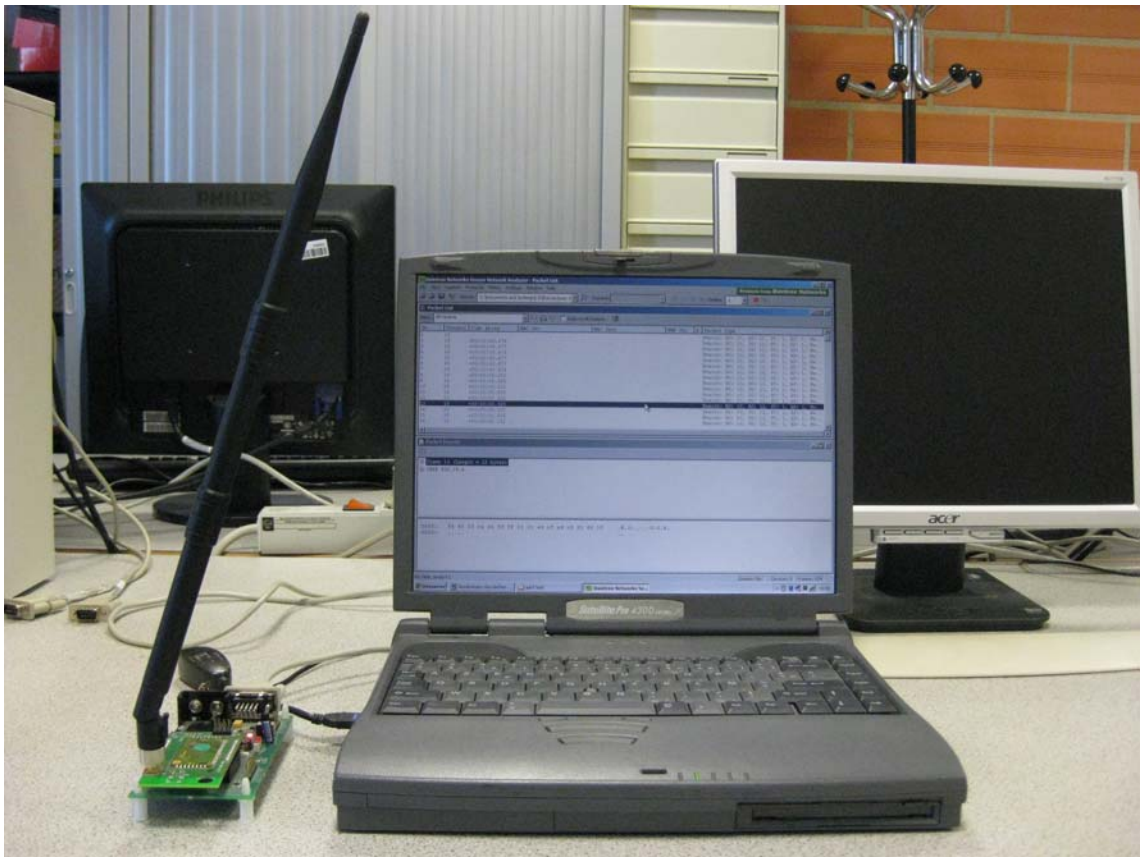

Figure 4.5 SNA software and hardware

2. ADCF implementation

All the preparation work for prototyping, including platform, targeted sensor nodes and performance analysis tools, has been presented. Now we are going to describe ADCF implementation.

2.1.Improvements for prototyping

Firstly, let's recall ADCF protocol. Figure 4.6 displays node state transition achieved in simulation. An ADCF node starts with *adcf_start* and enters *NB_DISCOVERY* state. When receiving beacons with IF of 1 from neighbors or sending *n* beacons itself, this node sets IF and CF (Initiator Flag and Convergence Flag equal 1) and goes to *INIT_SELECTION* state. After *m* beacon intervals, the node enters *WAITING_PRIORITY* state. Once a collision-free beacon slot is decided, the node is therefore in *BOP_READY* state. At last, different topological changes lead the node to different states. For example, *BOP_READY_TEMP* state which keeps the collision-free beacon slot or *INIT_SELECTION* state which means the spark of a network rebuilding process.

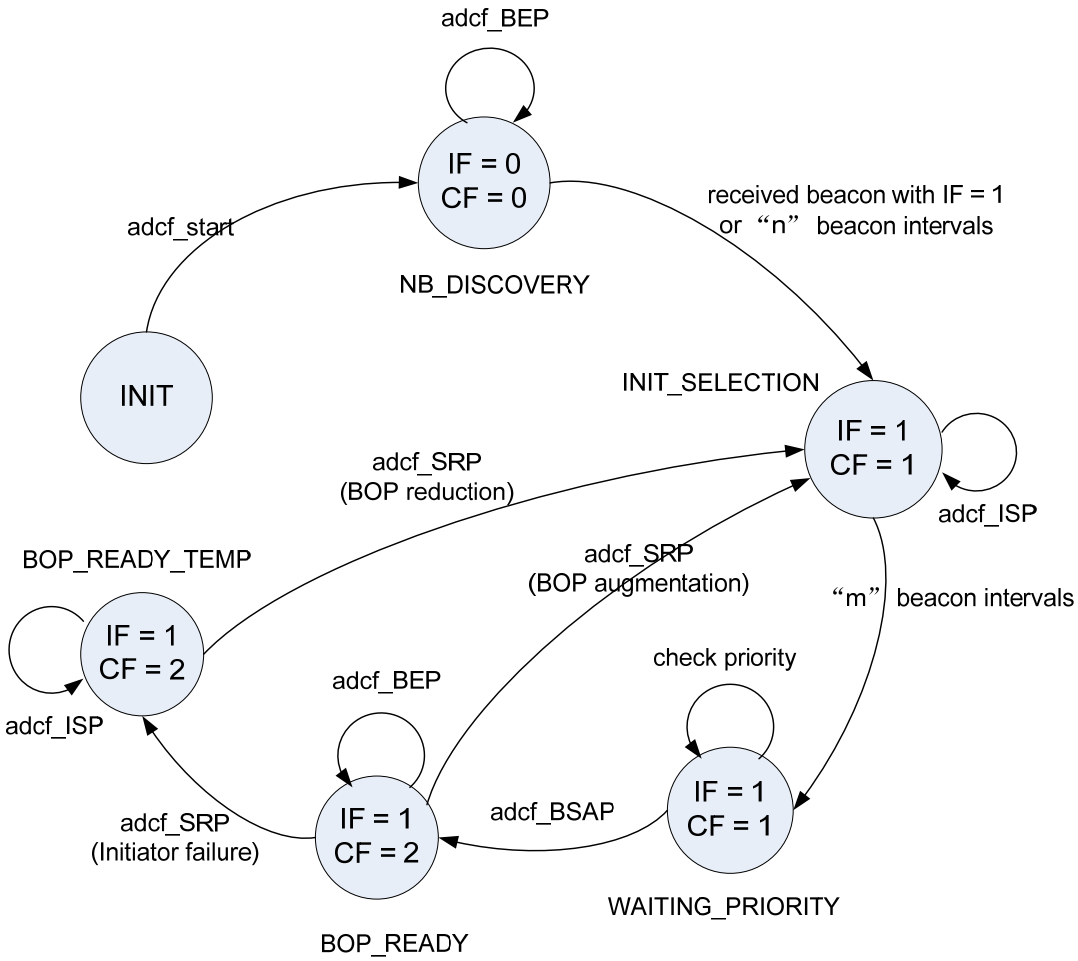

Figure 4.6 Node state transition

Now some details must be improved for prototyping. In the following parts, we will present link state confirmation mechanism, synchronization mechanism and the beacon frame finally implemented in prototype.

2.1.1.Link state confirmation

In part 2.2.6 of chapter 2, a link state confirmation mechanism has been explained. We estimate the unreliable wireless link by beacon loss. In simulation, link state is not considered and studied because of the utilization of an ideal physical model. Additionally, some constant parameters such as T_{sample} are related to the simulation time. However, targeted nodes have different clocks in real environment. So these constant parameters are defined with a distributive manner in prototype.

Another case, asymmetric link, may often happen in real environment. In fact, ADCF supports asymmetric link as the following reason. Two nodes connected with an asymmetric link could make two decisions when choosing CFBS. If they choose different CFBS, that's what we expect. Else, one node can not hear the neighbor node and may choose the same CFBS, at last this asymmetric link will be abandoned as the two nodes always send their beacons at the same moment. Also, the periodical check of CFBS has been added in prototype, ensuring the collision-free beacon slots during topological changes.

2.1.2.Synchronization mechanism

In reality, the clocks on different nodes usually have different values. Instead of time synchronization protocol planned in perspective, a simple synchronization mechanism is used in prototype. The superframe is fixed as a constant value even though there is no superframe in the initialization stage of ADCF. Initiator sends its beacon in slot 0 periodically. Once the nodes receive beacons from initiator, they first estimate the transmission time (based on the length of the beacon frame) and then run modulo operation to determine the offset between their local clock and the clock of the initiator. Then they modify their time of slot 0 and become synchronized. Step by step, multi-hop nodes are gradually synchronized when they receive synchronized beacons from their neighbors, of course even if they are not in the neighborhood of the initiator node. Hence, a *WAITING_SYNCHRONIZED_NB* state has been added in prototype, as shown in Figure 4.7.

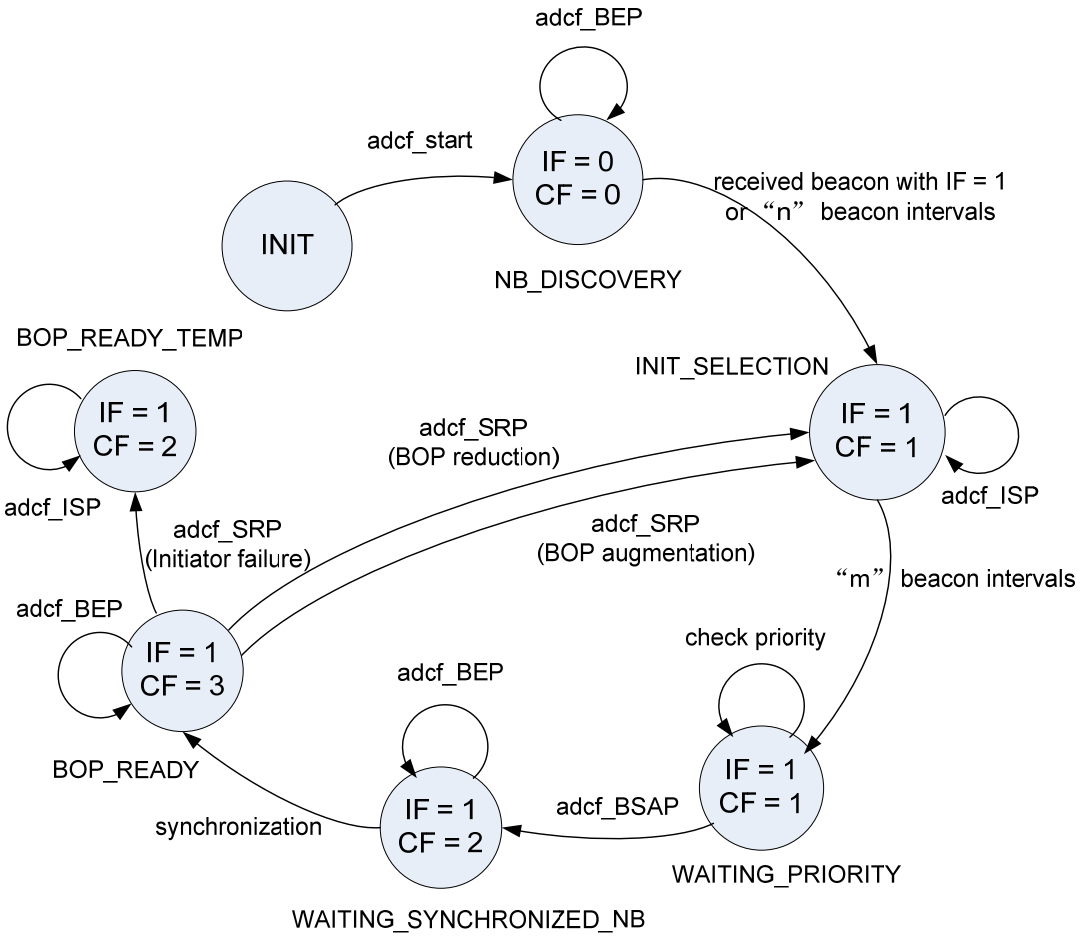

Figure 4.7 Node state transition for prototyping

2.1.3.Beacon frame format

Figure 4.8 presents the beacon frame format implemented on the prototype.

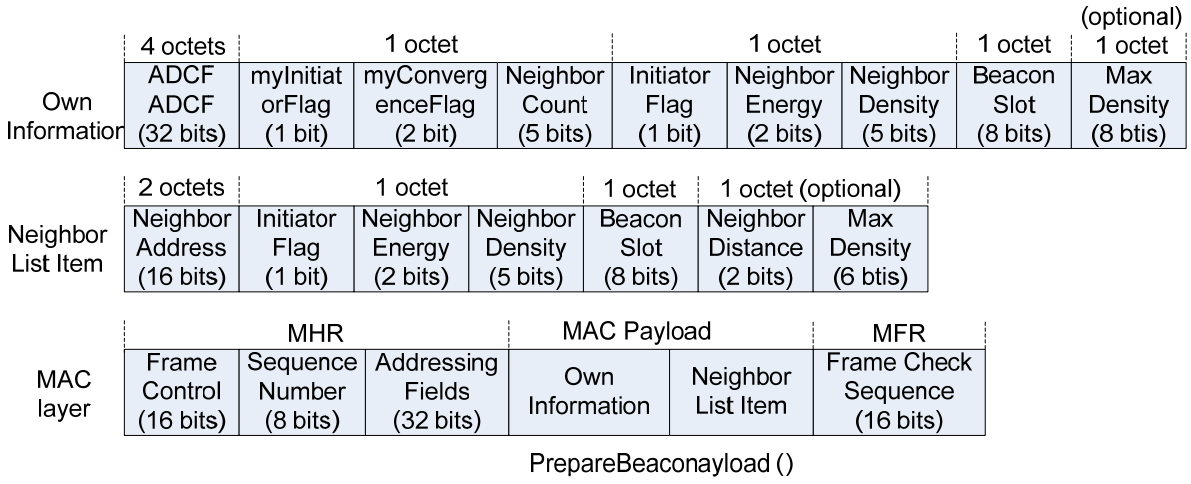

Figure 4.8 Beacon frame format for prototyping

Three modifications affect the fields order changes. *Own Information* and *Neighbor List* are separate to save 2 bytes of address field. Secondly, 4 bytes at the beginning of *Own Information* are reserved for a better time synchronization protocol. At last, 1 byte is added to detect the distance between source node of the beacon and initiator. This *Neighbor Distance* is useful in SRP.

2.2.An example with SNA

In this part, a simple example has been used to better explain ADCF implementation. We start 4 nodes 0101, 0102, 0103 and 0104 one by one. They are all in the communication range of the others, so a full mesh network would be built by ADCF. Figure 4.9 shows the results obtained with SNA.

At the top of each SNA terminal, the captured beacons and the corresponding time intervals are shown. The bottom of each terminal displays the details of a beacon frame. Yellow area indicates address field. Blue area contains *Initiator Flag*, *Neighbor Energy* and *Neighbor Density*. Purple area is beacon slot. Additionally, the octets with black underline are MAC header. Here the activity of node 0101 is concentrated as example and so its beacons are displayed.

In (1), node 0101 sends beacons each 0.476 s at the beginning. So the beacon only includes its own information. Beacon slot is not defined (10 by default).

In (2), node 0101 discovers three neighbors 0102, 0103 and 0104. Therefore *Neighbor Density* field becomes to 4. All the nodes send their beacons without superframe schedule at this moment.

In (3), node 0101 is selected as initiator as its highest priority. Blue area of node 0101 changes from 64 to e4, meaning *Initiator Flag* field is set to 1.

At last in (4), we can see that the nodes well choose the beacon slots. Node 0101 occupies slot 00, node 0102 uses slot 01, slot 02 is for node 0103, and node 0104 sends beacons in slot 03. From the red block at the top of (4), a superframe of about 1.421 s is well organized. We can see that the nodes send their beacons without collision. The beacon intervals are approximate 0.035 s. That’s just right the length of a beacon slot.

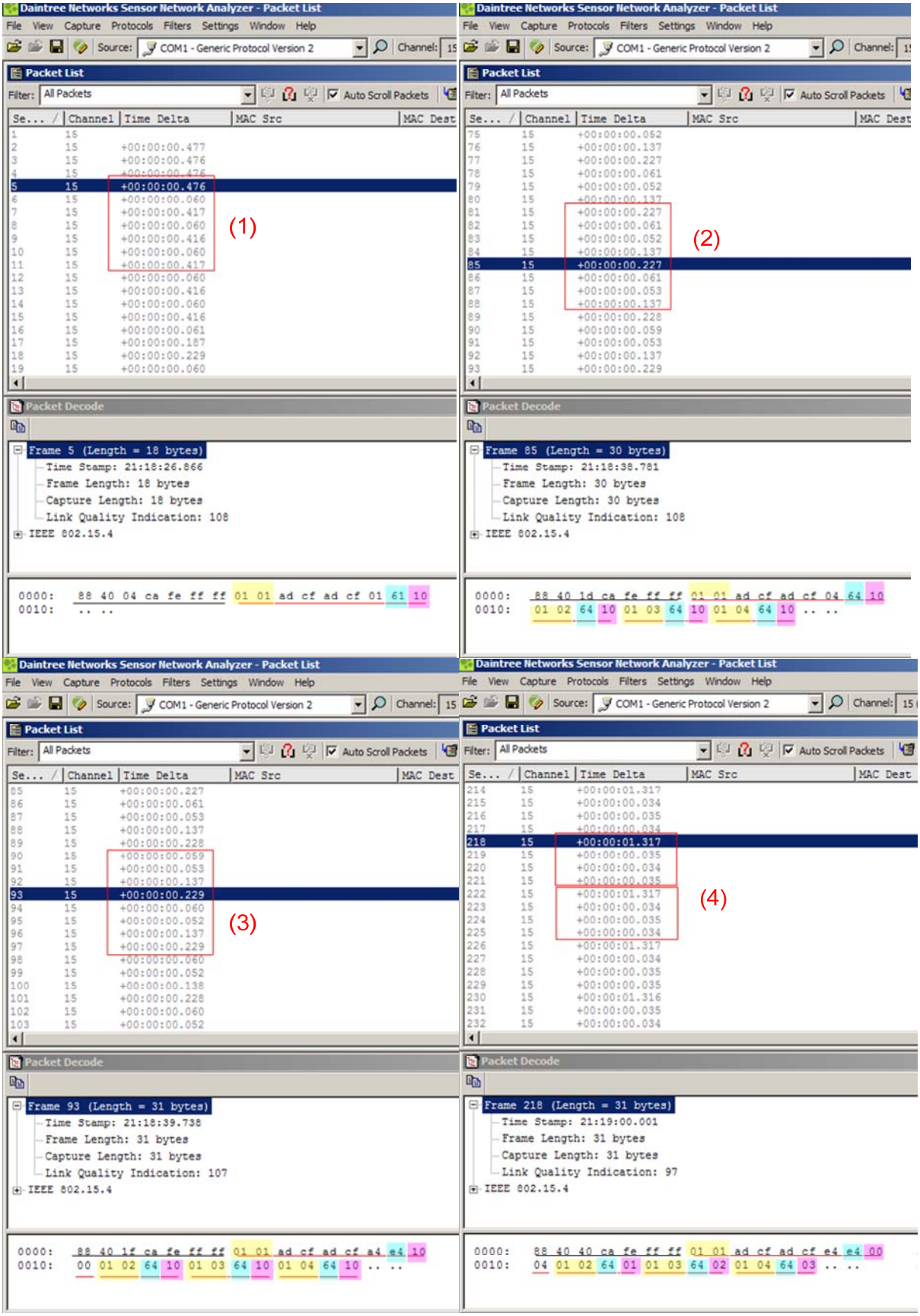

Figure 4.9 An example with SNA

To conclude this section 2, we explicitly illustrate the improvements for prototyping. Some implementation details such as synchronization mechanism are given. We demonstrate an example thanks to SNA, for better understanding ADCF and verifying this protocol. More prototype results will be shown in the next section 3.

