Université de REIMS CHAMPAGNE-ARDENNE

ECOLE DOCTORALE SCIENCES TECHNOLOGIE SANTE (547)

THÈSE

Pour obtenir le grade de

Docteur de l'Université de Reims Champagne-Ardenne

Discipline :

Informatique

Par :

Bandar Al Ghamdi

Etude des méthodes d'ordonnancement sur les réseaux de capteurs sans fil

Soutenue le, 6 Novembre 2015

JURY

Mme. Dominique Gaiti	Professeur à l'Université de technologie de Troyes	Président
M. Hacène Fouchal	Professeur à l'Université de Reims Champagne-Ardenne	Directeur de Thèse
M. Marwane Ayaida	Maître de Conférences à l'Université de Reims Champagne-Ardenne	Co-Encadrant
M. Adlen Ksentini	Maître de Conférences (HDR) à l'Université de Rennes	Rapporteur
M. Toufik Ahmed	Professeur à l'Université de Bordeaux	Rapporteur
Mme. Lynda Mokdad	Professeur à l'Université de Paris 12	Examinateur
M. Soufiene Djahel	Maître de Conférences, Manchester Metropolitan University, UK	Examinateur

À mes Parents et mon Épouse,
sans lesquels rien n'aurait été possible.

Cette thèse est aussi dédicacé à toute ma famille qui ont toujours été là pour moi.

To my Parents and my Wife, without whom nothing would have been possible.

This thesis is also dedicated to all my family who have always been there with me.

Résumé

Les Wireless Body Area (WBAN) sont une technologie de réseau sans fil basée sur les radio-fréquences qui consiste à interconnecter sur, autour ou dans le corps humain de minuscules dispositifs pouvant effectuer des mesures (capteurs). Ces réseaux sont considérés comme les plus critiques dans les réseaux de capteurs sans fil. Ils sont basés sur des architectures de réseaux auto-organisés. Chacun des capteurs corporels reçoit ou envoie des paquets du ou au coordinateur du réseau. Ce dernier est responsable de l'ordonnancement des tâches pour l'ensemble des noeuds fils. L'ordonnancement dans les WBAN nécessite un mécanisme dynamique et adaptatif pour gérer les cas d'urgence qui peuvent se produire et permet ainsi d'améliorer les paramètres les plus importants comme la qualité de la transmission, le temps de réponse, le débit, le taux de paquets délivres, etc.

Dans ces travaux de thèse, nous avons proposé trois techniques d'ordonnancement qui sont : la méthode semi-dynamique; la méthode dynamique et la méthode basée sur la priorité. De plus, une étude sur les plateformes WBAN est présentée. Dans cette étude, nous avons proposé une classification et une évaluation qualitative des plateformes déjà existantes. Nous avons aussi étudier les modèles de mobilité en proposant une architecture permettant de les décrire. Nous avons aussi mis en place une procédure de diagnostique afin de détecter rapidement des maladies épidémiques dangereuses. Par la suite, ces différentes propositions ont été validées en utilisant deux méthodes afin de vérifier leur faisabilité. Ces méthodes sont la simulation avec OPNET et l'implémentation réelle sur des capteurs TelosB et TinyOS.

Mots Clés – WSN; WBAN; Algorithmes d'ordonnancement; Optimisation; Couche MAC; Protocoles de communication;

Abstract

The Wireless Body Area Network (WBAN) is the most critical field when considering Wireless Sensor Networks (WSN). It must be a self-organizing network architecture, meaning that it should be able to efficiently manage all network architecture requirements. The WBAN usually contains at least two or more body sensors. Each body sensor sends packets to or receives packets from the Personal Area Network Coordinator (PANC). The PANC is responsible for scheduling its child nodes' tasks. Scheduling tasks in the WBAN requires a dynamic and an adaptive process in order to handle cases of emergency that can occur with a given patient. To improve the most important parameters of a WBAN, such as quality link, response time, throughput, the duty-cycle, and packet delivery, we propose three scheduling processes: the semi-dynamic, dynamic, and priority-based dynamic scheduling approaches.

In this thesis, we propose three task scheduling techniques, Semi-Dynamic Scheduling (SDS), Efficient Dynamic Scheduling (EDS) and High Priority Scheduling (HPS) approaches. Moreover, a comprehensive study has been performed for the WBAN platforms by classifying and evaluating them. We also investigate the mobility model for the WBANs by designing an architecture that describes this model. In addition, we detail a diagnosis procedure by using classification methods in order to solve very sensitive epidemic diseases. Then, our proposals have been validated using two techniques to check out the feasibility of our proposals. These techniques are simulation scenarios using the well-known network simulator OPNET and real implementations over TelosB motes under the TinyOS system.

Keywords – WSN; WBAN; Scheduling Algorithms; Optimization; MAC layer; Communication protocols;

Acknowledgements

I would like to express my sincere gratitude to my advisor Professor Hacène Fouchal who gave me the opportunity to work on this motivating topic. I would also like to thank him for his constant support during my Doctoral Thesis.

I am also thankful to my co-advisor Dr Marwane Ayiada for his constant availability and his help. I would like to express to him my deep gratitude for his enlightening advices.

My special gratitude goes to my beloved mother Khadija, my father Ali, my dear and sweet wife Farah, all my family members for their constant love, support and encouragement.

Lastly, I offer my regards to all of those who supported me during all my study.

Please, enjoy your reading !