3. Experimental scenarios and results

As the simulation, the cost of ADCF is our first challenge. Node failure and node join cases are studied in part 3.2. Then sensor nodes send simple application traffic by CFDS, in order to evaluate QoS capability of the protocol. At the end of this section, we will discuss this prototype work compared to simulation results.

As explained in 2.1.2, a superframe is fixed as 1.5 s by default in prototype. In this case, a beacon slot, like a data slot, lasts 31.25 ms. The targeted nodes are configured with maximum power, so typically the transmission range varies from 12 m to about 30 m. Transmission buffer is limited to five frames, including beacon frame and data frame. Each scenario is executed 20 times to make the results more accurate and reliable.

3.1. Protocol cost

Convergence Time and *Message Overhead* are always two interesting metrics in this experiment. They are related to four parameters of prototyping: node number (N), beacon interval (T_{cycle}), link confirmation parameters (T_{sample}) and maximum hop count (H_{max}).

In fact, convergence time contains *Convergence Time of Node* and *Network Convergence Time*. The node joins the network at different time in prototype, so *Convergence Time of Node* is defined as the average time duration of each node from its start to its selected collision-free beacon slot. Obviously, *Network Convergence Time* indicates the time duration from joining of the first node in the network to a well organized BOP of all the nodes. *Network Convergence Time* evidently includes a certain portion of the time which depends on when starting each node. Generally speaking, the deployment of a network in a house or a building like our laboratory takes tens of seconds up to several minutes according to the different topologies. *Network Convergence Time* gives the user of ADCF a reference time, meaning when the superframe could be well organized so that they can get a guaranteed medium access

service. *Message Overhead* here is the number of beacon frames sent in the network during *Network Convergence Time*.

3.1.1.Node number (N)

In this test, the full mesh networks of 4 nodes, 6 nodes and 8 nodes were built separately.

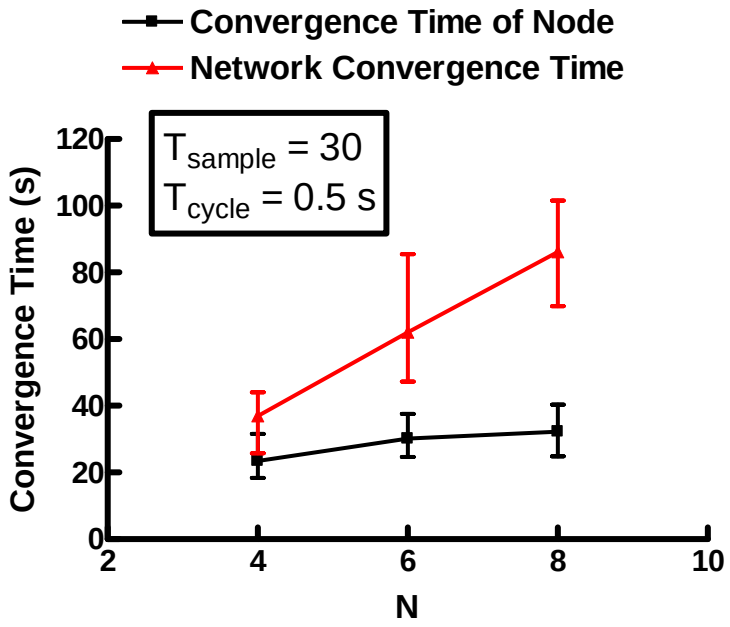

Figure 4.10 Convergence time vs. N

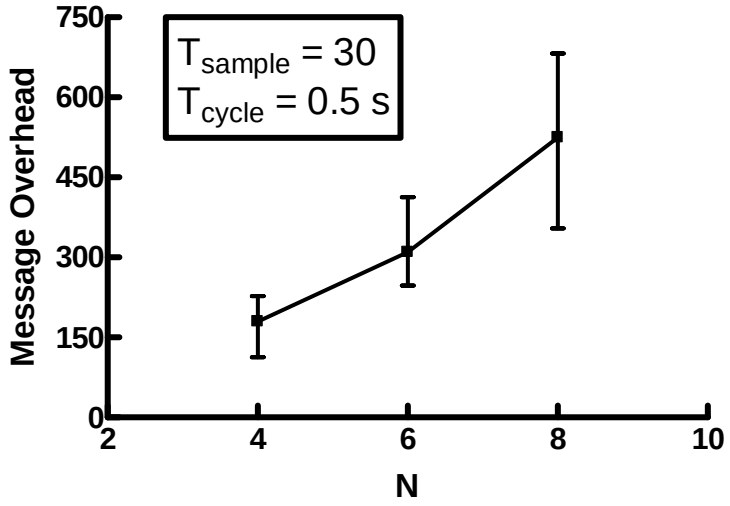

Figure 4.11 Message overhead vs. N

Figure 4.10 displays the node and network convergence times. We can see that both convergence times increase when the network scale is larger. On average, *Convergence Time of Node* keeps less than 35.60 s for all the three networks. *Network Convergence Time* sharply grows, with the bigger variances. For example, once the variance gets to 26.84 s for the network of 6 nodes. Generally, prototype tests show that we can successfully built a full mesh network of 8 nodes in 101.56 s. After this building phase, there is no more beacon collision.

Figure 4.11 indicates that *Message Overhead* increases with the augmentation of network scale. For the network of 8 nodes, 489 beacon frames are sent on average for synchronizing the network and avoiding the collisions. Also, variances become larger when there are more nodes in the network. We can see that the maximum variance, when N equals 8, is 43.15% deviation from the average value.

3.1.2.Beacon interval (T_{cycle})

We start 8 nodes one by one and modify beacon interval in this test. The network topology is still full mesh. Figure 4.12 shows that both *Convergence Time of Node* and *Network Convergence Time* grow with the increment of T_{cycle}. However, when T_{cycle} increases, *Message Overhead* becomes uncertain (Figure 4.13) as beacons are delivered less frequently.

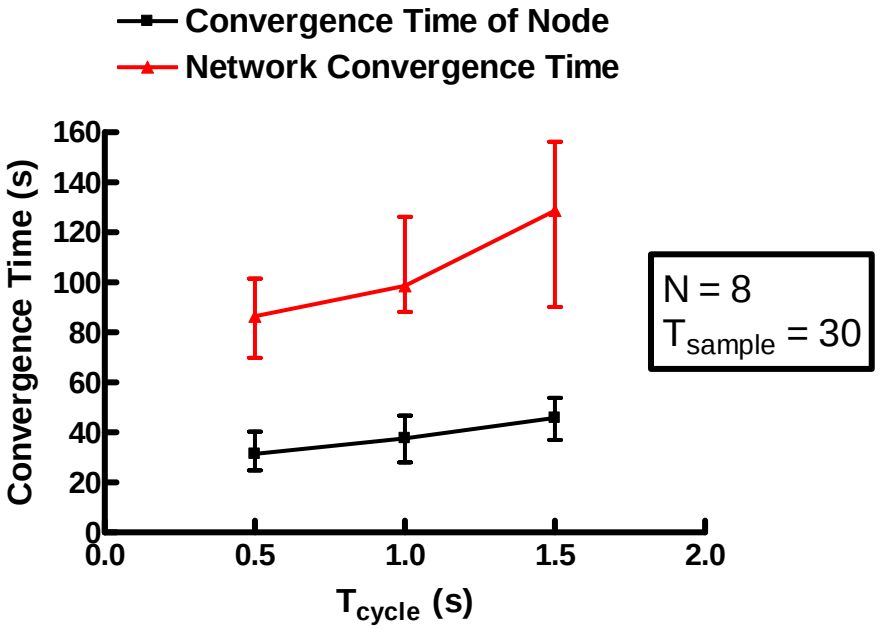

Figure 4.12 Convergence time vs. Tcycle

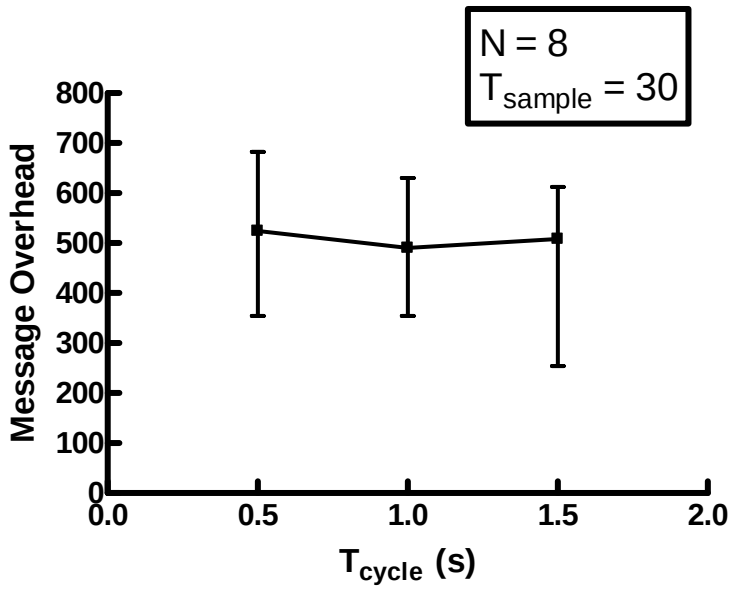

Figure 4.13 Message overhead vs. Tcycle

3.1.3.Link confirmation (T_{sample})

A network of 4 nodes and a network of 8 nodes were deployed in this test. Both of them are full mesh topology. Link confirmation mechanism is suitable for prototype as illustrated in part 2.1.1. In fact, T_{sample} here indicates a set of parameters. T_{sample} itself is defined as the number of T_{cycle} before starting to talk to a new node; *k*, *l*, *m*, *n* are parameters to alter the link state and has been fully explained in part 2.2.6 of chapter 2. When a link is confirmed rapidly in an ideal wireless environment, obviously T_{sample} should be smaller. On the other hand, a wireless environment with various interferences may take longer time to confirm a link state. T_{sample} also should be longer. Therefore, 3 series of parameters are chosen in the tests.

Table 4.1 Parameter sets

T_{sample}	k	l	m	n
10	2	2	4	2
20	4	4	8	4
30	8	8	16	8

Figure 4.14 shows that *Convergence Time of Node* grows about 48.57% when T_{sample} grows. But there is no huge difference between the network of 4 nodes and the network of 8 nodes. We can see that it takes only about 17.62 s to build the collision-free mesh network, at this moment T_{sample} equals 10, *k*, *l* and *n* are 2, and *m* is 4.

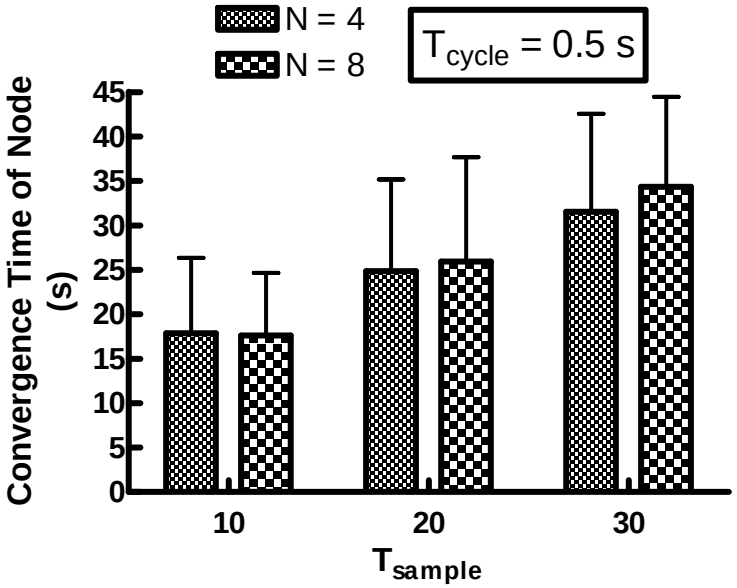

Figure 4.14 Convergence time vs. Tsample

For *Message Overhead*, as shown in Figure 4.15, the network of 8 nodes send beacons more than 3 times as much as that in the network of 4 nodes. When T_{sample} equals 10, about 118 beacons are delivered in the network of 4 nodes, on average 377 beacons are delivered in the network of 8 nodes, for organizing the converged network.

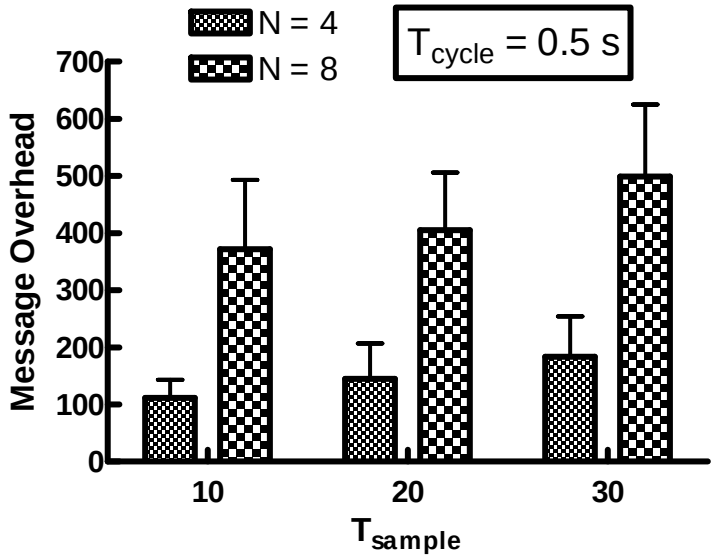

Figure 4.15 Message overhead vs. Tsample

3.1.4.Multi-hop network (H_{max})

To build the multi-hop network, 8 nodes are gradually deployed in our laboratory. In each scenario we start each node at different time with a random MAC address. Here we evaluate *Convergence Time of the Last Node*. It means the time duration from the start of the last node to a BOP well organized network. Here we use *Convergence Time of the Last Node* because the network may be rebuilt several times by SRP during this long start operation.

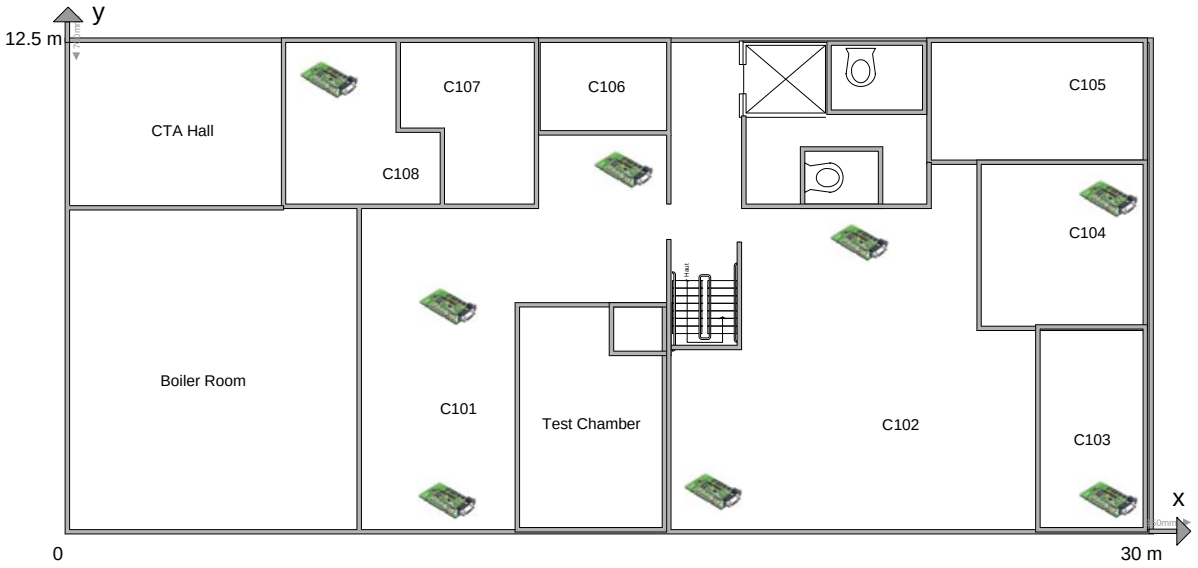

Figure 4.16 Network example 1

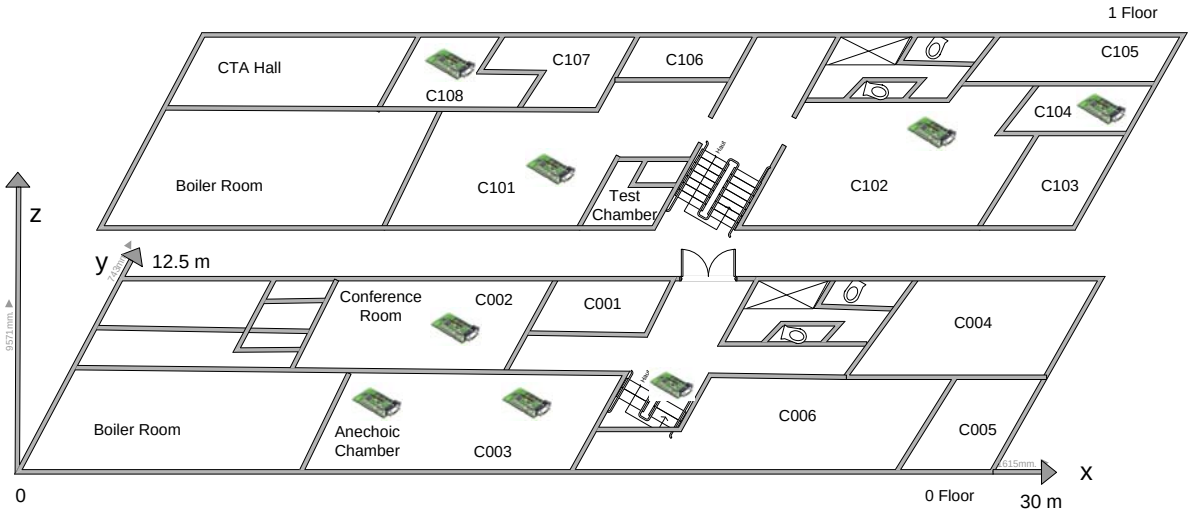

Figure 4.17 Network example 2

In fact, each multi-hop network is unique and has its own particularities. Figure 4.16 and Figure 4.17 just display two examples. In the first deployment, 8 nodes are disposed throughout the first floor of our laboratory. And in the second deployment, 4 nodes are

disposed in the first floor, 3 nodes are disposed in the ground floor, 1 node is disposed on stairway. Even though the fixed positions, network topology may change due to the wireless links. Based on the SNA results, H_{max} equals to 3 in the two network examples. Some interesting results are shown in the following table.

Table 4.2 Results for multi-hop network

		Network example 1	Network example 2
Convergence time of the last node	Average	123.99 s	97.74 s
	Minimum	55.04 s	42.63 s
	Maximum	252.30 s	173.25 s
Reuse 1 CFBS		19 times	11 times
Reuse 2 CFBS		Once	9 times
Asymmetric links		3/20	18/20

We can see that this *Convergence Time* takes about 1 to 2 minutes on average. This value is acceptable considering the total network lifetime which is several days or weeks. In a normal situation where the nodes are fixed, the network topology may not be reorganized frequently. The reuse of CFBS confirms the successful achievement of the organized superframe and the synchronized multi-hop network. In addition, this test presents that asymmetric link occurs frequently in the real environment, but ADCF can work well without disturbing of asymmetric link problem.

3.2.Node join and node failure

We can envisage that there are infinite topological change cases in reality. We consider that only one change occurs during the network operation, in order to better understand and simplify the verification work.

3.2.1.Node failure

Originally, 6 nodes self organize a full mesh network in this test. As shown in Figure 4.18, each red block contains a superframe. From the first superframe of (1) and (2), we can see that each node sends beacons in its CFBS and the beacon interval is about 35 ms. In Figure 4.18 (1), one node is suddenly stopped, so a free beacon slot presents in the next superframe. The beacon interval between the two nodes becomes 65 ms at this moment. Similarly, when we randomly stop two nodes of the network, as shown in Figure 4.18 (2), we can find two free slots in the next superframe. The rest of the network can work properly thanks to ADCF.

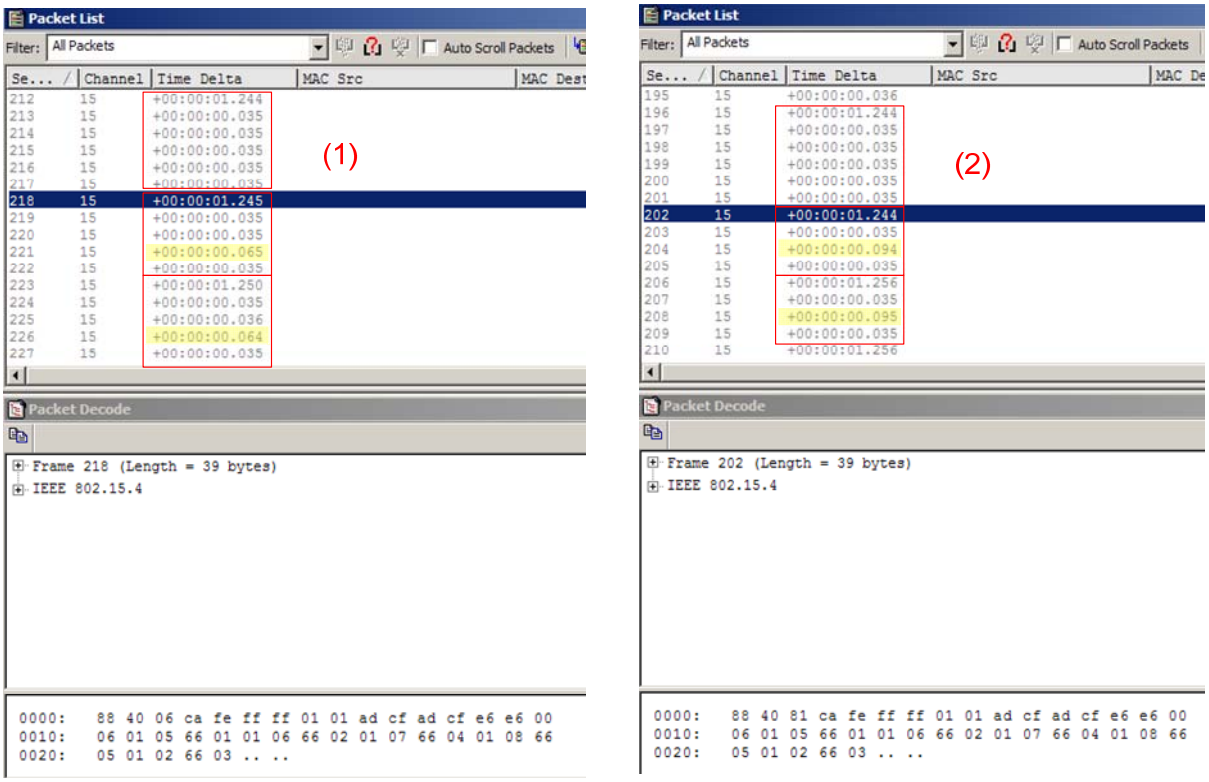

Figure 4.18 Node failure examples with SNA

3.2.2.Node join

In the beginning of this test, 4 nodes well organize a 2-hop mesh network without collision. We add new nodes one by one to the network. Specially, each new node is started when the original network has been in working stage. In this condition, BOP is not enough, so a network rebuilding process will be triggered each time.

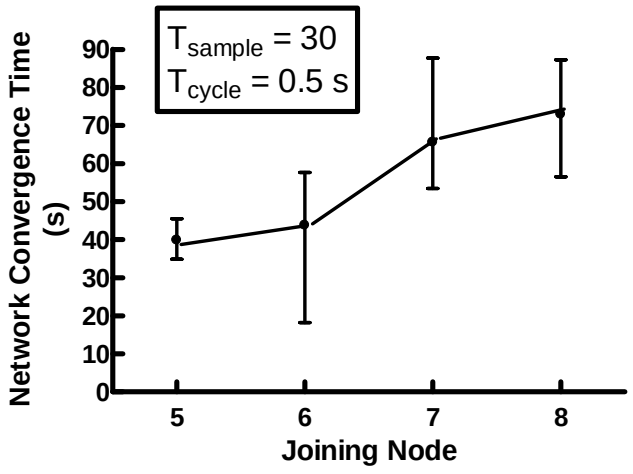

Figure 4.19 Convergence time vs. joining node

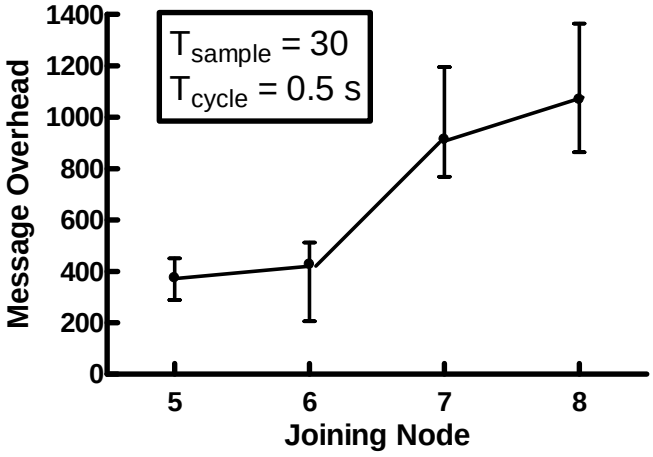

Figure 4.20 Message overhead vs. joining node

Figure 4.19 shows *Network Convergence Time* changes when the 5th, 6th, 7th and 8th node joins the network. Obviously, it takes more time to rebuild a larger scale network. At most, the network rebuilding process uses 87.50 s. Figure 4.20 shows the number of beacons delivered during the corresponding network convergence time. ADCF makes the network synchronized and access guaranteed each time after the rebuilding process execution.

3.3.QoS capability

There are 4 nodes in the network in this experiment. 2 of them are sources and 2 other nodes are recipients. Application traffic "test" will be periodically transmitted by CFDS mechanism. To generate the application traffic, a constant *SEND_DATA_TEST_PERIOD* is defined as *Application Packet Interarrival*, a random time is added to this constant to modify the packet reception time at MAC layer. In addition, the application traffic does not require acknowledgement.

The network topology could be full mesh, 2-hop topology or line topology in this experiment. We find that there is nearly no difference among them, it means the network topology do not impact the performance of CFDS mechanism.

3.3.1.Packet success ratio

Figure 4.21 shows *Packet Success Ratio* bars. 3 superframe durations are used in this test. Only when *Application Packet Interarrival* equals 0.5 s, transmission buffer for SD of 1.5 s or

3 s overflows as the frequent traffic load. Else, if transmission buffer is available, CFDS always performs a *Packet Success Ratio* of 100%.

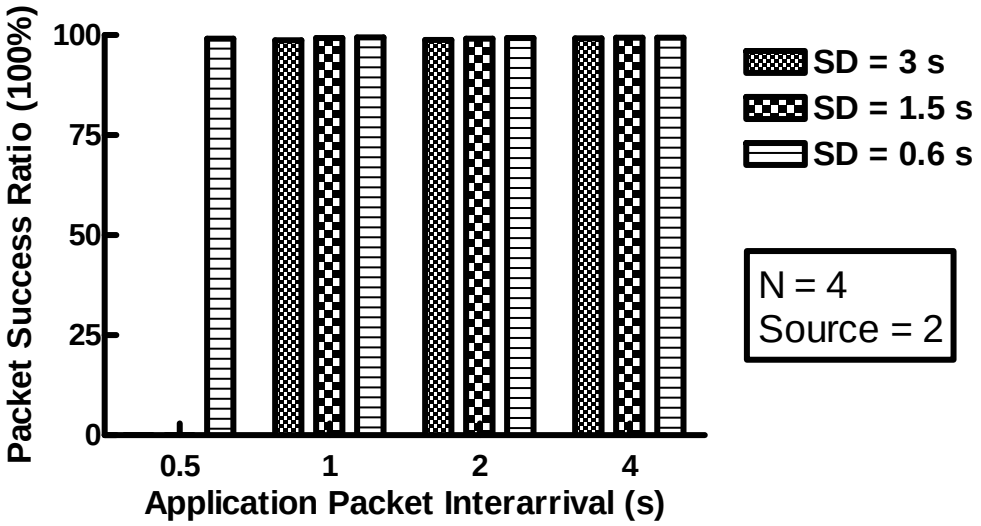

Figure 4.21 Packet success ratio

3.3.2.Delay

At MAC layer, *End-to-End Delay* here indicates the time duration from reception of a packet from application layer of the source node to reception of this packet from physical layer of the destination node.

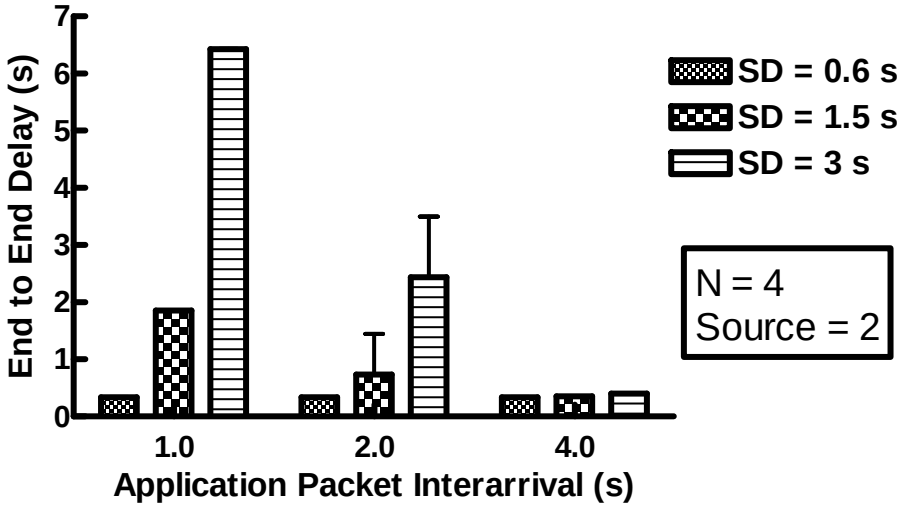

Figure 4.22 End-to-end delay

As shown in Figure 4.22, when *Application Packet Interarrival* equals 4 s, all the three test configurations have very small *End-to-End Delay*, about 0.37 s. When *Application Packet Interarrival* decreases, for SD of 1.5 s or 3 s, however *End-to-End Delay* largely grows as the accumulated traffic in buffer. For example, when SD is 1.5 s and *Application Packet Interarrival* is 2 s, some packets could be transmitted in the current superframe (about 0.37s), and some other packets have to be sent in the next superframe. Then when *Application Packet Interarrival* becomes 1 s, there is always one packet left in buffer and should be transmitted in the next superframe, so *End-to-End Delay* becomes more than 1.5 s.

We set SD 1.5 s and further investigate the composition of delay. As shown in Figure 4.23, no matter what *Application Packet Interarrival* the network with, *CFDS Reservation Time* keeps about 1.80 s. When *Application Packet Interarrival* is 1 s, *End-to-End Delay* takes about 1.84 s as the packet left in buffer and decreases with *Application Packet Interarrival* as seen previously.

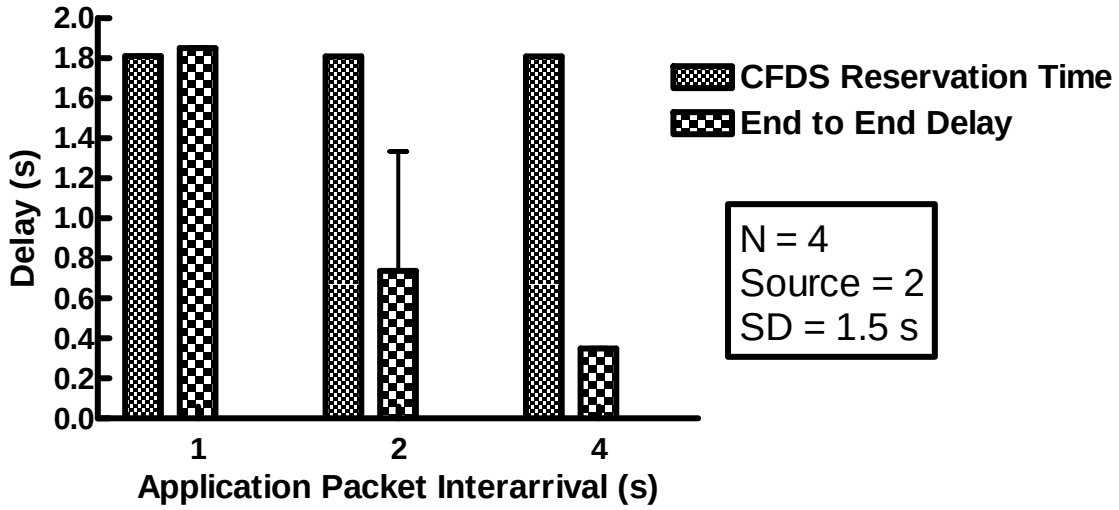

Figure 4.23 Delay composition

3.4.Discussion of prototype and simulation

Unfortunately, the prototype is not exactly the same as simulation. Some details had not been fully considered in simulation and some problems were identified in real tests. So the improvements such as synchronization mechanism must be added in prototype. Consequently, some parameters of prototype, e.g. beacon frame format and superframe duration, are also

altered. Even though the existing differences, we still try to analyse the results obtained by prototype and simulation.

First of all, both simulation and prototype confirm the feasibility of the proposed ADCF protocol. The sensor nodes can self organize and self repair in the mesh network. Importantly, each node can choose a collision-free beacon slot and require collision-free data slots.

Secondly, even if prototype results show the longer convergence times than that of simulation, these convergence times are acceptable (less than 82.54 s on average) for our application with a working stage of several months. Thanks to the distributed mesh architecture, both simulation and prototype certify that ADCF has the adaptability to wireless topological changes. Compared to simulation, the bigger cost of prototype is mainly due to two reasons. Link state confirmation mechanism takes more time to distinguish a neighbourhood which is ideally defined by distant in simulation but actually complex and volatile in real environment. On the other hand, the deployment of prototypes, meaning power-up of each node in the network, is a process which may contains network rebuilding as the mesh characteristics. Importantly, both simulation and prototype results confirm that the network takes advantage of mesh architecture and always can be converged within seconds or maximum two minutes in real tests.

At last, both from the simulation and the prototype, CFDS performances seem not to be affected by the parameters such as network scale and network density, etc. Also, we can see that buffer affects the CFDS performances both in simulation and prototype. Therefore, we compare the simulation with the prototype about CFDS performances under the condition of sufficient buffer space.

As shown in above tests, the superframe duration is set to 1.5 s and 3 s respectively in prototype. In simulation, the superframe duration lasts about 0.32 s when BO and SO are set to 3. When BO is 7 and SO is 4, the superframe duration is approximate 2.16 s. When buffer is available, both simulation and prototype show a *Packet Success Ratio* of 100%, as shown in Figure 4.24. Figure 4.25 shows *End-to-End Delay* comparison. In simulation, *End-to-End Delay* is 0.11 s if SD lasts about 0.32 s and is 0.97 s if SD lasts about 2.16 s. As the definition of *End-to-End Delay*, it contains a certain portion of the time which depends on the arrival time of application packets to the MAC layer. On the simulator, an OPNET built-in function is used to simulate the random arrival time of application packets. This function can generate the random number between 0 and SD. On the prototype, ADCF takes 0.35 s of *End-to-End*

Delay when SD is fixed as 1.5 s and about 0.4 s when SD is fixed as 3 s. Unfortunately, the prototype does not include a proper pseudo-random function: the random number is generated based on the local clock of the node. So the application traffic distribution model is changed. The arrival time of application packets is not affected by SD in prototype. In fact, this arrival time decides the time waiting for the corresponding date slot. In addition, *End-to-End Delay* of prototype contains the time within which the packets pass through different layers of the node. However, this packet processing time equals zero in simulation.

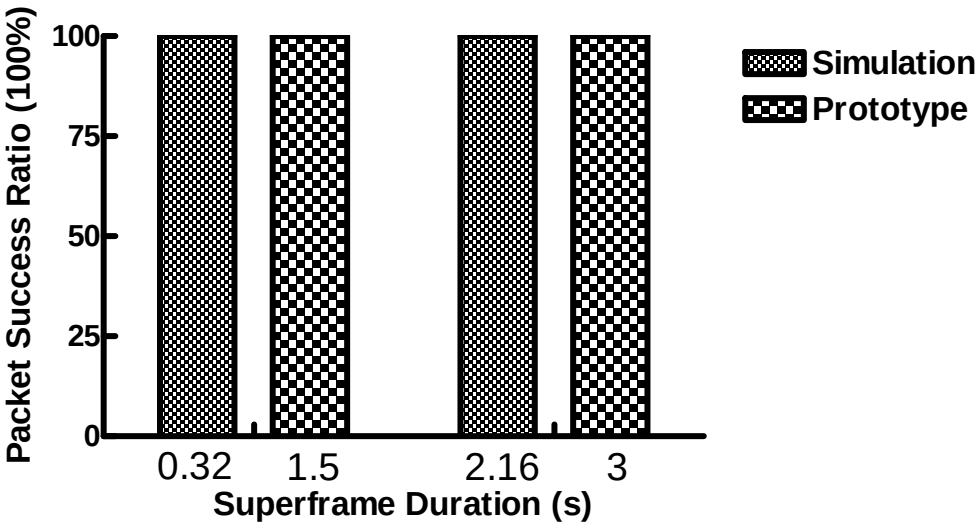

Figure 4.24 Packet success ratio in simulation and prototype

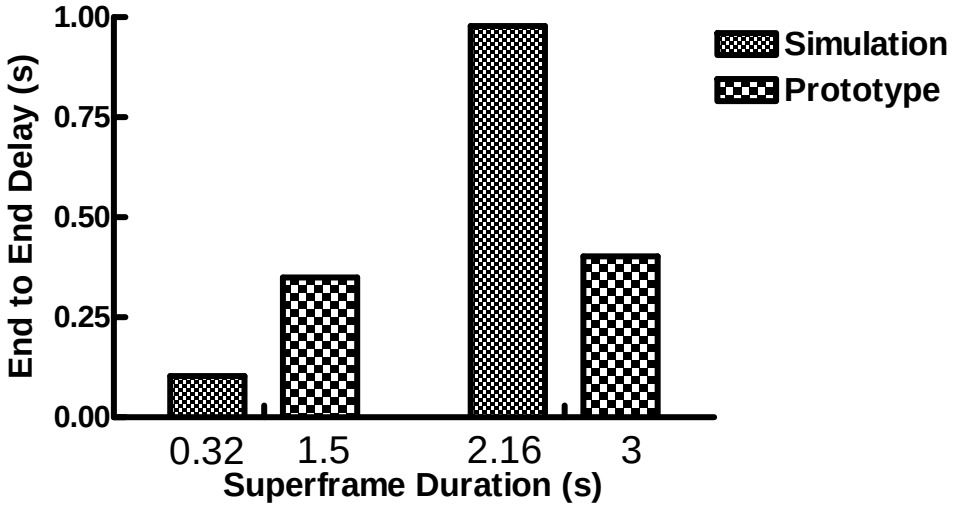

Figure 4.25 End-to-end delay in simulation and prototype

In this section, we have presented both the similarities and the differences between prototype and simulation and discussed about them.