Bandar Ali M. Al Rami Al Ghamdi

Table des matières

Contents

Liste des tableaux

Table des figures

Introduction (Version Fançaise)

Depuis la fin du siècle dernier, nous avons observé la croissance des réseaux de capteurs sans fil (WSN), qui ont été utilisés dans de nombreux domaines. Par exemple, un WSN peut être déployé sur une zone d'agriculture à large échelle pour surveiller les changements environnementaux. Ils peuvent également être mis sur les animaux pour étudier la façon dont ils évoluent dans leur habitat naturel. Ce type de réseau pourrait également être utilisé pour la surveillance des personnes âgées. De nombreux projets de développement académiques et industriels ont été mis en œuvre afin d'améliorer la surveillance de la santé basée sur les réseaux de capteurs. L'utilisation des réseaux de capteurs sans fil dans le domaine de santé pourrait aider à surmonter la pénurie du personnels médical dans les établissements de santé à travers le monde, alors même que la population humaine continue d'augmenter d'année en année. En outre, le nombre de personnes âgées de plus de 65 ans sera de 761 millions en 2025 [1]. Cela signifie qu'il y aura des obstacles à la fourniture d'un système de santé efficace. Trouver des solutions viables risquent de ne pas être facile, sans l'utilisation de nouvelles technologies telles que les réseaux de capteurs. Les systèmes d'automatisation de soins de santé basés sur les réseaux de capteurs sans fil sont appelés les réseaux corporels (WBANs). Les WBANs fournissent un moyen efficace qui permet d'améliorer la qualité des systèmes de soins de santé.

Les WBANs aident les patients à pratiquer leurs activités quotidiennes plus facilement qu'auparavant. En outre, la surveillance des patients peuvent être effectuées en dehors des établissements médicaux. Un système bien conçu peut aussi garder les patients plus à l'aise lors de leurs procédures de contrôle.

Motivations et objectifs

Dans les WBANs, le corps humain est composé d'un nombre d'objets qui sont reliés entre eux sans fil. Chaque objet a plusieurs capteurs qui sont reliés entre eux en sans fil. Tous les échantillons correspondent à certains événements environnementaux ou physiologiques. A titre d'exemples on peut citer: les signes vitaux pour un patient particulier ou le mouvement du corps humain et ainsi de suite.

Habituellement, chaque capteur exécute le même travail pendant une période bien définie. Le WBAN répète continuellement ses calculs pendant toutes les périodes. Cette période pourrait être fixée à l'avance ou pourrait varier. Avant de commencer, chaque nœud doit connaître ses slots de temps associés. Pour cette raison, un processus de planification est requis pour trouver la solution d'ordonnancement la plus appropriée (quand elle existe).

Cette étude vise à concevoir trois algorithmes d'ordonnancement multi-sauts pour les topologies de type arbre (cluster-tree) basés sur ZigBee / IEEE 802.15.4 en mode beacon-enabled. En outre, ils s'intéressent aux événements prioritaires dans les communications entre les capteurs. Ces algorithmes réalisent des tâches différentes en commençant par le support des réseaux de capteurs en général, et en finissant par la gestion des situations critiques dans les WBANs. Une topologie cluster-tree (arbre) basée sur le protocole ZigBee / IEEE 802.15.4 favorise deux caractéristiques importantes, la synchronisation et multi-sauts, car il fonctionne en mode beacon-enabled. Dans les normes IEEE 802.15.4, il n'y a aucune solution claire qui adresse comment activer efficacement ces deux caractéristiques. Même si certaines études ont proposé divers algorithmes afin de résoudre ces problèmes, les algorithmes eux-mêmes ne sont pas toujours applicables et ont besoin d'améliorations supplémentaires. L'importance de ces caractéristiques nous ont guidé à proposer trois différents algorithmes d'ordonnancement. Le concept principal est l'amélioration des algorithmes d'ordonnancement qui seront utilisés dans différentes applications. En outre, les algorithmes pourraient optimiser la qualité de service dans ce domaine.

Challenges

Les Sensor Wireless Devices (WSDs) sur le corps humain effectuent un certain nombre de tâches. Par exemple dans le domaine de la santé, ils pourraient permettre aux personnels médical de surveiller à distance un patient particulier dans sa maison et ses signes vitaux dans les hôpitaux, etc. Ces dispositifs sont interconnectés sans fil via un certain nombre de protocoles tels que ZigBee / IEEE 802.15.4 , Bluetooth, 4G, etc. Les protocoles des WSNs ne sont pas toujours prêts à l'emploi dans le domaine de la santé. La communication entre les WSDs doit être bien organisée (ordonnancée) afin d'éviter les collisions. L'échange des messages entre WSDs se fait via deux types des protocoles MAC : organisée (ordonnancée) ou non-organisée (randomisée). Un WSD dans le protocole MAC non-organisé enverra ses données après avoir écouté le médium. Lorsque le médium devient occupé, le WSD attend pendant une période aléatoire avant de tenter à nouveau jusqu'à ce que le médium devient disponible. Le principal inconvénient de ce mécanisme est le fait que chaque nœud du réseau devra augmenter sa consommation d'énergie puisque tous les capteurs doivent écouter le médium. D'autre part, le WSD dans le protocole MAC organisé transmet une requête d'envoi (RTS) et reçoit une autorisation à émettre (CTS). Ceci permet à tous les capteurs dans la même rangée de transmission de transmettre périodiquement dans des intervalles de temps définis.

Ce mécanisme utilise une référence commune sur le coordinateur pour organiser le temps de communication pour chacun des capteurs afin d'éviter les collisions avec d'autres. Le service de coordination qui est représenté par le coordinateur du réseau a pour but de synchroniser les périphériques connectés sans fil. Pour être beaucoup plus précis et afin d'examiner un tel mécanisme, la technologie de WSN dans sa forme commerciale doit être considérée. Ainsi, la technologie ZigBee est classée comme l'une des technologies les plus prometteuses dans ce domaine. La technologie ZigBee est basée sur la norme IEEE 802.15.4, qui a quelques caractéristiques intéressantes, y compris sa faible consommation d'énergie, son faible coût, sa convenance pour de nombreuses applications, et le support d'applications à large échelle. La ZigBee

/ WSN supporte trois types de topologies: la topologie étoile, la topologie arbre (cluster-tree), et la topologie maillée (Mesh). Il dispose de deux modes principaux : le mode beacon-enabled et le mode non-beacon-enabled. Les topologies étoile et arbre sont gérées par le mode beacon-enabled, tandis que la topologie maillée est uniquement pris en charge dans le mode non-beacon-enabled.