3.5.Deployment of ADCF in smart home

Our smart home “Maison Intelligente” of Blagnac targets the elderly and the disabled living alone and provides them health and medico-social assistance at home. Many types of equipment, including wired products, mainly KNX-based, have been installed and are operational in this smart home. ADCF network fills some application gaps and provides an alternative wireless solution.

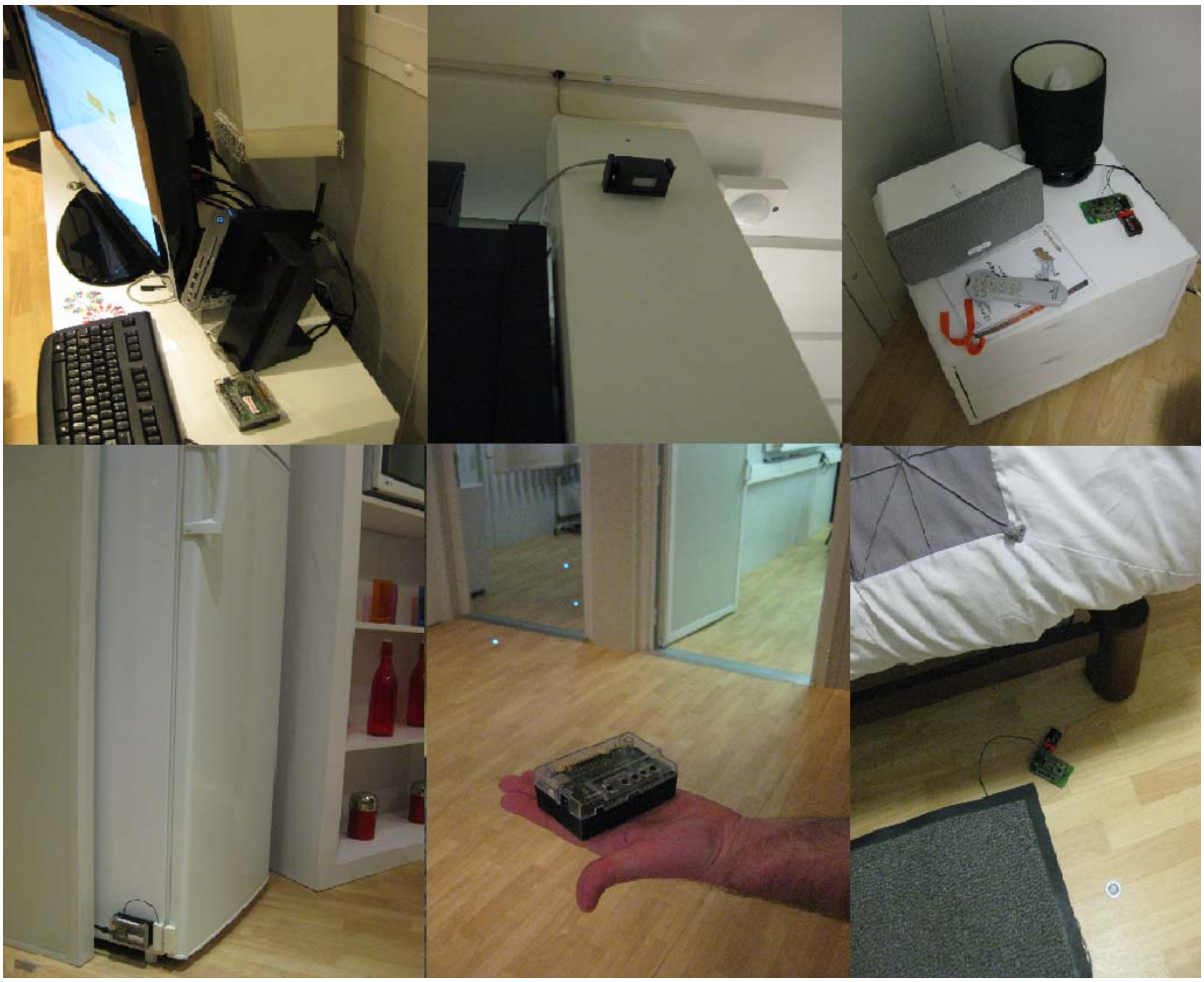

Figure 4.26 Deployment of ADCF in smart home

Six ADCF nodes were deployed in the smart home, as shown in Figure 4.26. From MAC point of view, these 6 nodes form a full mesh network (Figure 4.27). The nodes are connected with various sensors corresponding to the target of our application:

- Node @106 connecting with a magnetic sensor can monitor the open/closure of the refrigerator.
- Node @107 connecting with an infrared sensor can detect the motion of the person under his coverage area.
- Node @108 connecting with an emergency button is worn by the user to alert in case of fall, faintness, etc.
- Node @109 connecting with a sensor carpet placed near the bed can detect the getting up of the user.
- Node @10A connecting with lighting can be switch on/off by an order of another ADCF node.
- Finally node @105 connecting with screen is a sink for collecting and displaying all the network information. On the screen, we can see the superframe updating each second and log files from a web page.

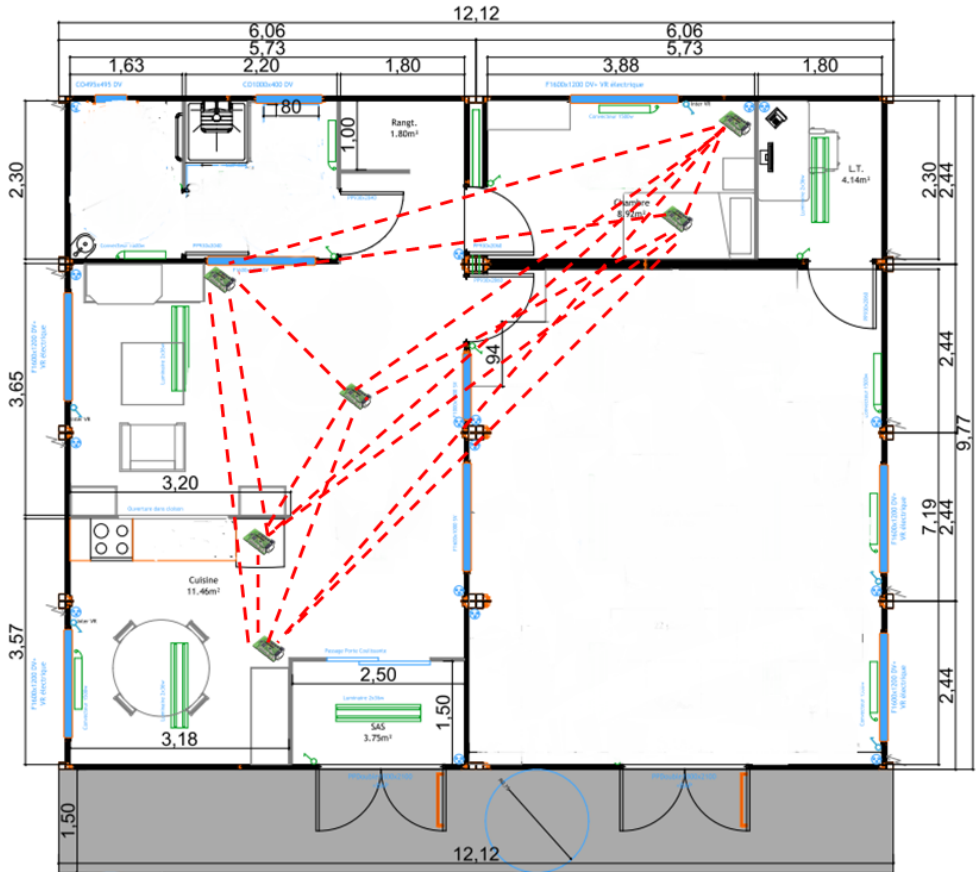

Figure 4.27 MAC layer – full mesh network

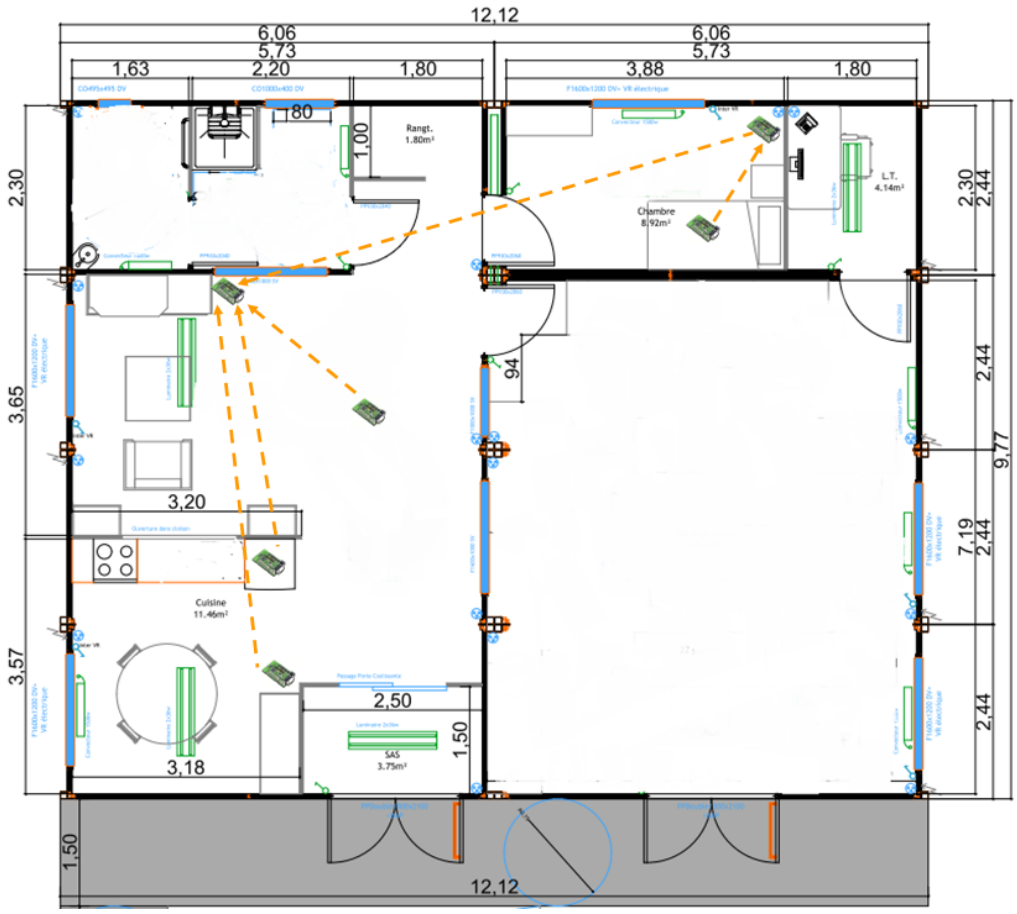

Figure 4.28 Application layer – data transmission

From application point of view, the exchanges are as following (Figure 4.28):

- @106 sends a message to @105 when the fridge is opened or closed,
- @107 sends a message to @105 when the person moves under the infrared sensor,
- @108 sends a message to @105 when the person presses the emergency button,
- @109 sends a message to @10A when the user arrives or leaves the carpet,
- @10A receives messages from @109 and switches on the lighting when the user is on the carpet. It switches off the lighting when the user leaves the carpet. When the lighting is switched on/off, @10A sends a message to @105 containing a return state of lighting.
- @105 receives messages from @106, @107, @108, @10A and logs actions in a journal. A voice synthesis software *eSpeake* [4.8] is also available and announces the reception of information from sensors (motion, carpet, etc.). Several scenarios

using the sensor information have been implemented; for example, when the sink detects the opening of the fridge for more than 10 seconds, the user can hear a voice alert.

In the initialization stage or rebuilding stage, frames are transmitted immediately without any medium access control precaution. In the working stage, frames are transmitted using CFDS.

Each node can join or leave the network freely and the rest of the network works properly. Generally, the network rebuilding time is less than 10s. Figure 4.29 shows a result of the above network deployment. At this moment, 6 nodes occupy 6 CFBS as they are all 1 hop neighbours. 5 CFDS are negotiated to transmit the corresponding application data. Thanks to a buzzer available on the nodes which is activated for each data frame reception, we can verify that the bounded time is verified as expected. It satisfies our application requirements. In addition, the superframe with its slot allocation is totally consistent with what we proposed in the chapter 2.

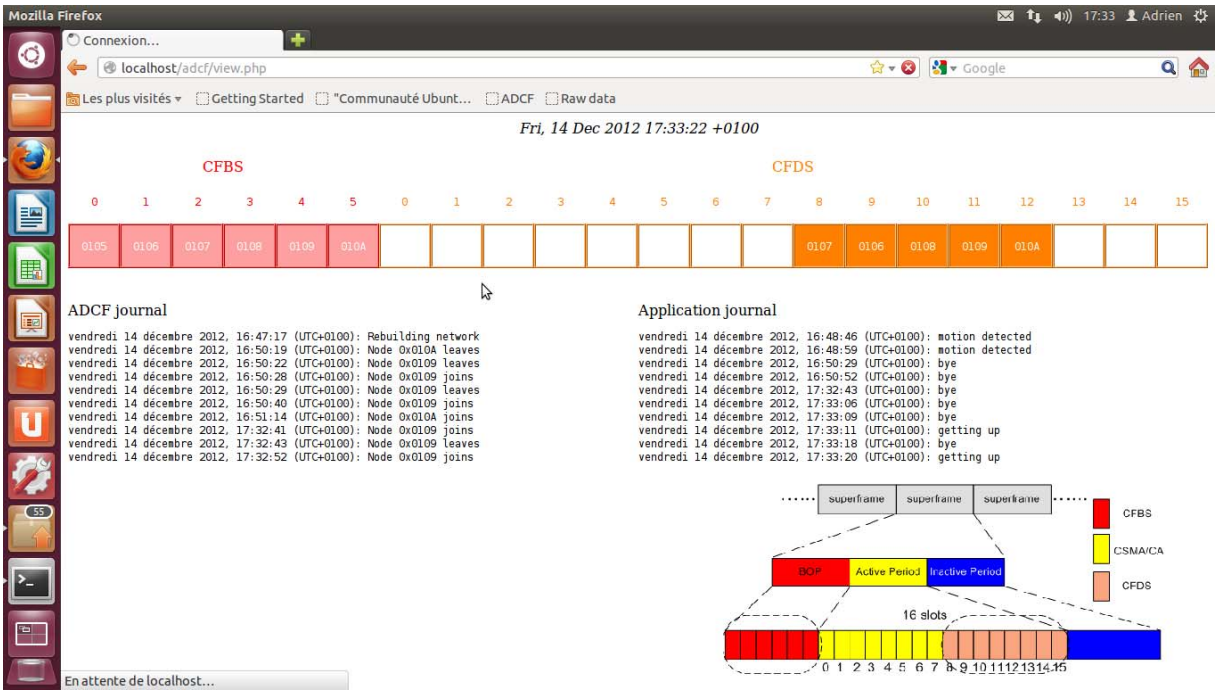

Figure 4.29 Example of a result

In this section, we have demonstrated the preliminary result obtained in the smart home. More applications are being considered and so more ADCF nodes will be deployed in the future.

4. Conclusion

In this chapter, we presented the prototype implementation of ADCF. A powerful platform WiNo and its useful features were presented. Two types of nodes, 13192-SARD and 1321x-SRB, were available for prototyping. We also explained the methods of performance analysis by using the tools such as console and SNA. Afterwards, we deployed ADCF nodes in real environment and some measurement results were given.

To conclude the prototype results, we primarily stress the possibility of ADCF implementation. Even though many difficulties such as very limited memory resource, unreliable wireless link states and changes and deploying multi-hop networks in real environment, ADCF works correctly. We can build a mesh network of 8 nodes with an organized superframe; each node can choose a collision-free slot to send beacons; we can start or stop the nodes as we want; via the collision-free beacon exchanges, the nodes can negotiate collision-free data slot; at last, the nodes can send application data without packet loss. That's very encouraging.

We analyze the differences between prototype and simulation. The results obtained by prototype are also encouraging. The protocol costs are admissible. For example, 8 nodes send 489 beacons in 82.54 s for a collision-free and QoS-guaranteed network. Once the nodes are in working stage, CFDS can provide the packet delivery in 0.37 s without loss, which is even better than simulation as the different application models.

At last, preliminary result evaluated in our smart home was shown. First of all, we can quickly install ADCF nodes, meaning several minutes. Compared to wired infrastructures, this is a great advantage of ADCF and very interesting to the application of home monitoring for just several weeks or months. In the situation studied, this network always can converge in about 10 s and then provide a guaranteed solution. We can hear the reception of an application data in 1 to 2 s. Therefore, we confirm that ADCF is an effective work to respond our application requirements and is an interesting alternative to home automation buses.

Reference

[4.1] WiNo, http://lab.iut-blagnac.fr/dokuwiki/doku.php/projets/wsnemu/doc

[4.2] A. Van Den Bossche, T. Val, R. Dalce, "SISP: A Lightweight Synchronization Protocol for Wireless Sensor Networks", 16th conference on Emerging Technologies and Factory Automation (ETFA 11), pp. 1-4, September 2011

[4.3] Freescale, "Simple Media Access Controller Demonstration Applications", http://www.freescale.com/, September 2011

[4.4] Freescale, "Sensor Applications Reference Design", http://www.freescale.com/, June 2004

[4.5] CodeWarrior Development Tools, http://www.freescale.com/webapp/sps/site/homepage.jsp?code=CW_HOME

[4.6] 13212 Evaluation Kits, http://www.freescale.com/, June 2007

[4.7] Sensor Network Analyzer, http://www.daintree.net/sna/sna.php

[4.8] eSpeak, http://espeak.sourceforge.net/

Conclusion and Perspectives

General conclusion

This manuscript presented an Adaptive and Distributed Collision Free MAC protocol based on the IEEE 802.15.4 standard. This protocol was designed to build and to maintain a wireless mesh sensor network providing QoS-guaranteed medium access and energy saving solution for home monitoring application. The work was organized in three phases: protocol presentation, implementation and results of the simulation and the prototype.

In Chapter 1, we summarized common architectures for habitat monitoring networks and focused on WSN. The objective was to provide an alternative wireless solution with the advantage of convenient installation and flexible deployment for several weeks or months, compared to wired technologies such as KNX, HART, etc. which impose a costly network deployment. Considering wired buses like KNX, the common medium enables direct end-to-end communications between nodes. However, the current wireless technologies such as ZigBee sometimes disable direct communications because of the tree topology restriction. In this case data must be routed from the source node to a supernode and then the supernode sends data to the destination node even though the two nodes are in the transmission range of each other. In addition, while IEEE 802.15.4 tree topology enables energy savings on routers, ZigBee does not use this mode so routers are always active. Other wireless technologies such as 6LoWPAN and Z-Wave do not consider QoS guaranteed medium access control method. Therefore, a novel communication protocol was expected to be energy saving, flexible, robust, and to have QoS capacities at the same time. We only worked at MAC layer to handle the above challenges and solve the problems such as beacon collisions, changing link states and multi-hop synchronization in a mesh network. Upper layers such as routing layer will be considered in perspective.

In Chapter 2, our proposition called ADCF was fully presented. This original MAC protocol is based on the IEEE 802.15.4 2.4 GHz DSSS physical layer and classical superframe structure. Indeed the standard supports very interesting mechanisms for QoS and energy saving and proposes many available commercial products. The focus was on adapting IEEE 802.15.4 to the mesh network in which all the nodes could sleep for energy saving and

may fail without disturbing the rest of the network. In general, ADCF includes 2 stages: in initialization stage, the nodes send beacons by unslotted CSMA/CA to build a mesh network. The network building costs, convergence time and message overhead, are related to the network parameters such as N, D_{max}, H_{max} which were fully studied in Chapter 2 and 3. In working stage, based on the IEEE 802.15.4 superframe structure, ADCF divides time into three parts: in BOP which is organized by CFBS and dynamically change according to the wireless topological changes, the nodes far away than 2-hop can reuse the same timeslots so that beacon collisions could be avoided. In active period, CFDS was proposed to enable the nodes to negotiate dedicated data slots in the mesh topology. Thanks to CFDS, application data could be transmitted in a bounded time without packet loss. At last, the nodes go to sleep mode in inactive period. So our contribution was CFBS and CFDS mechanisms. In order to achieve these protocol functions, ADCF was divided into a set of protocols/algorithms: BEP, SPA, ISP, BSAP, DSAP and SRP. Each of them was fully detailed and a theoretical study was given for evaluating the protocol cost in the worst case. At the end of this chapter, service primitives and related parameters used in the ADCF node were explained. The description of these primitives will allow the implementation of ADCF, for example on a network simulator software or a real node. An efficient multi-hop mesh network could be built and maintained with these ADCF nodes.

In Chapter 3, we presented the simulation of ADCF. OPNET network simulator was chosen as its high-quality programming, user-friendly GUI and data processing capability. Most importantly, OPNET contains a complete IEEE 802.15.4 implementation which makes the comparison of ADCF and IEEE 802.15.4 possible. After the presentation of simulation model and parameters, many experimental scenarios were simulated and some interesting results were shown. We discussed the simulation results from 3 parts: QoS, energy saving, flexibility and robustness, as required by our application. The simulation conclusions are:

- ADCF satisfies our application request of delivering QoS traffic. When buffers are available, CFDS allows the data delivery without packet loss. End-to-end delay depends on the superframe structure and our simulation results confirm that ADCF is never worse than 802.15.4 for delivery of QoS traffic. In some cases such as multi-hop network, ADCF can be even better than 802.15.4 thanks to the availability of shorter paths within the mesh topology.

- The costs (energy, protocol) of ADCF are acceptable. We can build a mesh network of 30 nodes in 25 s and with little overhead. Obviously, the cost of ADCF also includes its energy consumption. Simulation result shows that ADCF consumes less energy, about 37%, than 802.15.4. We can further improve the performances such as convergence time and end-to-end delay at the price of energy consumption. So the trade-offs should be made according to specific application environments.

- For flexibility and robustness, a lot of cases were considered and we given the worst cases, network rebuilding, as representative examples. Compared with star or tree topology, thanks to ADCF, the network works properly even though there are some failure nodes. Also, new nodes could join the network freely, increasing the flexibility of the network. In some cases such as a multi-hop network with free CFBS, new nodes can perfectly insert the superframe and send beacons without collision.

In addition, the simulation results show that both network scale and neighbor density have no influence on QoS traffic which is sent by CFDS mechanism. QoS traffic could be sent without packet loss, demonstrating the stable performance of ADCF, if the buffers are available. Therefore, the current simulation work and simulation results verify the advantages of ADCF.

In Chapter 4, we presented the prototype implementation of ADCF. By using platform WiNo and sensor application boards such as 13192-SARD and 1321x-SRB, we achieved the implementation of our proposition and resolved many difficulties in real environment such as deploying multi-hop networks. However, some details had not been fully considered in simulation and some problems were identified in real environment. So the improvements such as synchronization mechanism and link confirmation mechanism must be added in prototype. We discussed both similarities and differences between prototype and simulation and analyzed the reasons of these differences. Thanks to the tools such as node console and SNA protocol analyser, we obtained the following results through practical measurements: a mesh network of 8 nodes could successfully be built with an organized superframe where each node can choose a collision-free beacon slot. The protocol costs are also acceptable; for a network with 8 nodes, convergence time is 82.54 s on average and 489 beacons are sent during network initialization stage. Nodes can freely join or leave the network without disturbing the

global operation of the system. Via CFBS, the nodes can successfully negotiate collision-free data slots. Once the nodes are in working stage, CFDS can provide the data delivery in 0.37 s on average without packet loss, which is even better than simulation. So the prototype results are very encouraging. At last, six ADCF nodes were deployed in the "smart home of Blagnac". The nodes were connected with various sensors such as magnetic sensor, infrared sensor, emergency button and carpet sensor. Several interesting scenarios using the sensor information have been implemented to monitor the activities of the user. In the situation studied, the network can always converge in about 10 s with the suitable parameters configuration and, thank to the buzzers, we can hear the reception of an application data on the sink in 2 s at most, which confirm that ADCF is verified in real environment as expected.

The above study proves that ADCF presents some good performances and satisfies our needs that are, basically, to replace a wired bus in home monitoring application. Certainly ADCF is not perfectly suitable to all situations in the scope of monitoring domain. For example, some high-rate and heavy-traffics could not be transmitted by an ADCF network with correct performance. Secondly, usually the user walking at home with ADCF node does not cause many topological changes as the limited home space and it is perfectly acceptable. However, high-mobility nodes which lead to frequent topological changes and network rebuilding must use another technology. So another lack of ADCF is the mobility of the routers. Finally, the large scale network, hundreds to thousands of nodes, must use multi-tier architecture, instead of mesh architecture. In short, home monitoring application requires a complex system integrating with many different technologies. ADCF only works at MAC layer and provides an alternative wireless solution among them. Many issues are still open for the future.

Perspectives

Firstly, topology control algorithm has important influence on the mesh network. A topology control algorithm consists in optimizing the topology by modifying the transmission power or even the position of the nodes when the network is deployed or in working stage. A favorable topology control algorithm can reduce energy consumption and improve network capacity, while maintaining network connectivity. In fact, we have defined the network parameters such as N, D_{max}, H_{max}. Both simulation and prototype results show their impacts on the protocol performances. However, the theoretical study and a ensuing topology control

algorithm is expected in order to better take advantage of mesh characteristics such as link redundancy, optimize the network configuration management and finally help us to deploy the suitable network for each specific application scenario.

Secondly, the ADCF prototype implementation is not exactly the same as that of simulation. Currently, the available targeted boards can not sleep and can not estimate the consumed energy as the hardware design limitations. Some simple improvements of ADCF such as sleeping in the unused or non possessed CFDS have not been tested due to the lack of possibility of the hardware. So some other testbeds should be considered to achieve the whole implementation and to evaluate energy consumption performance. In the same way, the benchmarking of ADCF on testbed platforms such as SensLab should be considered.

As explained in chapter 4, ADCF prototype lacks synchronization protocol. One of the assumptions was that ADCF nodes were considered synchronized. In simulation, the nodes were synchronized by the network simulator, which was strong challenge in the implementation of the prototype: Approximately each 4 hours, nodes lose their clock synchronization in prototype due to an overflow of the 32-bits clock counter. Hence a clock synchronization protocol such as SISP is need. An interesting perspective is the integration of SISP in ADCF.

A simple and efficient routing protocol is being considered. ADCF focuses on MAC layer but offers a gainful basis to the upper layers. For example, ADCF node has already maintained a 2-hop neighbor table which can be very useful information to the network layer. Then ADCF implements CFDS negotiation between 1-hop neighbors. The routing protocol is expected to implement multi-hop CFDS negotiation. If the allocated CFDSs are sequentially connected in the superframe along the route from the source node to the destination node, we can see that the end-to-end delay will be minimized. A very interesting perspective is the optimization of CFDS scheduling along the whole route, either for reduce the end-to-end delay, or save energy grouping the free CFDSs together. Therefore, a cross-layer design taking advantage of ADCF will be implemented for reserving multi-hop CFDS and routing packets as soon as possible.

In addition, ADCF network is used in a smart home in which many solutions including wired and wireless technologies may also be used. We should make ADCF compatible with theses networking technologies, for example, body area network with physiological sensors or high-mobility sensors and home automation network using technologies such as Bluetooth,

WiFi, RF, etc. Sometimes, ADCF network also might need to interact to other networks such as mobile phone network when the user is outside, hospital network or Internet, in order to play a better and more important role in the system of home monitoring.

General Bibliography

[1.1] M. Chan, D. Estève, C. Escriba, and E. Campo, “A Review of Smart Homes—Present State and Future Challenges”, Computer Methods and Programs in Biomedicine, Volume 91, No. 1, pp. 55-81, July 2008

[1.2] S. Nourizadeh, C. Deroussent, Y.Q. Song and J.P. Thomesse, “A Distributed Elderly healthcare System”, MobiHealth 2009, http://hal.inria.fr/inria-00431202

[1.3] S. Kiefer, “Implementation of a Telematic Homecare Platform in Cooperative Health Care Provider Networks”, www.topcare-network.com

[1.4] V. Rialle, F. Duchene, N. Noury, L. Bajolle and J. Demongeot, “Health Smart Home: Information Technology for Patients at Home”, Telemedicine Journal and e-Health, Volume 8, Issue 4, pp. 395-409, July 2004

[1.5] A. Anfosso, S. Rebaudo, “Gérontechnologies et Contrôle de L'environnement au Service du Maintien à Domicile: le projet GERHOME”, Gérontologie et Société, No.136, pp. 119-131, 2011

[1.6] Y. Charlon, W. Bourennane, E. Campo, “Mise en Oeuvre d’une Plateforme de Suivi de L’actimétrie Associée à un Système D’identification”, SMS 2010

[1.7] Y. Zatout, E. Campo and J.F. Llibre, “WSN-HM: Energy-Efficient Wireless Sensor Network for Home Monitoring”, 5th International Conference on Intelligent Sensor, Sensor Networks and Information Processing (ISSNIP), pp. 367-372, December 2009

[1.8] Wi-Fi Alliance, “IEEE 802.11 Specification”, www.wi-fi.org

[1.9] Bluetooth Special Interest Group, “IEEE 802.15.1” www.bluetooth.org

[1.10] J.R. Boulanger, C. Deroussent, “Preliminary Based Service Evaluation for Elderly People and Healthcare Professionals in Residential Home Care Units”, 2nd International Conference on Digital Society (ICDS), pp. 93-101, September 2008

[1.11] I.F. Akyildiz, T. Melodia, K.R. Chowdhury, “A survey on wireless multimedia sensor networks”, International Journal of Computer and Telecommunications Networking, Volume 51, Issue 4, pp. 921-960, March 2007

[1.12] I.F. Akyildiz, W.L. Su, Y. Sankarasubramaniam, E. Cayirci, “Wireless Sensor Networks: A Survey”, IEEE Computer Networks, Volume 40, Issue 8, pp. 393-422, August 2002

[1.13] H.M. Ammari and S.K. Das, "Integrated Coverage and Connectivity in Wireless Sensor Networks", IEEE Transactions on Computers, Volume 57, Issue 10, pp.1423-1434, October 2008

[1.14] G.Z. Zheng, Q.M. Liu, "A Survey on the Topology of Wireless Sensor Networks Based on Small World Network Model", 2nd International Conference on Future Computer and Communication (ICFCC), pp. 67-71, May 2010

[1.15] A. Vogel, B. Kerherve, G.V. Bochmann, J. Gecsei, "Distributed Multimedia and QoS: A Survey", IEEE Multimedia Journal, Volume 2, Issue 2, pp. 10-19, August 2002

[1.16] H. Fattah, C. Leung, "An Overview of Scheduling Algorithms in Wireless Multimedia Networks", IEEE Wireless Communications, Volume 9, Issue 5, pp. 76-83, October 2002

[1.17] M.A. Yigitel, O.D. Incel, C. Ersoy, "QoS-Aware MAC Protocols for Wireless Sensor Networks: A Survey", International Journal of Computer and Telecommunications Networking, Volume 55, Issue 8, June 2011

[1.18] S. Mahfoudh, P. Minet, "Maximization of Energy Efficiency in Wireless Ad hoc and Sensor Networks with SERENA", Mobile Information Systems-Advances in Wireless Networks, Volume 5, Issue 1, pp.32-53, Juanuary 2009

[1.19] W. Allcock, J. Bresnahan, K. Kettimuthu, J. Link, "The Globus Extensible Input/Output System (XIO): A Protocol Independent IO System for the Grid", 19th International Parallel and Distributed Processing Symposium (IPDPS), pp. 8-16, April 2005

[1.20] J. Laudon, D. Lenoski, "System Overview of the SGI Origin 200/2000 Product Line", IEEE Compcon Proceedings, pp. 150-156, February 1997

[1.21] IEEE-SA Standards Board, "IEEE 802.3 Local Area Network Protocols", IEEE Standard for Information Technology

[1.22] KNX Association, "KNX Specification" http://knx.org

[1.23] Wolfgang Kohler, "Simulation of a KNX Network with EIBsec Protocol Extensions", Chapter 2, April 2010

[1.24] HART communication foundation, "WirelessHART Technical Data Sheet", http://www.hartcomm.org

[1.25] IEEE-SA Standards Board, "IEEE 802.15.4 Standard (2006) Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)", IEEE Standard for Information Technology

[1.26] IEEE-SA Standards Board, "802.15.4a (2007)", IEEE Standard for Information Technology

[1.27] Crossbow, "MicaZ Datasheet", www.xbow.com

[1.28] Crossbow, “TelosB Datasheet”, www.xbow.com

[1.29] Crossbow, “Imote2 Datasheet”, www.xbow.com

[1.30] http://www.ieee802.org/15/pub/TG6.html

[1.31] K.S. Kwak, S. Ullah, and N. Ullah, “An Overview of IEEE 802.15.6 Standard”, 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL), February 2011

[1.32] IEEE-SA Standards Board, “IEEE P802.15.6/D06: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs) Used in or around A Body, IEEE Draft Standard for Information Technology

[1.33] S.C. Ergen, “ZigBee/IEEE 802.15.4 Summary” http://citeseerx.ist.psu.edu

[1.34] ZigBee Alliance, “ZigBee Specification”, Document 053474r17, http://zigbee.org

[1.35] Freescale, “13192 Sensor Applications Reference Design”, www.freescale.com

[1.36] IEEE-SA Standards Board, “Part 15.5: Mesh Topology Capability in Wireless Personal Area Networks (WPANs)”, IEEE Recommended Practice for Information Technology

[1.37] M.J. Lee, R. Zhang, J.L. Zheng, G.S. Ahn, C.H. Zhu, T.R. Park, S.R. Cho, C.S. Shin, J.S. Ryu, “IEEE 802.15.5 WPAN Mesh Standard-Low Rate Part: Meshing the Wireless Sensor Networks” IEEE Journal on Selected Areas in Communications, Volume 28, Issue 7, pp. 973-983, Septembre 2010

[1.38] C.H. Zhu, J.L. Zheng, C. Ngo, T. Park, R. Zhang, M. Lee, “Low-Rate WPAN Mesh Network – An Enabling Technology for Ubiquitous Networks”, IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6, April 2009

[1.39] IETF 6LoWPAN Working Group, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks,” IETF RFC 4944

[1.40] K. Kim, S. Park, G. Montenegro, S. Yoo, N. Kushalnagar, “6LoWPAN Ad Hoc On-Demand Distance Vector Routing (LOAD)”, draft-daniel-6lowpan-load-adhoc-routing-03

[1.41] K. Kim, S. Park, I. Chakeres, C. Perkins, “Dynamic MANET On-demand for 6LoWPAN (DYMO-low) Routing”, draftmontenegro-6lowpan-dymo-low-routing-03

[1.42] K. Kim, S. Yoo, S. Park, J. Lee, “Hierarchical Routing over 6LoWPAN (HiLow)”, draft-deniel-6lowpanhilow-hierarchical-routing-00

[1.43] IETF 6LoWPAN Working Group, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals”, IETF RFC 4919

[1.44] http://www.contiki-os.org/

[1.45] Arch Rock, “IPsensor Node”, www.archrock.com

[1.46] Jennic, “Jennic’s 6LoWPAN”, www.jennic.com

[1.47] Z-Wave Alliance, http://www.z-wavealliance.org/

[1.48] Niels Thybo Johansen, “Software Design Specification: Z-Wave Protocol Overview”, April 2006

[1.49] P. Soldati, H.B. Zhang and M. Johansson, “Deadline-Constrained Transmission Scheduling and Data Evacuation in WirelessHART Networks”, Technical Report, https://eeweb01.ee.kth.se

[1.50] A. Lehto, “WirelessHART Smart Wireless Solutions”, www.hartcomm.org

[1.51] Dust Networks, “Technical Overview of Time Synchronized Mesh Protocol (TSMP)”, www.dustnetworks.com

[1.52] Siemens, “Sitrans Data Sheet”, www.siemens.com

[1.53] Emerson, “Fisher Manual”, www.emerson.com

[1.54] Dust Networks, “SmartMesh WirelessHART”, www.dustnetworks.com

[1.55] Task Group 15.4b, http://grouper.ieee.org/groups/802/15/pub/TG4.html

[1.56] A. Koubaa, A. Cunha, M. Alves, “A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks”, 19th Euromicro Conference on Real-Time Systems (ECRTS), pp. 125-135, July 2007

[1.57] A. Koubâa, M. Alves, M. Attia, A. Van Nieuwenhuyse, “Collision-Free Beacon Scheduling Mechanisms for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks”, 7th International Workshop on Applications and Services in Wireless Networks (ASWN), May 2007

[1.58] K. Alagha, G. Chalhoub, A. Guitton, E. Livolant, S. Mahfoudh, P. Minet, M. Misson, J. Rahme, T. Val, A. van den Bossche, “Cross-Layering in An Industrial Wireless Sensor Network: Case Study of OCARI”, Journal of Networks, Volume 4, Issue 6, pp. 411-420, August 2009