Les applications de santé basées sur le protocole ZigBee / IEEE 802.15.4 nécessitent des algorithmes intelligents qui maintiennent, identifient et répondent aux normes de qualité médicale. La considération la plus importante dans ce domaine est le fait de savoir si un algorithme est dynamique (adaptatif), temps réel, mobile et évolutif. Selon la nature du corp humain, les topologies en étoile ou en arbre sont les plus adaptées par rapport à la topologie maillée. Par conséquent et en se basant sur ZigBee, ces deux topologies travaillent en mode beacon-enabled, ce qui signifie qu'il y a un coordonnateur unique (PANC) du réseau qui organise les communications parent-enfant. Cela signifie que le PANC a un rôle crucial dans ce mode. La participation à la communication des WSD doit être effectuée selon un procédé de gestion de temps bien organisé par le PANC. Dans ce cas, personne ne peut transmettre sans l'autorisation préalable du PANC. Par la suite, trames balise (beacon) sont responsables de la synchronisation des WSDs. Pour cette raison, un algorithme d'ordonnancement doit être correctement conçu afin de gérer efficacement le trafic de données. La communication en temps réel dans un tel domaine sensible est critique et très difficile à mettre en oeuvre. Plusieurs études ont proposé de résoudre ce problème grâce à l'utilisation de ressources telles que TDBS (Time Division Beacon Scheduling) dans [2] . Le principal inconvénient de cette solution est la nécessité de fixer le rapport cyclique pour chaque arbre de clusters, ce qui augmente la latence globale. Cet effet ne convient pas pour la conception d'un réseau à grande échelle et ne supporte pas la mobilité, car c'est un processus d'ordonnancement statique à la base. Le processus d'ordonnancement statique basé sur les travaux dans [2] ne résout les problèmes d'ordonnancement que pour un petit environnement fixe. Dans le même contexte, le processus d'ordonnancement dynamique démontre l'inconvénient de la fixation du rapport cyclique pour chaque cluster de l'arbre tout en supportant à la fois l'environnement à large échelle et la mobilité des nœuds.

Contributions

Notre première contribution dans WSN et WBAN était de proposer une nouvelle stratégie pour l'apprentissage supervisé dans la détection et la description de cas de différents stades de la maladie et surtout dans les cas d'épidémies. Ensuite, nous avons proposé trois approches d'ordonnancement afin de résoudre les inconvénients qui sont causés par la communication non organisée entre les dispositifs. L'interaction entre ces dispositifs doit être programmée de manière efficace. Les trois algorithmes d'ordonnancement sont basés sur le protocole ZigBee / IEEE 802.15.4. Les deux premiers sont des algorithmes pour des applications de type WSN en général, tandis que le troisième est pour les applications de type WBAN. Nous les désignons comme : l'ordonnancement semi-dynamique (Semi-Dynamic Scheduling SDS), l'ordonnancement dynamique (Efficient Dynamic Scheduling EDS), et l'ordonnancement à base de priorité (High Priority Scheduling HPS), respectivement.

La première approche est appelée Semi-Dynamic Scheduling (SDS) et est composée de deux types d'ordonnancement en mélangeant le statique et le dynamique. L'idée est de les fusionner afin de fixer les capteurs de haut niveau dans le réseau (coordinateur et routeurs), et à auto-organiser le reste des dispositifs de réseau (noeuds à l'extrémité de l'arbre). Ce mécanisme permet aux nœuds feuilles d'interagir avec leurs parents dans d'une manière plus simple et plus rapide. Les capteurs de haut niveau sont prédéfinis, à savoir, ils sont affectés par des adresses prédéfinies et les intervalles de temps fixes. En revanche, les capteurs de bas niveau se voient attribués une adresse basée sur le modèle de trafic réel. Ainsi, l'approche semi-dynamique automatise le processus d'ordonnancement pour les périphériques de bas niveau.

La deuxième approche est appelée Efficient Dynamic Scheduling (EDS) et fournit un processus d'ordonnancement entièrement automatisé qui permet une communication auto-adaptative entre les capteurs de haut niveau et bas niveau dans l'arbre. Ce mécanisme permet de gérer le processus d'ordonnancement afin de supporter au mieux le modèle de mobilité ainsi que des applications à large échelle.

Le troisième processus d'ordonnancement est High Priority Scheduling (HPS), qui est une extension issu de l'ordonnancement dynamique afin d'optimiser et d'améliorer la communication pour les événements prioritaires dans les cas critiques. Le processus d'ordonnancement basé sur la priorité est dédié aux applications WBAN.

Les algorithmes d'ordonnancement ont été validés en utilisant deux techniques. La première a été utilisée afin de développer trois scénarios différents pour vérifier la faisabilité des algorithmes proposés. Ces scénarios ont été développés sur le simulateur réseau OPNET. Des modifications ont été apportées à la couche MAC reprenant la notion dynamique et pour supporter la structure théorique de notre Superframe. La Superframe proposée est conçue afin de gérer plusieurs messages prioritaires dans les situations critiques. La simulation prend en charge un nombre limité de capteurs en raison de la limitation de la version de la licence d'éducation d'OPNET.

La seconde technique est une véritable mise en œuvre en utilisant le langage de programmation NesC basé sur l'environnement de TinyOS, qui est une plateforme libre et open-source. TinyOS est un système d'exploitation qui est écrit dans la langue NesC qui est en fait une extension du langage C. Il a été développé à pour la prise en charge de l'intercommunication temps réel entre capteurs ainsi que pour répondre à la limitation de la taille mémoire des capteurs. Les sorties ont été obtenus en utilisant notre version étendue de l'outil Z-Monitor [3] que l'on a appelé B-Monitor. Quelques caractéristiques importantes ont été ajoutées afin de valider nos résultats, tels que la séparation des identifiants des capteurs pour vérifier la performance de chaque capteur séparé, le résumé des statistiques détaillé suivant l'identifiant du capteur. En outre, l'exécution de l'outil B-Monitor est automatiquement programmé avec une fonction limite de temps. Les algorithmes d'ordonnancement sont exécutées au cours de fonctionnement dans le PANC. La démonstration de la faisabilité a été réalisée en utilisant un nombre limité de dispositifs en raison du coût élevé des capteurs qui sont disponibles dans le commerce.

Tout d'abord, nous avons proposé trois algorithmes d'ordonnancement pour les WSN et WBAN. Les algorithmes proposés sont élaborés en se basant sur le protocole ZigBee / IEEE 802.15.4 en mode beacon-enabled. En outre, les algorithmes sont conçus afin de supporter le réseau en arbre multi-sauts.

En outre, pour la comparaison avec nos résultats, nous avons comparé ces algorithmes contre un algorithme d'ordonnancement bien connu appelé TDBS (Time Division Beacon Scheduling). Les comparaisons ont été effectuées sur la base de critères importants tels que le taux de paquets délivrés (PDR), le débit et la latence (délai). Les résultats de la comparaison confirment la faisabilité de notre approche.

Les résultats obtenus lors de cette étude ont permis de vérifier et de prouver que le processus dynamique est plus efficace que l'approche statique pour des applications critiques. Il reflète un niveau élevé de traitement en termes de flexibilité de communication couplée avec des hautes performances.

Puis, nous nous sommes concentrés sur l'approche d'ordonnancement dynamique à haute priorité, qui permet de prendre en charge le modèle de mobilité. Les résultats des expérimentations ont confirmé que l'approche dynamique à base de priorité est plus efficace que l'approche dynamique seule.

Enfin, il est clair que ces propositions d'ordonnancement ont amélioré les performances du réseau en terme de PDR, de débit et de latence. De plus, la combinaison d'ordonnancements (dynamique et statique) a montré une amélioration notable par rapport à celle de l'ordonnancement statique. Ceci nous a conduit à proposer une autre approche basé sur un modèle entièrement dynamique. Ainsi, l'approche dynamique a confirmé ces caractéristiques en améliorant les performances du réseau par rapport à l'approche semi-dynamique. En outre, la version dynamique a amélioré la prise en charge de la mobilité globale des nœuds capteurs et a aussi révéler la nécessité d'améliorer la prise en charge des événements prioritaires pour les cas critiques. Cette question a propulsé cette étude dans la conception d'un modèle d'ordonnancement dynamique basée sur la priorité.

Organisation de la Thèse

La thèse se compose de six chapitres. Le premier chapitre contient les motivations et les objectifs. Aussi, il aborde les défis de cette étude de travail et présente les grandes lignes de cette thèse.

Le deuxième chapitre présente les travaux connexes qui sont divisés en deux sections principales. La première section se concentre sur un aperçu des réseaux de capteurs. Il fournit également des définitions des différentes normes et protocoles de réseaux de capteurs. Ensuite, il détaille le protocole ZigBee / IEEE 802.15.4 en décrivant la formation de l'arbre et aborder la façon dont les affectations des adresses sont faites sur la base de la structure de l'arbre. La deuxième section de ce chapitre présente des informations générales sur les WBANs. Il met en évidence des travaux connexes sur des plateformes WBAN. Il montre également les protocoles MAC qui sont utilisés dans les WBAN.

Le troisième chapitre présente le modèle de mobilité dans les WBAN. Ensuite, les pré-requis des plateformes WBAN sont présentés. Les plateformes WBAN ont été classés et évalués sur la base de paramètres spécifiques. La dernière partie de ce chapitre élabore notre propre proposition pour le diagnostique d'une maladie et le suivi d'épidémies.

Le quatrième chapitre de la thèse est divisé en plusieurs sections qui introduisent l'ordonnancement des tâches sur les réseaux de capteurs. Des affirmations sont discutées sur la base des différentes définitions du mécanisme d'ordonnancement. Ensuite, les métriques d'ordonnancement et les objectifs des processus d'ordonnancement sont présentés. Ce chapitre détaille également les approches d'ordonnancement dans les réseaux ZigBee / IEEE 802.15.4. Les deux dernières sections présentent nos deux principales approches d'ordonnancement qui sont l'ordonnancement semi-dynamique (SDS) et l'ordonnancement dynamique (EDS). Enfin, ce chapitre se termine par quelques évaluations des performances basées sur des résultats de simulation et des résultats expérimentaux.

Dans le cinquième chapitre, l'ordonnancement des tâches dans les WBANs est présenté à travers l'introduction de notre proposition. Notre nouvelle approche, axée sur l'ordonnancement des tâches prioritaires qui ont été générés à partir des différents périphériques, est mise en exergue. Ce chapitre se termine aussi avec certaines évaluations de performance basées sur des résultats de simulation ainsi que des résultats expérimentaux.

Le dernier chapitre met en évidence les perspectives de ce travail et les travaux futurs de notre recherche.

Introduction (English Version)

Since the end of the last century, we have observed the growth of Wireless Sensor Networks (WSNs), which have been used in many domains. For example, a WSN may be deployed over a wide agriculture area to monitor environmental changes. They can also be put on animals to study how they evolve in their natural habitat. WSNs could also be used for monitoring the elderly. Many academic and industrial development projects have been implemented to improve healthcare monitoring based on WSNs. Using WSNs in healthcare might help to overcome the shortage of medical staff in medical institutions around the world, even as the human population continues to increase year by year. Moreover, the number of elderly individuals over the age of 65 will be 761 million by 2025 [1]. This means that there will be obstacles to address in order to provide an efficient healthcare system. Finding viable solutions might not be easy without the use of new technologies such as WSNs. Healthcare automation systems based on WSNs are called Wireless Body Area Networks (WBANs). WBANs provide an efficient mean to enhance the healthcare systems quality.

WBANs help patients practice their daily activities more easily than before. Moreover, monitoring patients can be done outside of medical establishments. A well-designed system can also keep patients more comfortable during their monitoring procedures.

Motivations and objectives

In WBANs, the human body is understood as a number of objects that are interconnected wirelessly. Each object has further elements that are interconnected wirelessly called Body Sensor Devices (BSDs). Every element samples some environmental or physiological events. As examples we can cite: the vital signs for a particular patient or the movement of the human body and so on.