[1.59] T. Dang, C. Devic, E. Livolant, A. Van Den Bossche, T. Val, “OCARI: Optimization of Communication for Ad Hoc Reliable Industrial Networks”, 6th IEEE International Conference on Industrial Informatics (INDIN), pp. 688-693, July 2008

[1.60] E. Livolant, A. Van Den Bossche, T. Val, “MAC Specificaitons for A WPAN Allowing Both Energy Saving and Guaranteed Delay Part B: Optimization of the Inter-Star Exchanges for MaCARI”, Computer Science-Wireless Sensor and Actor Networks, Volume 264, pp. 233-244

[1.61] P.S. Muthukumaran, R. Alberola, R. Spinar and D. Pesch, “MeshMAC: Enabling Mesh Networking over IEEE802.15.4 through Distributed Beacon Scheduling”, AD HOC Networks, Volume 28, Issue1, pp. 561–575, January 2010

[1.62] B.C. Villaverde, R. Alberola, S. Rea, D. Pesch, “Experimental Evaluation of Beacon Scheduling Mechanisms for Multihop IEEE 802.15.4 Wireless Sensor Networks”, 4th Conference on Sensor Technologies and Applications (SENSORCOMM), pp. 226-231, July 2010

[2.1] Juan Lu, A. van den Bossche, E. Campo, “An Adaptive and Distributed Collision-Free MAC Protocol for Wireless Personal Area Networks”, 6th International Symposium on Intelligent Systems Techniques for Ad hoc and Wireless Sensor Networks (IST-AWSN 11), Volume 5, pp. 798-803, September 2011

[2.2] Juan Lu, A. van den Bossche, E. Campo, “Enabling Guaranteed Beacon and Data Slots in Multi-hop Mesh Sensor Networks for Home Health Monitoring”, 8th International Conference on Wireless and Mobile Communications (ICWMC 12), pp. 98-102, June 2012

[2.3] S. Mahfoudh, P. Minet, “Maximization of Energy Efficiency in Wireless Ad hoc and Sensor Networks with SERENA”, Mobile Information Systems, Advances in Wireless Networks, Volume 5 Issue 1, pp. 33-52, April 2009

[2.4] A. van den Bossche, T. Val, E. Campo, “Prototyping and performance analysis of a QoS MAC layer for industrial wireless network”, 7th International Conference on Fieldbuses and nETworks in industrial and embedded systems (IFAC 07), Volume 7 Part 1, November 2007

[2.5] P. Minet, S. Mahfoudh, “SERENA: SchEduling RoutEr Nodes Activity in wireless ad hoc and sensor networks”, 4th International Wireless Communications and Mobile Computing Conference (IWCMC 08), pp. 511-516, August 2008

[2.6] S. Mahfoudh, P. Minet, “Performance evaluation of the SERENA algorithm to SchEdule RoutEr Nodes Activity in wireless ad hoc and sensor networks”, 22nd International Conference on Advanced Information Networking and Applications (AINA 08), pp. 287-294, March 2008

[3.1] E. Egea Lopez, J. Vales Alonso, A. S. Martinez Sala, P. Pavon Marino, J. Garcia Haro, “Simulation Tools for Wireless Sensor Networks”, International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECT 05), pp. 2-9, July 2005

[3.2] Harsh Sundani, Haoyue Li, Vilay Devabhaktuni, Mansoor Alam, Prabir Bhattacharya, "Wireless Sensor Network Simulators – A Survey and Comparisons", International Journal of Computer Networks (IJCN), Volume 2 Issue 5, pp. 249-265, April 2010

[3.3] Fei Yu, Raj Jain, "A Survey of Wireless Sensor Network Simulation Tools", http://www.cse.wustl.edu/~jain/cse567-11/ftp/sensor/index.html#tossim

[3.4] NS-2, http://www.isi.edu/nsnam/ns/

[3.5] TOSSIM, http://docs.tinyos.net/index.php/TOSSIM

[3.6] OMNeT++, http://www.omnetpp.org/home/what-is-omnet

[3.7] OPNET, http://www.opnet.com/

[3.8] P. Jurcik, A. Koubaa, M. Alves, E. Tovar, Z. Hanzalek, "A Simulation Model for the IEEE 802.15.4 Protocol: Delay/Throughput Evaluation of the GTS Mechanism", 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOT 07), pp. 109-116, October 2007

[3.9] IEEE 802.15.4/ZigBee OPNET Simulation Model, http://www.open-zb.net/wpan_simulator.php

[3.10] N. Fourty, A. Vand Den Bossche, T. Val, "An Advanced Study of Energy Consumption in an IEEE 802.15.4 Based Network: Everything but the Truth on 802.15.4 Node Lifetime", Computer Communications, Special Issue: Wireless Green Communications and Networking, Volume 35 Issue 14, pp. 1759-1767, May 2012

[4.1] WiNo, http://lab.iut-blagnac.fr/dokuwiki/doku.php/projets/wsnemu/doc

[4.2] A. Van Den Bossche, T. Val, R. Dalce, "SISP: A Lightweight Synchronization Protocol for Wireless Sensor Networks", 16th conference on Emerging Technologies and Factory Automation (ETFA 11), pp. 1-4, September 2011

[4.3] Freescale, "Simple Media Access Controller Demonstration Applications", http://www.freescale.com/, September 2011

[4.4] Freescale, "Sensor Applications Reference Design", http://www.freescale.com/, June 2004

[4.5] CodeWarrior Development Tools, http://www.freescale.com/webapp/sps/site/homepage.jsp?code=CW_HOME

[4.6] 13212 Evaluation Kits, http://www.freescale.com/, June 2007

[4.7] Sensor Network Analyzer, http://www.daintree.net/sna/sna.php

[4.8] eSpeak, http://espeak.sourceforge.net/

Glossary

ADCF, *Adaptive and Distributed Collision Free*
AES, *Asynchronous Energy Saving*
AODV, *Ad-hoc On-demand Distance Vector*
BAN, *Body Area Network*
BEP, *Beacon Exchange Protocol*
BI, *Beacon Interval*
BO, *Beacon Order*
BOP, *Beacon Only Period*
BSAP, *Beacon Slot Allocation Protocol*
CAP, *Contention Access Period*
CCA, *Clear Channel Assessment*
CF, *Convergence Flag*
CFBS, *Collision Free Beacon Slot*
CFDS, *Collision Free Data Slot*
CFP, *Contention-Free Period*
CFTS, *Contention-Free Time Slot*
CSMA/CA, *Carrier Sensor Multiple Access with Collision Avoidance*
CSMA/CD, *Carrier Sensor Multiple Access with Collision Detection*
DSAP, *Data Slot Allocation Protocol*
DSSS, *Direct Sequence Spread Spectrum*
DYMO, *Dynamic MANET On-demand*
EAP, *Exclusive Access Phase*
ED, *Energy Detection*
EE, *Energy Entity*
FFD, *Full-Function Device*
FHSS, *Frequency Hopping Spread Spectrum*
FSK, *Frequency Shift Keying*
GTS, *Guaranteed Time Slot*
HBC, *Human Body Communications*
HPS, *Health Professional Stations*
IF, *Initiator Flag*
ISP, *Initiator Selection Protocol*
LLC, *Logical Link Control*
LQI, *Link Quality Indicator*
MAC, *Medium Access Control*
MCPS, *MAC Common Part Sublayer*
MFR, *MAC Footer*
MHR, *MAC Header*
MPDU, *MAC Protocol Data Unit*
MSDU, *MAC Service Data Unit*
NA, *Neighbor Address*
NB, *Narrowband*
NC, *Neighbor Count*
ND, *Neighbor Density*
NE, *Neighbor Energy*
NT, *Neighbor Table*

O-QPSK, *Offset Quadrature Phase-Shift Keying*
PAN, *Personal Area Network*
PHY, *Physical Layer*
PSDU, *PHY Service Data Unit*
PD, *PHY Data*
PLME, *Physical Layer Management Entity*
PIB, *PAN Information Base*
QoS, *Quality of Service*
RAP, *Random Access Phase*
RFD, *Reduced-Function Device*
SAP, *Service Access Point*
SD, *Superframe Duration*
SES, *Synchronous Energy Saving*
SNA, *Sensor Network Analyzer*
SO, *Superframe Order*
SPA, *Simple Priority Algorithm*
SRP, *Smart Repair Protocol*
TD, *Time Division*
TDMA, *Time Division Multiple Access*
TG, *Task Group*
THP, *Telematic Homecare Platform*
THS, *Telematic Home Stations*
TSMP, *Time Synchronized Mesh Protocol*
UWB, *Ultra-WideBand*
WPAN, *Wireless Personal Area Network*
WSN, *Wireless Sensor Network*

Table of Figures

List of Tables

List of Equations

Résumé en Français

Modélisation, simulation et implémentation d'un protocole de communication adaptatif dans un réseau de capteurs sans fil basé sur IEEE 802.15.4 et adapté à la surveillance de personnes à domicile.

Introduction

Aujourd'hui, le vieillissement de la population est en constante augmentation ainsi que le comportement de la surveillance des personnes âgées et des handicapes vivant seuls est devenue un problème majeur de mante publique dans nos sociétés modernes (T. Fent et al., 2006 ; J.R. Boulanger et C. Deroussent, 2008). Ces personnes attachent une grande importance à l'autonomie qui leur permet de vivre la plupart du temps à la maison et dans leur environnement immédiat, en leur fournissant la liberté et une meilleure qualité de vie. Masi, dans le cas d'un accident comme une chute, malaise…, que l'autonomie peut rapidement se transformer en dépendance. De fournir des solutions, certaines personnes portent des systèmes embarqués sur leur corps, tels que des capteurs physiologiques ou des capteurs d'automne (Jianchu Yao et al., 2005 ; Kwang Yong Lim et al., 2008 ; H. Mamaghanian et al., 2011). Ces dispositifs sont intrusifs et les limites deviennent apparents en raison du fait que le patient est souvent incapable d'utiliser un système d'alerte, soit parce qu'il ne porte pas son équipement ou, s'il se sent soudain malaise, est incapable d'accomplir le geste d'activation d'alerte.

La solution que nous considérons est à l'instrument de l'environnement de la personne. En effet, en surveillant les principales caractéristiques environnementales de leur espace de vie, il semble être possible d'obtenir un modèle de vie de la personne (V. Rialle et al., 2004 ; Y. Zatout et E. Campo, 2009 ; A. Anfosso et S. Rebaudo, 2011). Par exemple, mesure de la température, l'humidité, la luminosité, le bruit, la présence…, dans nombreux domaines stratégiques à la maison peuvent fournir des données utiles pour interpréter une activité physique dans l'espace et le temps. Le traitement des données permettra de déterminer les rythmes circadiens d'activité de la personne et ainsi contribuera à détecter les situations inhabituelles et les cas d'urgence. En générale, le défi est de proposer un réseau de capteurs approprié qui permet la transmission de données sans interruption dans un temps limité.

Dans ce contexte, l'objectif de ce travail est de modéliser et de mettre en place un réseau de capteurs hétérogènes complet permettant la mesure et la transmission de courte portée des données recueillies par les capteurs environnementaux. Le futur réseau sera déployé dans une maison ou même un bâtiment et de transmettre des messages d'alerte causée par un dysfonctionnement des paramètres environnementaux par un suive permanent. Ainsi, une échelle limitée, jusqu'à 50 nœuds, semble être suffisante pour cette application de surveillance à domicile. Ces nœuds échangent de données entre eux selon un protocole de communication qui permet d'optimiser la consommation d'énergie, le délai de transmission et de perte d'information. Un autre principe à considérer est que si un nœud tombe en panne, le réseau devrait réparer automatiquement et doit fonctionner normalement avec une perte minimale d'information.

Transmission considérée va utiliser la technologie sans fil à faible puissance combinée le cas échéant avec une communication filaire. Nous commençons donc ce travail à partir de l'état de l'art des technologies filaires et sans fil utilisés dans le domaine de la surveillance à domicile. Nous constations que certaines technologies filaires soutenir à la fois le faible taux et des communications à haut débit. Cependant, notre travail se concentre sur les réseaux à faible débit qui ciblent principalement à la transmission des données du capteur, même dans des cas exceptionnels ou d'urgence. D'autre part, un réseau de capteurs sans fil permettant un suivi efficace pendant quelques semaines ou quelques mois pourrait constituer un scenario très intéressant, au lieu de pénétrer les murs pour installer un réseau câblé. Par conséquent, nous profitons des technologies sans fil en termes d'installation facile, un déploiement flexible et environnement confortable pour les personnes suivies. L'un des défis dans ce cas est les capteurs d'énergies limitées.

En fait, du point de vue du réseau, l'essentiel de ce travail est le réseau personnel sans fil, WPAN, qui tente de fournir des solutions à faible puissance, à faible coût et à courte portée. Parmi eux, IEEE 802.15.4 est considéré comme une voie prometteuse en termes d'économies d'énergie et d'accès au médium avec garantie. Beaucoup d'autres technologies principales telles que ZigBee, IEEE 802.15.5 et 6LoWPAN sont basés sur IEEE 802.15.4 MAC ou rétro-compatible avec cette norme. Par conséquent, nous considérons IEEE 802.15.4 comme point de départ pour notre travail. En fait, nous utilisons IEEE 802.15.4 couche physique telle qu'elle est, sans aucun changement. De l'autre côté, nous avons optimisé la norme IEEE 802.15.4 couche MAC afin de mieux adapter nos contraintes spécifiques. La couche MAC a un impact fondamental et important dans une pile de protocoles. Les couches supérieures, y

compris la couche réseau, la couche transport, la couche application, etc. ne sera considérée âpres une couche MAC robuste.

C'est pourquoi nos travaux sont axés sur la couche MAC du modèle OSI. De cette façon, nous améliorons norme IEEE 802.15.4 pour satisfaire notre demande particulière. Le nouveau protocole de communication devrait avoir la capacité de donner des priorités différentes à diverses données en fonction de leurs exigences en contrôlant le partage du médium. Il signifie en fait la nécessité de différentes méthodes d'accès au médium. Comme on le sait, CSMA/CA est une méthode d'accès basé sur la contention qui fournit un service best-effort. Cependant, notre application exige que les communications avec une faible latence et sans perte de paquets, en particulier pour les messages d'alerte qui peuvent directement affecter la sécurité de la sante des personnes surveillées. La méthode d'accès garantie est donc attendue instamment.

Pendant ce temps, nous décidons de profiter de l'architecture maillée pour construire et entretenir le réseau sans fil. L'architecture maillée permet l'organisation automatique, pas de gestion centralisée à l'aide d'un nœud super et la récupération voie rapide car les communications sont possibles avec tous les nœuds voisins. Contrairement ZigBee, nous désirons que tous les nœuds, y compris les routeurs, puissent dormir dans le réseau maillé pour économiser l'énergie. Le mécanisme intelligent pour calendrier est également prévu d'étendre la durée de vie du réseau autant que possible. Un autre avantage de l'architecture maillée est sa robustesse. Tous les nœuds, y compris les routeurs, peuvent échouer, mais le reste du réseau devrait fonctionner correctement à l'aide de la redondance des liaisons de l'architecture maillée. Enfin, contrairement à la topologie en étoile ou en arbre dans lequel est typiquement un nœud super préalablement fixé et planifier les ressources partagées, une topologie maillée permet de mieux s'adapter aux changements topologiques et renforcer la flexibilité et la sécurité de la surveillance.

En général, nous travaillons à la couche MAC du réseau de capteurs sans fil maillé, ce protocole MAC permet différents niveaux de QoS avec une consommation rationnelle de l'énergie. Modélisation et simulation sont importants méthodes de travail aident à vérifier notre protocole de communication, d'évaluer ses performances et à améliorer les propositions. Mise en œuvre prototype est également atteinte avec les cartes d'application disponibles dans des situations réelles. Ce travail peut vérifier la faisabilité et la précision du modèle de simulation et permet d'optimiser le modèle de protocole en retour.

D'où le manuscrit est structure de la manière suivante: tout d'abord, notre contexte d'application et les défis sont détaillées dans le chapitre 1 qu'il explique la motivation et les objectifs de ce travail. Les principales technologies filaires et sans fil sur la surveillance des habitats, y compris IEEE 802.15.4, sont étudies. Leurs limites pour nos besoins d'application conduisent à la nécessite d'un nouveau protocole d'adaptation, ce qui permet d'accès au médium déterministe et l'économie d'énergie pour tous les nœuds.

Dans le chapitre 2, un nouveau protocole MAC est proposé afin d'améliorer la robustesse et la flexibilité du réseau de capteurs multi sauts. Ce chapitre contient la description de la formation de réseaux, architecture de nœud, la fonction du protocole et de ses détails de fonctionnement.

Dans le chapitre 3, nous simulons le protocole proposé avec OPNET réseau simulateur pour évaluer la portée de notre contribution. Les résultats des simulations montrent des performances dans les aspects de coût protocole, la capacité de qualité de service et la consommation d'énergie, etc.

Enfin, le chapitre 4 présente le prototype afin de vérifier notre proposition et d'améliorer le protocole en résolvant les problèmes non pris en compte dans la simulation. Nous mettons en œuvre le projet de protocole sur les cartes de capteurs intègres et déployer le réseau constitué de plusieurs nœuds dans un environnement réel dans une maison intelligente.

En guise de conclusion et quelques perspectives sont données dans la dernière partie de ce manuscrit.

Chapitre 1

Dans le chapitre 1, la motivation de cette thèse est adressée. Plusieurs projets sur la surveillance des habitats en utilisant un réseau de capteurs sans fil sont rapidement présentés et leurs caractéristiques communes sont identifiées et discutées. Cette thèse se concentre sur WSN pour la mesure à distance des paramètres environnementaux et sanitaires.

Dans notre application, le WSN présente les caractéristiques suivantes :

- L'environnement intérieur : les nœuds sont mis sur le plafond, sur ou des meubles à la maison. Par conséquent, les émetteurs et récepteurs sans fil sont limités à la communication de courte portée avec une puissance faible.

- Bande passante basse : il y a généralement deux types des données dans notre application. Les paramètres environnementaux tels que la température sont communiqués périodiquement. D'autre part, les données d'éclatement comme alarme de température en incendie ou de la chute de malaise devraient être livrés avec garantie. Le débit est généralement faible.

- Nombre variable de multi-capteurs : le WSN inclut des capteurs environnementaux tels que les capteurs de température, humidité, luminosité, etc. et de la santé tel que des accéléromètres ou des capteurs physiologiques. Notre WSN devrait soutenir environ 50 nœuds.

- Topologie maillée: notre application nécessite que certains messages essentiels tels qu'alarme de chute doivent être transmis en temps sans perte. Quand un nœud tombe en panne, reste du réseau doit trouver une nouvelle façon d'envoyer des messages vitaux. Donc nous avons besoin d'une topologie maillée qui permet de mieux s'adapter aux changements topologiques et renforcer la sécurité et la robustesse de la surveillance.

- Les économies d'énergie : nous nous attendons non seulement pour maximiser la durée de vie de la batterie de capteur, mais aussi de prolonger la durée de vie de l'ensemble du réseau. Tous les nœuds, y compris les routeurs, pouvaient dormir pour économiser l'énergie.