Usually, each BSD executes the same job during a defined period. The WBAN continuously repeats its computations during a period. This period could be fixed in advance or could vary. Before starting, each BD should know its associated time slots. For this reason a smart scheduling process is required to find the most appropriate scheduling solution (when it exists).

This research study aims to design three multi-hop scheduling algorithms for cluster-tree topology based on ZigBee/IEEE 802.15.4 in beacon-enabled mode. Moreover, it focuses on priority events control in communications between sensor devices (SDs). These algorithms have different tasks started by supporting WSNs in general and ended by supporting critical situations in WBANs. A cluster-tree topology based on ZigBee/IEEE 802.15.4 protocol enables two important features, synchronization and multi-hop, since it works in beacon-enabled mode. In the IEEE 802.15.4

standards, there is no clear solution addressing how to enable these two features efficiently. Even though some studies have proposed various algorithms to solve these issues, the algorithms themselves are not always applicable or need additional improvements. The importance of enabling these features guided us to propose three different scheduling algorithms. The main concept is to improve the scheduling algorithms that will be used in different applications. Moreover, the algorithms would optimize the QoS in this area.

Challenges

Wireless Sensor Devices (WSDs) on the human body perform a number of tasks. For instance in the healthcare field, they could allow medical personnel to remotely monitor a particular patient at home, sample the vital signs of patients in hospitals, etc. These devices are interconnected wirelessly via a number of protocols such as ZigBee/IEEE 802.15.4, Bluetooth, 4G, etc. The WSNs complaint protocols are not always ready for use in the healthcare field. Communication between WSDs has to be well organized (scheduled) in order to avoid collisions. Message exchange between WSDs is done via two types of medium access control (MAC) protocols: scheduled (organized) or unscheduled (randomized). The WSD in the unscheduled MAC protocol will send its data after listening to the medium. When the medium becomes busy, the WSD waits for a period before trying again until the medium becomes clear. The main drawback of this mechanism is the fact that every SD in the network will need to increase power consumption since all the SDs have to listen to the medium. On the other hand, the WSD in the scheduled MAC protocol performs a request to send (RTS) and clear to send (CTS) mechanism, which allows all SDs in the same transmission range to work periodically within defined time slots This mechanism uses a common schedule on the coordinator to arrange communication time for each device in order to avoid collisions with other devices. The coordinating service that is represented by the network coordinator has to synchronize the attached devices wirelessly. To be much more specific and examine such an interesting mechanism, WSN technology in its commercial off-the-shelf (COTS) form must be considered. Thus, ZigBee technology is classified as one of the most promising technologies in this field. ZigBee technology is based on the IEEE 802.15.4 standard, which has some interesting features, including its low power consumption, low cost, suitableness for many applications, and support of large-scale environments. The ZigBee WSN supports three types of topologies: star, cluster-tree, and mech. It has two main modes: beacon-enabled mode and non-beacon-enabled mode. The star and cluster-tree topology work in beacon-enabled mode, while the mesh topology is only supported in the non-beacon-enabled mode.

Healthcare applications based on the ZigBee/IEEE 802.15.4 protocol need smart algorithms that maintain, identify, and meet medical quality standards. The most important considerations in this domain are whether an algorithm is dynamic (adaptive), real-time, mobile, and scalable. According to the nature of the human body, star or cluster-tree topologies are more suitable than a mesh topology. Therefore based on ZigBee, both of them work in beacon-enabled mode, which means that there is a unique Personal Area Network Coordinator (PANC) that arranges its parent-child devices. This means that the PANC has a crucial role in this mode. WSD participation must be undertaken according to a time scheduling process by the PANC. In this case, no one can speak without prior permission from the PANC. Thereafter, beacon frames are responsible for synchronizing WSDs. For this reason, a scheduling algorithm must be properly designed to efficiently handle a given traffic pattern. Real-time communication in such a sensitive domain is critical and very challenging. Several studies have proposed to solve this problem through the use of resources such as TDBS in [2]. The main drawback of this solution is the need to fix the duty-cycle for each cluster tree, which increases overall latency. This effect is not suitable for large-scale networking and does not support mobility because it is a static-based scheduling process. The static-based scheduling process in [2] only solves the scheduling process for a small and fixed environment. In the same context, the dynamic scheduling process shows the needlessness of fixing the duty-cycle for each cluster-tree while supporting both the application in a large-scale environment and a specific SDs mobility feature.

Contributions

Our first contribution in WSN and WBAN was to propose a new strategy for supervised learning in the detecting and describing of instances from different stages of disease in the case of epidemics. Then, we propose three different scheduling approaches to solve the drawbacks that are caused by the non organized communication between the devices. The interaction between these devices has to be scheduled efficiently. The three smart scheduling algorithms are based on the ZigBee/IEEE 802.15.4 protocol. The first two algorithms are for WSN applications in general, while the third one is for WBAN applications. We denote them as the Semi-Dynamic Scheduling (SDS), Efficient Dynamic Scheduling (EDS), and High Priority Scheduling (HPS) algorithms, respectively.

The first approach is called the Semi-Dynamic Scheduling (SDS) and is composed of two types of scheduling by mixing the static with the dynamic. The idea is to merge them in order to fix high-level devices in the network (coordinator and routers), and to self-organize the rest of the SDs (leaf nodes). This mechanism allows the leaf nodes to interact with their parents in a simpler and faster way. The high-level devices are predefined, i.e., they are assigned by predefined addresses and time slots. In contrast, low-level devices are assigned an address based on the traffic pattern. Thus, the semi-dynamic approach automates the scheduling process for low-level devices.

The second approach is called the Efficient Dynamic Scheduling (EDS) and provides a fully automated scheduling process that allows self-adaptive communication between high-level and low-level devices. This mechanism drives the scheduling process in order to support the mobility model as well as large-scale applications.

The third scheduling process is High Priority Scheduling (HPS), which is extended from dynamic scheduling in order to optimize and improve communication for priority events in critical cases. The priority-based scheduling process is dedicated to WBAN applications.

The three scheduling algorithms were developed based on the System Development Life Cycle (SDLC) phases. Theses phases include investigation, analysis, design, environments, and testing and evaluation, respectively. They are used to verify the proposed models by following the SDLC principle to produce a high quality work.

The scheduling algorithms have been validated using two techniques. The first one is used to develop three different scenarios for verifying the feasibility of the proposed algorithms. These scenarios are developed over the OPNET modeler. Some modifications have been made to the logical MAC layer to fit the dynamic concept and to support the theoretical SuperFrame structure. The proposed SuperFrame is designed to handle more priority messages in critical situations. The simulation supports a limited number of SDs due to the licensing limitation of the educational version of OPNET.

The second technique is a real implementation using the nesC (network embedded systems C) programming language based on the TinyOS environment, which is a free and open-source platform. TinyOS is an operating system that is written in the nesC language, which is actually an extension of C language. It has been developed to support a sensor's real-time intercommunication as well as to suit the limitation of the sensor's memory. The outputs have been obtained using our extended version of the original Z-Monitor [3] sniffer tool that is called a B-Monitor tool. Some important features have been added to validate the results, such as separate sensor identities to check out the performance of each sensor. The summary of statistics is detailed to support the ID separations. Also, the runtime for the sniffer tool is automatically programmed with a time limit feature. The scheduling algorithms are executed over the PANC to promote its sensor activities. The demonstration is done using a limited number of devices due to the high cost of the SDs that are commercially available.

Firstly, we proposed three scheduling algorithms for both WSN and WBAN platforms. The proposed algorithms are developed based on the ZigBee/IEEE 802.15.4 protocol in beacon-enabled mode. Besides, the algorithms are designed to support the cluster-tree multi-hop network.

Additionally, for comparison with our results, we measured the proposed algorithms against a well-known scheduling algorithm called Time Division Beacon Scheduling (TDBS). The comparisons were made based on important criteria such as the packet delivery rate (PDR), throughput, and latency (delay). The results of the comparison confirm the feasibility of our approaches.

The results obtained during this study verify and prove that the dynamic-based scheduling process is more efficient than the static-based approach for critical applications. It reflects a high level of processing in terms of communication flexibility coupled with high performance.

Then, we focused on the high-priority dynamic scheduling approach, which enables the mobility model. The test bed experimentation results have confirmed that high-priority dynamic-based scheduling is more efficient than dynamic-based scheduling.

Finally, it is clear that the scheduling proposals have improved network performance in terms of PDR, throughput, and latency. Besides, the scheduling combination has shown a noticeable improvement above that of static-based scheduling, which encourages us to take another step forward in developing a fully dynamic model. Thus, dynamic-based scheduling has confirmed this mission by increasing network performance more effectively than semi-dynamic-based scheduling. Additionally, the dynamic version has enabled overall mobility to the SDs, which reveals the need to improve the guaranteed events for critical cases. This issue has propelled this study to design a high-priority dynamic scheduling model.

Thesis Outline

The thesis is composed of six chapters. The first chapter contains the motivations and objectives. Besides, it addresses the challenges of this work study. Also, it presents the thesis outline.

Second chapter discusses the related works, which are divided into two main sections. The first section focuses on an overview of WSNs. It also provides different definitions for the standards and protocols of WSNs. Next, it details the ZigBee/IEEE 802.15.4 protocol by describing Cluster-tree formation and addressing how assignments are made based on the tree structure.The platforms of the WBAN have been classified and evaluated based on specific parameters. The last section of this chapter presents background information on WBANs. It highlights related work on WBAN platforms. It also shows the WBAN MAC protocols.

Third chapter presents the WBAN mobility model, which is explained in detail. Next, the platform requirements and considerations of a WBAN are presented. The last section of this chapter elaborates our proposal for disease diagnosis and epidemic tracking.

The fourth chapter of the thesis is divided into several sections that introduce task scheduling over WSNs. Statements are made based on various definitions of the scheduling mechanism and scheduling assumptions are discussed. Next, both the scheduling metrics and the policy goals of the scheduling process are presented. This chapter also details the scheduling approaches taken over cluster-tree networks with regard to ZigBee/IEEE802.15.4. The last two sections introduce our two main scheduling approaches, which are the Semi-Dynamic Scheduling (SDS) and the Efficient Dynamic scheduling Scheduling (EDS) approaches, respectively. Finally, this chapter concludes with some performance evaluations based on simulation results and experimental results.

In the fifth chapter, task scheduling over a WBAN is presented through the introduction of our proposal. Our new approach focused on the scheduling of priority tasks that have been generated from different devices is highlighted. This chapter also concludes with some performance evaluations based on simulation results and experimental results.

The last chapter highlights the outlook of this work. It finally introduces the future work of our research.

Chapter 1

Related Works

Contents

1.1 Overview of Wireless Sensor Networks (WSNs)

Wireless networking technologies have seen incredible improvements over time. Sensors are embedded in wireless network systems, which are called wireless sensor networks (WSNs). In recent years, the growth of WSNs has been observed. Wireless sensors (WSs) provide a new way to perform unusual tasks. Typically, WSs can be deployed in a variety of areas to monitor environmental changes, such as temperature changes or luminance changes. They can also be tagged on animals to study how they evolve in their natural habitats. A well-studied field is patient monitoring. In this context, three main scenarios are known. One consists of monitoring a patient in an emergency site (in-situ monitoring), another one deals with an in-hospital monitoring scenario, and the third is related to monitoring individuals in their homes (in-home monitoring).

In the same context, SDs need to be interconnected wirelessly in order to exchange messages with each other. The process of exchange is a complicated issue due to a number of considerations, such as choosing a suitable communication protocol, managing duty-cycles, maintaining the synchronization process (using beacon frames), and improving the SuperFrame structure. These considerations are important in order to avoid transmission collisions caused by scheduling conflicts. Thus, a critical question should be answered: how can smart scheduling algorithms that will efficiently organize a device's tasks based on traffic patterns be created? This question must be taken into account alongside other priorities.