Deuxièmement, nous présentons un résumé des normes filaire et sans fil de réseau domestique. Le point important est mis sur la comparaison des avantages et inconvénients de ces technologies pour notre application. De manière générale, les appareils filaires sont plus complexe parce qu'il n'est pas toujours possible d'installer des câbles dans l'habitat ou les personnes âgées ne peut être libre avec les câbles. Donc notre travail se concentre sur le réseau sans fil et compare les technologies comme indiquées dans le tableau 1.

Enfin, nous avons choisi IEEE 802.15.4 parce que ses caractéristiques de QoS, des économies d'énergie et des facteurs pratiques tels que les nombreux produits disponibles. Ensuite, cette norme est entièrement étudie et les problématiques comme les collisions de balise, l'allocation des slots dynamique, économie d'énergie sur les routeurs etc. sont présentés. Nous essayons d'adapter IEEE 802.15.4 pour le réseau maillé.

Tableau 2 Comparaison des technologies

Technologies	QoS garantie	Consommation d'énergie	Topologie maillée	Produits disponibles
IEEE 802.15.4	Oui	Faible	Non	Oui
IEEE 802.15.6	Oui	Faible	Non	Non
ZigBee	Non	moyenne	Non	Oui
IEEE 802.15.5	Oui	Faible	Oui	Non
6LoWPAN	Non	Haut	Oui	Oui
Z-Wave	Non	Faible	Oui	Oui
WirelessHART	Oui	Moyenne	Oui	Oui

En conclusion, nous travaillons à la couche MAC pour construire un réseau qui devrait être l'économie d'énergie, flexible, robuste et avoir la capacité de QoS. Ils sont autant de facteurs importants qui leur seront soigneusement pris en compte dans la conception de protocole et progressivement testé en simulation et prototype.

Chapitre 2

Comme illustré dans le chapitre 1, notre application nécessite un protocole de communication adaptatif fournissant un service de QoS garantie par une consommation d'énergie raisonnable dans un réseau maillé. Les technologies actuelles et les protocoles associés ne peuvent pas résoudre tous les problèmes ensemble. Par conséquent, nous avons proposé un protocole original ADCF et l'a présenté dans ce chapitre.

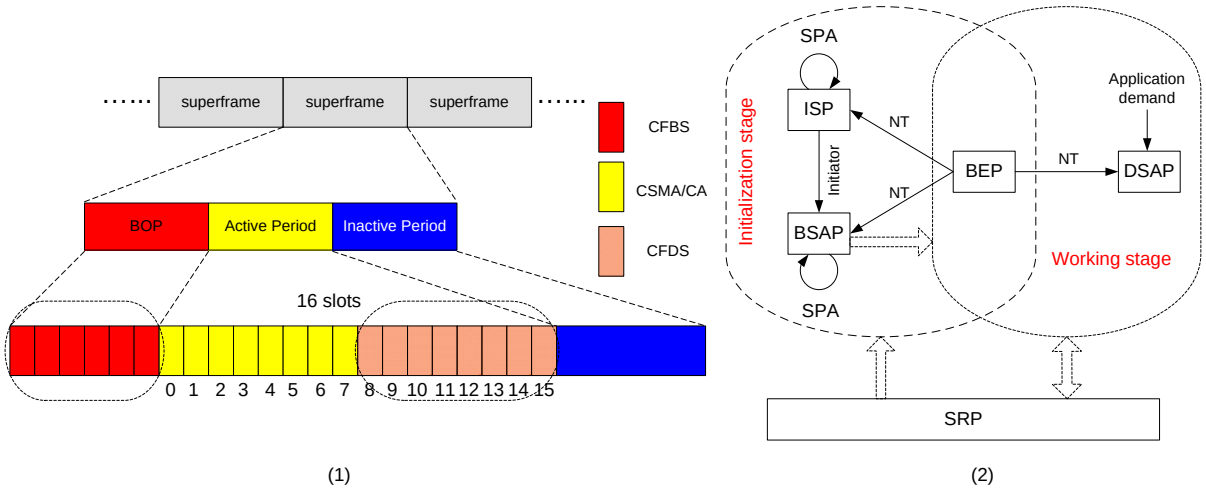

Figure 1 Le structure de super trame et le schéma de fonctionnement d'ADCF

ADCF inclut 2 étapes, l'étape d'initialisation et l'étape de travail. Il n'y a pas de super trame dans l'étape d'initialisation, les nœuds envoient des données par unslotted CSMA/CA pour construire un réseau maillé. Dans l'étape de travail, il y a la super trame basée sur la structure de super trame classique de la norme IEEE 802.15.4.

Deux mécanismes, CFBS et CFDS ont été proposées, comme indiqués dans la figure 1(1). CFBS permet aux nœuds loin que 2-saut de réutilise les slots de sorte que les collisions de balises pourraient être évités. Les nœuds peuvent rejoindre ou quitter le réseau comme ils veulent, car BOP change dynamiquement en fonction des changements topologiques. Par CFBS, CFDS permet aux nœuds de négocier les slots sans collisions dans la topologie maillée. Le médium sans fil est donc dédié aux nœuds qui utilisent CFDS pour transmettre les données dans un délai limité. En outre, ADCF permet à tous les nœuds, y compris les routeurs, de dormir pour économiser l'énergie.

Afin de simplifier la compréhension de l'ensemble du processus, ADCF est divisé en plusieurs protocoles et algorithmes associés, comme montré sur figure 1(2).

- BEP : le début et la base d'ADCF sont BEP qui met en place et mises à jour le tableau de voisinage 2-saut en deux étapes. Chaque nouveau nœud va tout d'abord écouter le canal dans une période déterminée. Selon les balises reçues, le nouveau nœud envoie sa propre balise par des mécanismes différents. Chaque nœud diffuse sa balise dans 1-saut et note les informations intéressant comme l'adresse, l'énergie et la densité dans le tableau de voisinage. Par conséquent, toutes les informations des voisins 2-saut sont obtenues par ce nouveau nœud.

- SPA : il est mis en œuvre en comparant 3 paramètres des nœuds. L'ordre de comparaison est la densité, l'énergie et l'adresse. Dans un premier temps, le nœud avec un maximum densité est sélectionné. Si les nœuds ont la même densité, SPA choisit celui avec un maximum énergie. Enfin, le nœud avec l'adresse minimale a la plus haute priorité si deux autres paramètres sont identiques.

- ISP : avec les informations contenues dans le tableau de voisinage, ISP est exécutée. L'objectif de ce protocole est de choisir un initiateur qui a deux fonctions. Il spécifie le début de BOP et mesure la longueur de BOP. Chaque nœud sélectionne un candidat initiateur par SPA. Si un candidat initiateur est différent de les voisons, SPA est utilisé à plusieurs reprises pour décider d'un initiateur unique.

- BSAP : il est déclenché lorsque l'initiateur est décidé. Ce protocole permet à chaque nœud de choisir un CFBS dans BOP. Les nœuds exécutent SPA localement et un nœud avec une priorité plus haute choisie CFBS d'abord. Il prend la première slot

disponible qui n'est pas utilisé par ses voisins 2-saut et stocke le numéro de slot dans son tableau de voisinage. A la Fin de BSAP, le nœud entre dans l'étape de travail.

- DSAP : il est déclenché par une demande d'un niveau supérieur. Chaque nœud peut demander CFDS par balise à tous ses voisins. Quand un nœud reçoit la balise du voisin et trouve son adresse comme destination, il va vérifier son tableau de voisinage, allouer le première slot disponible au nœud demandeur et annoncer cette allocation dans sa balise suivante. Lorsque le nœud demandeur reçoit la balise avec le numéro de slot décidé, les deux nœuds peuvent communiquer dans ce slot CFDS.

- SRP : il permet de passer des nœuds entre deux étapes. Ce protocole tente de réduire l'impact d'un changement topologique autant que possible. En générale, les changements topologiques sont classés comme 4 types, l'augmentation du BOP, la réduction du BOP, la séparation du réseau et l'intégration du réseau. Nous discutons les 4 types et présentons les mécanismes correspondants. Ce protocole est vital dans ADCF car il améliore la souplesse et la robustesse du réseau.

Enfin, les primitives de service et les paramètres utilisés dans le nœud ADCF ont été expliqués. Dans les deux chapitres suivants, les implémentations de simulation et de prototype seront présentées pour évaluer les contributions d'ADCF.

Chapitre 3

Nous avons commencé ce chapitre d'une enquête d'outils de simulation pour WSN. Enfin OPENT est choisie en raison de la qualité de sa programmation, GUI conviviale et des capacités de traitement des données. Plus important encore, OPNET contient une complète mis en œuvre IEEE 802.15.4.

Ensuite, nous avons présenté notre modèle de simulation qui met en œuvre toutes les propositions du chapitre 2. La modélisation du réseau a été organisée avec des modules de nœud ADCF. Chaque couche du nœud a été illustrée dans le domaine de processus. Pendent ce temps, les configurations et les paramètres de simulation ont été donnés.

Enfin, de nombreux scénarios expérimentaux ont été simulés et les résultats intéressants ont été présentés. Dans les problématiques d'application du chapitre 1, nous avons classé les indicateurs des performances de la couche MAC : la capacité de QoS, la flexibilité, la

robustesse et l'économie d'énergie. Donc, nous avons discuté des résultats de la simulation de ces 3 parties.

Figure 2(1) montre la comparaison du délai de bout en bout pour des données QoS. Il y a une petite différence pour les données 1-saut, en moyenne, ADCF a 60 ms avantage quand les mémoires sont disponibles. Pour les données multi sauts, ADCF est mieux que IEEE 802.15.4 grâce à la disponibilité de la topologie maillée. D'autre part, tous les taux de réussite de paquets toujours gardent 100% lorsque les mémoires sont disponibles. Donc, ADCF satisfait notre demande d'application de transmettre des données d'urgence dans un temps limité et sans perte de paquet.

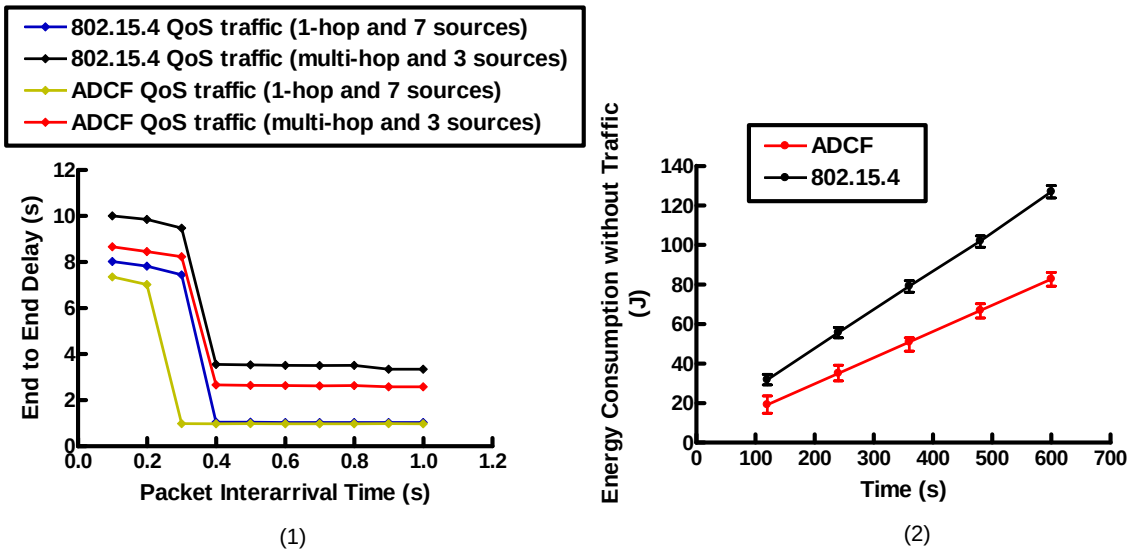

Figure 2 Comparaison d'ADCF avec IEEE 802.15.4: délai et énergie

Figure 2(2) montre qu'ADCF consomme moins d'énergie qu'IEEE 802.15.4. C'est parce qu'il y a 2 périodes actives dans le réseau en arbre de la norme. Donc plus de temps sont consacré à écouter le médium. En générale, environ 37.5% de l'énergie peut être sauvé par ADCF. D'autre part, nous étudions le coût de construction d'un réseau maillé. Le pire des cas dans l'étape de travail est de reconstruire le réseau de sorte que le coût d'entretien pourraient être considérées par cette étude aussi. Les résultats montrent que le coût est rationnel, par exemple moins de 30 s pour un réseau de 50 nœuds. Dans l'application de la surveillance à domicile, une étape de travail dure habituellement plusieurs mois. Par rapport à ce réseau à long terme organisé, le coût du temps de convergence est intéressant et acceptable. En outre, le temps de convergence peut être encore réduit au prix d'un plus grand frais généraux de messages.

En fait, la relation entre les performances du réseau et des changements topologiques est difficile à étudier en raison du nombre énorme de cas complexes et spécifiques. Nous simulons les cas les plus courants comme des exemples et un seul changement de topologie est généré à chaque fois. Les conclusions suivantes peuvent être obtenus auprès de la simulation. Tout d'abord, ADCF permet au réseau d'être bien fait sans point de défaillance unique. Dans le même temps, le réseau peut fonctionner correctement avec certains nœuds d'assemblage. Enfin, dans le pire des cas, un processus de reconstruction du réseau est nécessaire et le temps de la reconstruction est acceptable.

Ainsi, le travail actuel et les résultats obtenus de simulation vérifient nos propositions et montrent les avantages d'ADCF. Certaines limites de simulation seront discutées dans la conclusion générale.

Chapitre 4

Mise en œuvre du prototype est une approche fondamentale de vérifier un protocole et ses performances. Nous commençons ce chapitre par l'introduction d'une plateforme de prototypage rapide nommé WiNo. Un développeur avec WiNo peut maîtriser non seulement le temps d'accès au médium et le cycle veille-sommeil, mais aussi le temps CPU et mémoire généralement limité par le matériel dans un WSN.

Les cartes prises en charge, 13192-SARD et 1321x- SRB, sont présentées. En fait, deux outils comme la console du nœud et l'analyseur SNA sont principalement utilisés dans notre prototype pour évaluer les performances du protocole dans des conditions réelles.

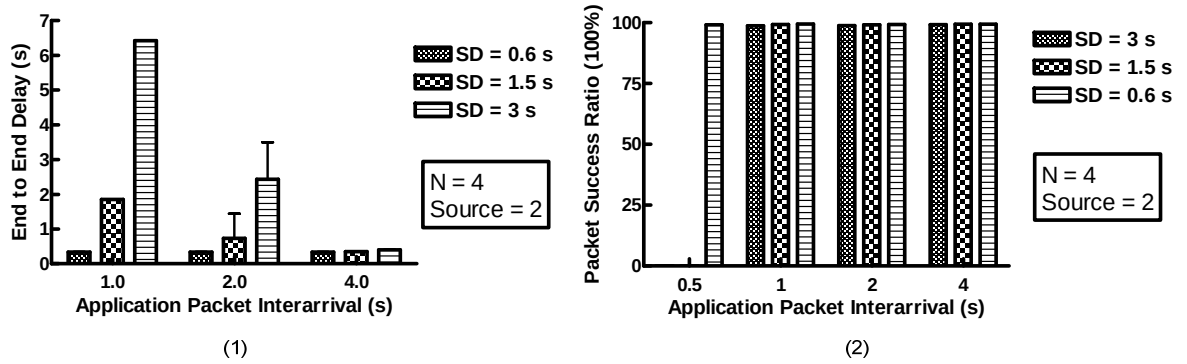

Figure 3 Les résultats du prototype: délai et taux de réussite de paquets

Ensuite, nous expliquons la mise en œuvre d'ADCF, y compris des améliorations telles que le mécanisme de synchronisation et le mécanisme de confirmation des liens. La partie

suivante présente les résultats représentatifs obtenus grâce à des mesures concrètes dans un environnement réel et compare ces résultats avec la simulation.

Tout d'abord, le coût protocolaire est acceptable. Par exemple, le temps de convergence est 82.54 s en moyenne et 489 balises sont envoyées pendant l'étape d'initialisation pour construire un réseau maillé de 8 nœuds sans collisions. Les nœuds peuvent librement rejoindre ou quitter le réseau sans perturber le fonctionnement du reste du réseau. Le point plus intéressant est la négociation des slots des données avec succès via les balises. Comme montré sur figure 3, CFDS peut transmettre des données en 0.37 s en moyenne et sans perte de paquets quand les mémoires sont disponibles. C'est encore mieux que la simulation à cause des modèles de couche d'application différents. En conclusion, les résultats du prototype montrent qu'ADCF répond à nos exigences de l'application.

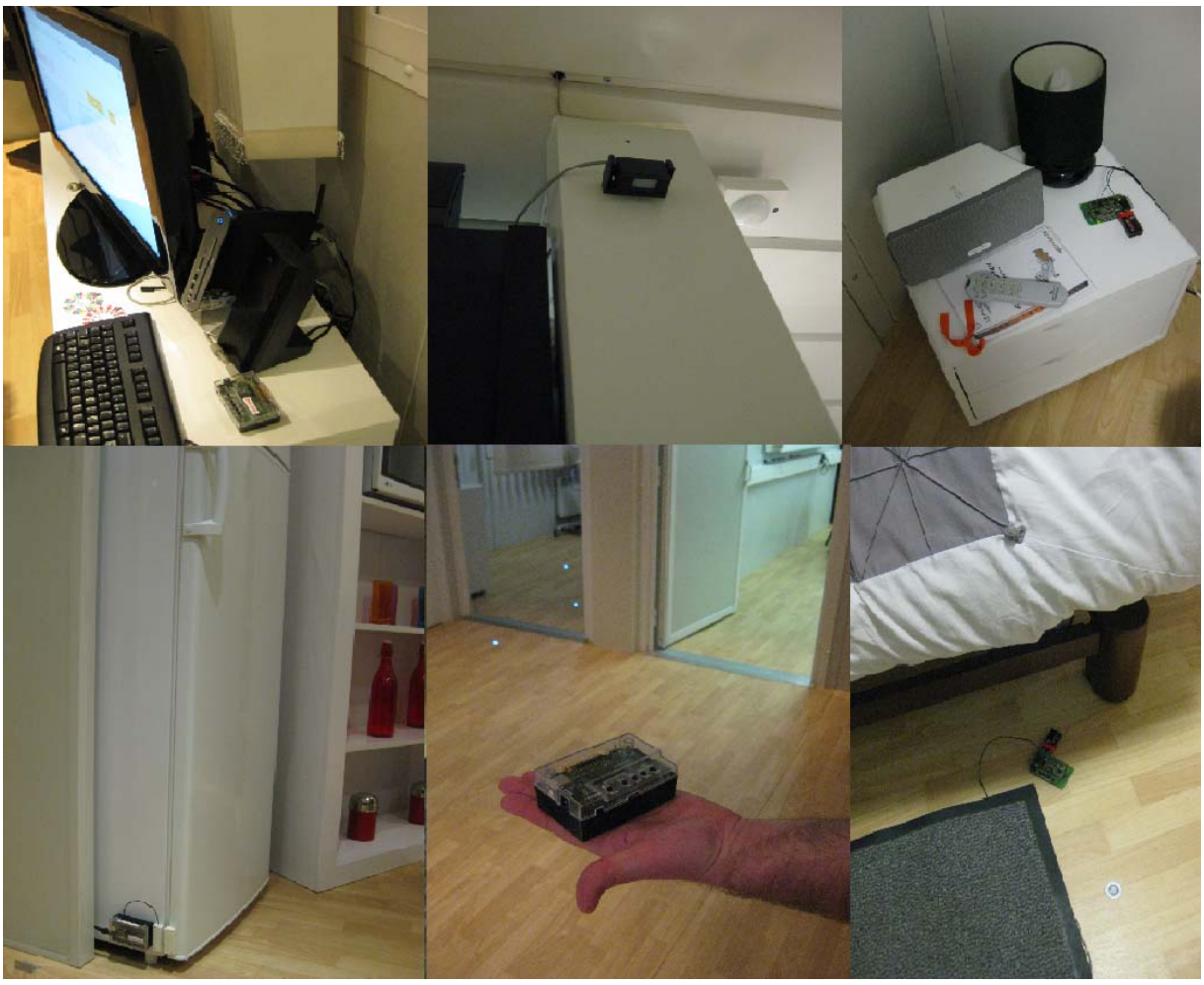

Figure 4 Les nœuds ADCF dans la maison intelligente de Blagnac

Finalement, 6 nœuds ADCF ont été déployés dans la maison intelligente comme montré sur figure 4. Du point de vue MAC, ces 6 nœuds forment un réseau maillé complet. Les

nœuds sont reliés par différents capteurs correspondant à la cible de notre application : le sink, le capteur magnétique, le capteur infrarouge, bouton d'urgence, le capteur lumière et le capteur moquette. Du point de vue applicatif, les communications entre les nœuds ADCF sont réalisées par CFDS. Merci à le buzzer disponible sur les nœuds qui est activé à chaque réception des données, nous pouvons vérifier que le temps borné est vérifié comme prévu. Donc ADCF remplit certaines lacunes d'application et fournit une solution sans fil alternative dans la maison intelligente.