The new scheduling algorithms must be able to manage network resources in an adaptive manner. They should also enable the mobility model to support outdoor applications such as those to be used in the healthcare domain for monitoring patients outside their homes. Furthermore, they should support time-critical applications in order to handle priority events in critical situations. These situations most often arise in life and health-related applications.

The importance of this study comes from its impact on freedom. For a particular patient that is equipped with some SDs, daily activities will need to be done without any restrictions. The freedom to live a rich life is a fundamental right of humans, which should be reflected in people's work. For this reason, developing applications that will make it easier for people to live their lives is very important. Thus, freedom of lifestyle is a component of this study. Three different approaches are proposed to overcome the stated drawbacks.

This chapter is divided into two main sections. The first section presents an overview and general aspects of WSNs as well as some common protocols that have been used in WSN applications. The second section analyzes WBAN platforms by studying some important related works and the MAC protocols that have been used in the WBANs. The subsequent subsections present the WBAN platform architecture and the mobility model. Additionally, some important considerations have been investigated to classify and evaluate the related work platforms.

1.1.1 Related Work on WSNs

This section talks about some MAC protocols that are based on WSNs. We will briefly explain the characteristics of each relevant work. In general, the MAC solutions being considered are focused on energy consumption. In fact, this issue is one of the most challenging parameters with regard to WSNs.

In [4], the author proposes a smart collision-free MAC layer protocol for WSNs. The authors present a time-based scheduling technique to ensure efficient access to the medium without any collisions. Since the scheduling approach is proactive, each node turns off its radio when it is not needed.

Time division scheduling is also considered in [5]. Indeed, this approach investigates time division as well as end-to-end delay. In this paper, a balanced use of the medium is guaranteed since wake-up intervals are scheduled in order to permit data packet delivery at a maximum rate after one sleep period.

The Dynamic Power Management (DPM) issue was also considered in many papers in order to propose specific architectures based on WSNs. For example, [6] works on a novel WSN architecture that handles smart transitions from one sleep period to another depending on the specifics of given situations. The main purpose of this study is to ensure a high level of energy conservation for sensors, one that has longer sleep periods with low end-to-end delay.

In [7], the authors studied how to reduce power management at the application level. They implemented a simple distributed solution via MicaZ sensors in order to obtain longer sleeping periods when needed. In both [8] and [9], the authors propose an efficient MAC protocol that guarantees very low energy consumption. An accurate performance evaluation procedure validates the protocol.

1.1.2 On Standards and Protocols of WSNs

Recent years have reflected too much effort being spent on developing suitable WSN standards that address many fundamental issues. The IEEE 802.14.5 standard is considered the most essential project in this domain. This project was created by the IEEE 802.15 task group 4 (TG4) in 2003. They focused on designing a new standard for low-rate wireless personal area networks (LR-PANs) that define two main layers: the physical (PHY) layer and the media access control (MAC) layer.

Definition A WSN is a network that consists of a number of SDs interconnected wirelessly to monitor a specific environment by sampling data. The collected data is then transferred and processed at the base station.

WSNs should be supported by self-organizing algorithms. Moreover, the protocols that will be used in a WSN must be capable of managing these algorithms. For instance, in the medical domain, WSNs can be used to continuously monitor a particular patient by sampling the individual's vital signs, such as temperature and blood pressure.

Some specific features of a WSN include the fact that they consume power, have self-solving nodes in case of failure, can support node mobility, can solve communication problems, have supporting dynamic algorithms, and are created for large-scale deployment.

Currently there are four main types of sensors: temperature, light, sound, and vibration sensors.

The ZigBee and IEEE 802.15.4 Protocol Standard

The IEEE 802.15.4 protocol is a part of the IEEE 802.15 family. This general family includes the most common standard, IEEE 802.15.1, which is as Bluetooth. The OSI model from this family contains an APP layer, a NWK layer, a PHY layer, and a MAC layer. As stated before, the IEEE 802.15.4 standard defines only the two lowest-level layers. High-level layers are defined subsequently by other protocols, such as ZigBee technology (see Figure 1.1).

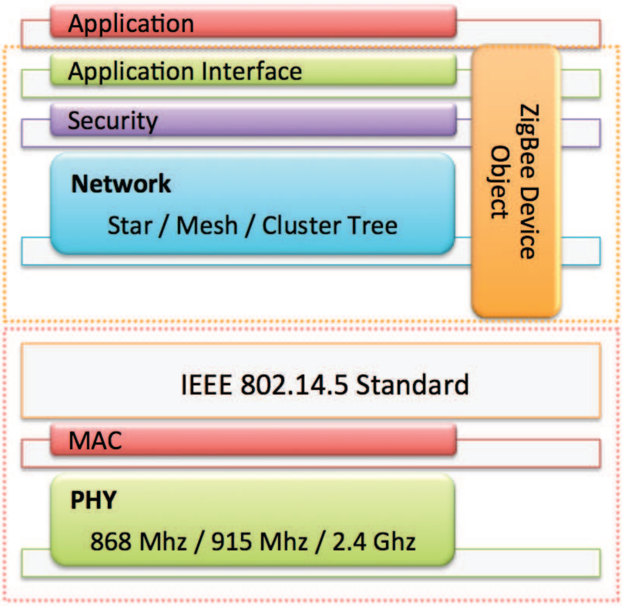


Figure 1.1: OSI Model of ZigBee/IEEE 802.15.4

NWK Layer In the NWK layer the nodes joining and disjoining can be managed. The NWK is also responsible for managing routing and security. Moreover, any information from the table of a neighbor of each node is managed and maintained by the NWK. This layer provides some services, including the network layer data entity service access point (NLDE-SAP), which provides a data exchange service with the APS. There are two types of devices that are defined by ZigBee. A full function device (FFD) can be used for two tasks. Firstly, it can be used as the PANC, and secondly, it can be used as a router, which works as a relay node (routing) to forward packets between the PANC and its parent-child end devices. The second type of ZigBee device is a reduced function device (RFD) that can be used as leaf nodes, which are employed to sample environmental data only.