Conclusion générale

Ce manuscrit a présenté une «Adaptive and Distributed Collision Free » MAC protocole basé sur la norme IEEE 802.15.4. Ce protocole a été conçu pour construire et maintenir un réseau capteur maillé sans fil fournissant une solution avec QoS garantie et d'économie d'énergie pour les applications de surveillance à domicile. Le travail a été organisé en trois phases : présentation du protocole, la mise en œuvre et les résultats de la simulation et le prototype.

Dans le chapitre 1, nous avons résumé les architectures communes pour les réseaux de surveillance de l'habitat et mis l'accent sur WSN. L'objectif était de fournir une solution alternative sans fil avec l'avantage d'une installation simple et un déploiement flexible pour plusieurs semaines ou mois, par rapport aux technologies filaires comme KNX, HART, etc. qui imposent un déploiement de réseau coûteux. Considérant bus câbles comme KNX, le médium commun permet des communications de bout en bout entre les nœuds. Cependant, les technologies sans fil actuelles telles que ZigBee parfois désactivent les communications directes en raison de la restriction de la topologie en arbre. Dans ce cas, les données doivent être acheminées à partir du nœud de source à un nœud super puis le nœud super envoie des données au nœud de destination, même si les deux nœuds se trouvent dans la portée de transmission de l'autre. En outre, alors que IEEE 802.15.4 topologie en arbre permet d'économie d'énergie sur les routeurs, ZigBee n'utilise pas ce mode, donc les routeurs sont toujours actives. D'autres technologies sans fil telles que 6LoWPAN et Z-Wave ne considèrent pas la qualité de service garantie. Par conséquent, un nouveau protocole de communication a été prévu pour être économie d'énergie, flexible, robuste, et à avoir des capacités de QoS en même temps. Nous avons seulement travaillé à la couche MAC pour gérer ces défis et résoudre les problèmes tels que les collisions de balises, les changements

d'états de liens et la synchronisation sur des chemins multi sauts dans un réseau maillé. Des couches supérieures tells que la couche de routage sera considéré en perspective.

Dans le chapitre 2, notre proposition appelée ADCF avait été entièrement évoquée. Ce MAC protocole original est basé sur la norme IEEE 802.15.4 2.4 GHz DSSS couche physique et la structure de super trame classique. En effet, le standard prend en charge les mécanismes très intéressants pour la QoS et l'économie d'énergie et propose de nombreux produits commerciaux disponibles. Le point important était mis sur l'adaptation IEEE 802.15.4 pour le réseau maillé dans lequel tous les nœuds pouvaient dormir pour économiser l'énergie et peut tomber en panne sans perturber le reste du réseau. En général, ADCF comprend 2 étapes : dans l'étape d'initialisation, les nœuds envoient des balises par unslotted CSMA/CA pour construire un réseau maillé. Les coûts de construction du réseau, les tempes de convergence et les frais généraux de messages, sont liés aux paramètres tels que N, D_{max}, H_{max} qui ont été entièrement étudiées dans les chapitres 2 et 3. Dans l'étape de travail, basé sur la norme IEEE 802.15.4 structure de super trame, ADCF divise le temps en trois parties: dans BOP qu'il est organisé par CFBS et change dynamiquement en fonction des changements topologiques des liens sans fil, les nœuds loin que 2 sauts peuvent réutiliser les mêmes slots de sorte que les collisions de balises pourraient être évites. Dans le période active, CFDS a été proposé pour permettre aux nœuds de négocier les slots de données dédiées à la topologie maillée. Merci à CFDS, les données d'application peuvent être transmises en un temps limité et sans perte de paquet. Enfin, les nœuds d'aller dormir dans le période inactive. Donc, notre contribution a été les mécanismes CFBS et CFDS. Afin de réaliser ces fonctions protocolaires, ADCF a été divise en un ensemble de protocoles/algorithmes : BEP, SPA, ISP, BSAP, DSAP et SRP. Chacun d'entre eux a été entièrement détaillé et une étude théorique a été donnée pour évaluer le coût de protocole dans le pire des cas. A la fin de ce chapitre, les primitives de service et les paramètres utilisés dans le nœud ADCF ont été expliqués. La description de ces primitives permettra la mise en œuvre d'ADCF, par exemple sur un logiciel de simulation ou un nœud réel. Un réseau maillé efficace et multi sauts pourrait être construits et entretenus avec ces nœuds ADCF.

Dans le chapitre 3, nous avons présenté la simulation d'ADCF. OPNET simulateur a été choisie comme la qualité de sa programmation, interface utilisateur conviviale et des capacités de traitement des données. Plus important encore, OPNET contient une complète mise en œuvre IEEE 802.15.4 qui rend la comparaison des ADCF et IEEE 802.15.4 possible. Après la présentation du modèle de simulation et les paramètres, de nombreux scénarios

expérimentaux ont été simulés et les résultats intéressants ont été présentés. Nous avons discuté des résultats de la simulation à partir de 3 parties : QoS, économie d'énergie, la flexibilité et la robustesse, tel que requis par notre application. Les conclusions de simulation sont les suivants:

- ADCF satisfait notre demande d'application de transmettre des données avec QoS différents. Quand les mémoires sont disponibles, CFDS permet la livraison des données sans perte de paquets. Le délai de bout en bout dépend de la structure de super trame et nos résultats de simulation confirment qu'ADCF n'est jamais pire que 802.15.4 pour la livraison des données QoS. Dans certains cas, tels que le réseau multi sauts, ADCF peut être encore mieux que 802.15.4 grâce à la disponibilité des trajets plus courts au sein de la topologie maillée.

- Les coûts d'ADCF sont acceptables. Nous pouvons construire un réseau maillé de 30 nœuds dans 25 s et avec peu de surcharge. Evidemment, le coût d'ADCF comprend également sa consommation d'énergie. Résultat de la simulation montre qu'ADCF consomme moins d'énergie, environ 37%, par rapport à 802.15.4. Nous pouvons encore améliorer les performances telles que le temps de convergence et le délai de bout en bout au prix de la consommation d'énergie. Ainsi, le compromis doit être fait selon les environnements applicatifs spécifiques.

- Pour plus de flexibilité et de robustesse, de nombreux cas ont été examinés et nous avons donné le pire des cas, la reconstruction du réseau, comme des exemples représentatifs. Par rapport à la topologie en étoile ou en arbre, grâce à ADCF, le réseau fonctionne correctement même si il y a des nœuds d'échec. En outre, les nouveaux nœuds pourraient rejoindre le réseau librement, ce qui augmente la flexibilité du réseau. Dans certains cas, comme un réseau multi sauts avec CFBS disponibles, de nouveaux nœuds peuvent parfaitement insérer la super trame et envoyer les balises sans collisions.

En outre, les résultats des simulations montrent que l'échelle du réseau et la densité voisine n'ont aucune influence sur les données QoS qui est envoyé par le mécanisme CFDS. Les données QoS peut être envoyé sans perte de paquets, ce qui démontre la stabilité des performances d'ADCF, si les mémoires sont disponibles. Par conséquent, le travail de simulation et les résultats vérifient les avantages de l'ADCF.

Dans le chapitre 4, nous avons présenté la mise en œuvre du prototype ADCF. En utilisant la plateforme WiNo et les cartes d'application comme 13192-SARD et 1321x-SRB, nous avons obtenu la mise en œuvre de notre proposition et résolu de nombreuses difficultés dans l'environnement réel tels que le déploiement de réseau multi sauts. Cependant, certains détails n'ont pas été pleinement pris en compte dans la simulation et certains problèmes ont été identifiés dans un environnement réel. Ainsi, les améliorations telles que le mécanisme de synchronisation et le mécanisme de confirmation des liens doit être ajouté sous forme de prototype. Nous avons discuté des similitudes et des différences entre le prototype et la simulation et analysé les raisons de ces différences. En utilisant des outils tels que la console du nœud et l'analyseur SNA, nous avons obtenu les résultats suivants par des mesures pratiques: un réseau maillé de 8 nœuds pourrait être construit avec succès. La super trame est organisée où chaque nœud peut choisir un slot sans collisions. Les coûts protocolaires sont également acceptables. Pour un réseau de 8 nœuds, le temps de convergence est 82.54 s en moyenne et 489 balises sont envoyées pendant l'étape d'initialisation du réseau. Les nœuds peuvent librement adhérer ou quitter le réseau sans perturber le fonctionnement global du système. Via CFBS, les nœuds peuvent négocier les slots des données avec succès. Une fois les nœuds sont en l'étape de travail, CFDS peut fournir la livraison des données en 0.37 s en moyenne et sans perte de paquets, ce qui est encore mieux que la simulation. Donc les résultats de prototype sont très encourageants. Enfin, 6 nœuds ADCF ont été déployés dans la maison intelligente de Blagnac. Les nœuds sont connectés avec différents capteurs tels que le capteur magnétique, le capteur infrarouge, bouton d'urgence et le capteur moquette. Plusieurs scénarios intéressants en utilisant les informations du capteur ont été mis en place pour surveiller les activités de l'utilisateur. Dans la situation étudiée, le réseau peut toujours converger dans environ 10 s avec les paramètres appropriés. Grâce aux buzzers, nous pouvons entendre la réception d'une donnée d'application sur le sink dans 2 s au plus, qui confirme qu'ADCF est vérifiée dans un environnement réel comme prévu.

L'étude ci-dessus prouve qu'ADCF présente de bonnes performances et répond à nos besoins qui sont au fond pour remplacer un bus câblé en application de surveillance à domicile. Certainement, ADCF n'est pas parfaitement adapté à toutes les situations dans le cadre du domaine de surveillance. Par exemple, certaines applications avec haut débit et lourd données n'ont pas pu être transmis par un réseau ADCF avec des performances correctes. En suit, généralement l'utilisateur marche à la maison avec nœud ADCF ne pas provoquer de nombreux changements topologiques à cause de l'espace domicile limités et il est donc

parfaitement acceptable. Cependant, les nœuds avec haute vitesse qui conduisent à fréquenter les changements topologiques et les reconstructions du réseau doivent utiliser une autre technologie. Donc, un autre manque d'ADCF est la mobilité des routeurs. Enfin, le réseau à grand échelle, des centaines de milliers de nœuds, doivent utiliser architecture multi-tiers, au lieu de l'architecture maillée. En bref, l'application de surveillance à domicile nécessite un système complexe intégrant de nombreuses technologies différentes. ADCF fonctionne uniquement à la couche MAC et propose une solution alternative sans fil entre eux. De nombreuses questions sont encore ouvertes pour l'avenir.

Perspectives

Tout d'abord, l'algorithme de contrôle de topologie a une influence importante sur le réseau maillé. Un algorithme de contrôle de topologie consiste à optimiser la topologie en modifiant la puissance de transmission ou même la position des nœuds lorsque le réseau est déployé ou dans l'étape de travail. Un algorithme de contrôle de topologie favorable peut réduire la consommation d'énergie et améliorer la capacité du réseau, tout en maintenant la connectivité du réseau. En fait, nous avons défini les paramètres du réseau tels que N, D_{max}, H_{max}. Les résultats de la simulation et du prototype montrent leurs impacts sur les performances du protocole. Cependant, l'étude théorique et un algorithme de contrôle de topologie efficace est prévu afin de mieux profiter des avantages de maillé comme la redondance des liens, optimiser la gestion de la configuration du réseau et finalement nous aider à déployer le réseau approprie pour chaque scénario d'application spécifique.

Deuxièmement, la mise en œuvre du prototype ADCF n'est pas exactement la même que celui de la simulation. A l'heure actuelle, les cartes disponibles ciblées ne peuvent pas dormir et estimer la consommation d'énergie à cause de les limites de conception du matériel. Quelques améliorations simples d'ADCF comme repos dans CFDS non utilisés ou non possédés n'ont pas été testés en raison de l'absence de possibilité du matériel. Ainsi, certains autres cartes devraient être envisagées pour réaliser la mise en œuvre ensemble d'ADCF et pour évaluer la performance de la consommation d'énergie. Da la même manière, l'analyse comparative d'ADCF sur des plateformes tells que SensLab doit être envisagée.

Comme expliqué dans le chapitre 4, le prototype manque un protocole de synchronisation. Une des hypothèses est que les nœuds ADCF ont été considérés comme synchronisés. Dans la simulation, les nœuds sont synchronisés par le simulateur OPNET, ce qui était enjeu fort dans

la mise en œuvre du prototype : environ toutes les 4 heures, les nœuds perdent leur synchronisation d'horloge sous forme de prototype en raison d'un dépassement du compteur d'horloge 32-bits. Ainsi, un protocole de synchronisation de l'horloge comme SISP est nécessaire. Une perspective intéressante est l'intégration des SISP en ADCF.

Un protocole de routage simple et efficace est à l'étude. ADCF se concentre sur la couche MAC, mais offre une base lucrative pour les couches supérieures. Par exemple, le nœud ADCF a déjà entretenu un tableau de voisinage 2-saut qui peut être une information très utile à la couche réseau. Puis ADCF met en œuvre la négociation de CFDS entre les voisins 1-saut. Le protocole de routage est prévu de mettre en œuvre la négociation de CFDS multi sauts. Si les CFDS alloués sont reliés séquentiellement dans la super trame suivant la route à partir du nœud de source vers le nœud de destination, nous pouvons voir que le délai de bout en bout seront réduits au minimum. Un point de vue très intéressant est l'optimisation de l'ordonnancement de CFDS suivant la route, soit pour réduire le délai de bout en bout, ou économiser d'énergie regroupant les CFDS libre ensemble. Par conséquent, une conception « cross-layer » en profitant des avantages d'ADCF sera mis en œuvre pour réserver CFDS multi sauts et router des paquets dès que possible.

En outre, le réseau d'ADCF est utilisé dans une maison intelligente dans laquelle de nombreuses solutions, y compris autres technologies filaire et sans fil. Nous devons faire ADCF compatible avec ces technologies, par exemple, un réseau de capteur physiologique du corps ou à forte mobilité et des réseaux domotique utilisant des technologies telles que Bluetooth, WiFi, RF, etc. Parfois, le réseau d'ADCF pourrait aussi avoir besoin d'interagir avec d'autres réseaux tels que le réseau mobile lorsque l'utilisateur est à l'extérieur, le réseau de l'hôpital ou Internet, afin de jouer un rôle plus important dans le système de surveillance à domicile.

Modeling, simulation and implementation of an 802.15.4 based adaptive communication protocol in wireless sensor network: application to monitoring the elderly at home

Juan LU

Abstract: Monitoring behavior of the elderly and the disabled living alone has become a major public health problem in our modern societies. Among the various scientific aspects involved in the home monitoring field, we are interested in the study and the proposal of a solution allowing distributed sensor nodes to communicate with each other in an optimal way adapted to the specific application constraints. More precisely, we want to build a wireless network which consists of several short range sensor nodes exchanging data between them according to a communication protocol at MAC (medium access control) level that optimizes energy consumption, transmission time and loss of information. To achieve this objective, we have analyzed the advantages and the limitations of WPAN (wireless personal area network) technology and communication protocols currently used in relation to the requirements of our application. We then proposed a deterministic, adaptive and energy saving medium access method based on the IEEE 802.15.4 physical layer and a mesh topology. It ensures the message delivery time with strongly limited collision risk due to the spatial reuse of medium in the two-hop neighborhood. This proposal was characterized by modeling and simulation using the OPNET network simulator. We then implemented the proposed mechanisms on hardware devices and deployed a sensors network in real situation to verify the accuracy of the model and evaluate the proposal according to different test configurations.

Keywords: Wireless sensor network, adaptive protocol, medium access method, IEEE 802.15.4, multisensors network, mesh topology, quality of service, guaranteed medium access, energy saving, elderly monitoring, modeling, simulation, implementation.

Modélisation, simulation et implémentation d'un protocole de communication adaptatif dans un réseau de capteurs sans fil basé sur IEEE 802.15.4 et adapté à la surveillance de personnes à domicile

Juan LU

Résumé : Le maintien à domicile des personnes fragiles vivant seules est devenu une préoccupation majeure de santé publique dans nos sociétés modernes. Parmi les différents aspects scientifiques traités dans le domaine de la surveillance à domicile, nous nous intéressons à l'étude et à la proposition d'une solution permettant à des capteurs répartis de communiquer entre eux de façon optimale et adaptée aux contraintes spécifiques de l'application. Plus précisément, nous souhaitons construire un réseau sans fil courte portée constitué de plusieurs nœuds capteurs échangeant entre eux des données selon un protocole de communication de niveau MAC (contrôle d'accès au médium) qui optimise à la fois l'énergie, le délai de transmission et la perte d'informations. Pour cela, nous avons finement analysé les avantages et les limites des technologies WPAN (réseau local personnel sans fil) et des protocoles de communication actuellement utilisés en rapport aux exigences de notre application. Nous avons ensuite proposé une méthode d'accès au médium déterministe, adaptative et économe en énergie basée sur la couche physique IEEE 802.15.4 et une topologie maillée. Elle permet de garantir le délai d'acheminement des messages avec un risque de collisions très fortement limité, grâce à une réutilisation spatiale du médium dans un voisinage à deux sauts. Cette proposition a été caractérisée par modélisation et simulation à l'aide du simulateur de réseau OPNET. Nous avons alors implémenté les mécanismes proposés sur des dispositifs matériels et déployé un réseau de capteurs en situation réelle afin de vérifier la pertinence du modèle et évaluer la proposition selon différentes configurations de test.

Mots-clés : Réseau de capteurs sans fil, protocole adaptatif, méthode d'accès au médium, IEEE 802.15.4, réseau multicapteurs, topologie mesh, qualité de service, garantie d'accès au médium, économie d'énergie, surveillance personnes âgées, modélisation, simulation, implémentation.