MAC Layer The MAC layer of IEEE 802.15.4 provides two main services. According to the standards, the first service is the data service and the second one is the management service. This service interfaces the MAC sub-layer management entity (MLME) service access point (SAP), MLME-SAP. The purpose of the MAC data service is to manage the transmission and reception of MAC protocol data units (MPDU) via the PHY data service.

The guiding principles of the MAC layer are to manage beacon frames, provide guaranteed time slots, validate frames, provide acknowledgment frames to report delivery, and manage the network devices association and disassociation services.

The superframe is divided into two main portion, active and inactive period. The active portion is further divided into two parts, the contention access period and free contention period which is the guaranteed time slots (GTS). The active period is divided into 16 time slots in equitable way. See Figure 1.2.

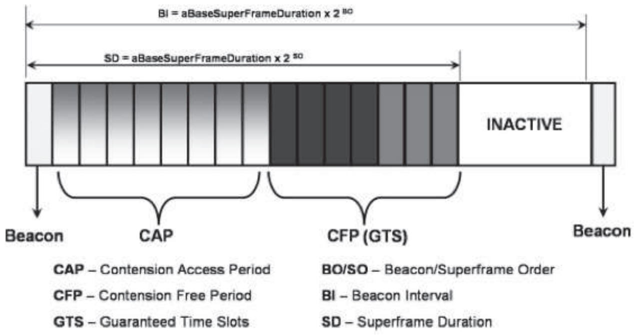


Figure 1.2: Superframe of ZigBee/IEEE 802.15.4

PHY Layer The PHY layer of IEEE 802.15.4 provides two main services. According to the standards, the first service is the data service and the second one is the management service. This service interfaces the PHY sub-layer management entity (PLME). The purpose of the PHY data service is to manage the transmission and reception of PHY protocol data units (PPDU) via the PHY radio channel.

The guiding principles of the PHY layer are to turn on/off the radio transceiver, perform energy detection (ED), provide a link quality indication (LQI), do channel selection, initiate clear channel assessment (CCA), and transmit and receive packets via the medium.

In MAC IEEE 802.15.4, channel access is determined by using the CSMA/CA algorithm. There are three different options for frequency bands in PHY IEEE 802.15.4. The first one consists of only 1 channel that is a European channel using 868 to 868.8 MHz. The second one has 30 channels that are used in Canada and the United States of America, ranging from 902 to 928 MHz. The third possibility has 16 channels that can be used internationally for medical, industrial, and scientific purposes; they occur in the range of 2.400 to 2.4835 GHz.

Star topology is also called centralized topology, which has only one main coordinator that is surrounded by some devices. There is only one link between the

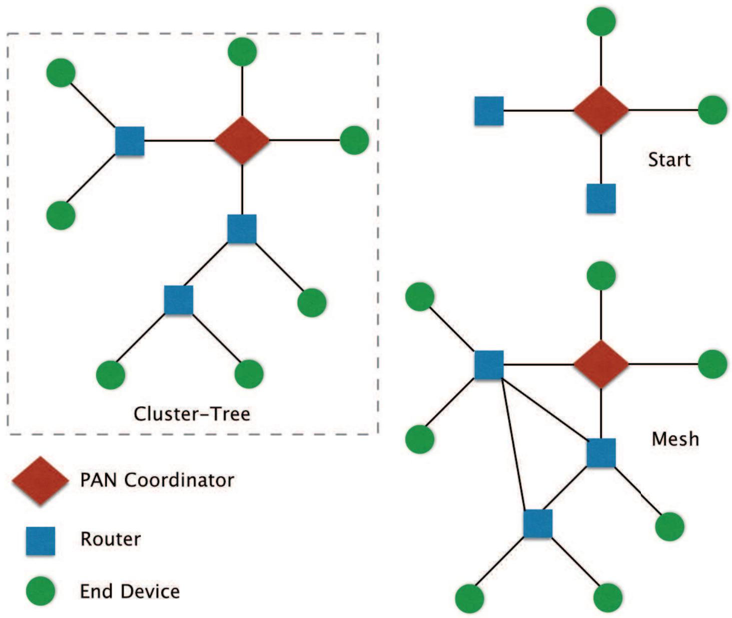


Figure 1.3: The ZigBee Toplogies

PANC and its child nodes. This topology has two main drawbacks. Firstly, it limits the application area. It can only be used for a small-scale environment so it not scalable. Another drawback is that the PANC can consume a lot of energy because it must be always active to synchronize with its child nodes.

In a mesh topology, the network is somehow more complex and is not well-organized. In this topology, all devices can create a link with each other when they are located in the same radio range. There is a unique PANC that is responsible for the network. The mesh topology is more flexible than the star topology but it is more complex in terms of node communications. It supports multi-hop communications and routing service in an ad hoc manner. It is much more efficient in terms of power consumption because it does not depend on the PANC in its interconnectivity.

The cluster-tree topology works in beacon-enabled mode. In this topology, communication between the nodes is done by creating a unique link (path). It is maintained in a parent-child connectivity that is supported by a synchronization service (beacon-mode). The network synchronization service is maintained by using beacon frames during the message exchange process. The beacon frames are generated by any FFD in the network. Hence, the PANC and its parent-child routers are allowed to send beacon frames.

The synchronization process in a cluster-tree has some disadvantages. Firstly, it allows the tree devices to reduce energy consumption by managing their duty-cycle efficiently. This process dynamically enables both the active and inactive modes on each device in the network.

Additionally, beacon frames are meant to dynamically manage the network duty-cycle. Hence, they enable the dynamic allocation of the guaranteed bandwidth. This feature highlights priority devices in order to avoid the contention period by sending data directly through the CFP. This mechanism could cause some delay, since it enables buffering storage on some relay nodes (routers). For this reason, improving scheduling algorithms could overcome the shortcoming.

Moreover, designing a smart scheduling algorithm that can overcome the stated drawbacks is not an easy job. The synchronization process could be considered as an advantage and a disadvantage at the same time. The problem with this mechanism is the question of how to enhance network efficiency without affecting other resources. For instance, fixing the duty-cycles of a particular cluster-tree network could increase the waiting time and in turn the throughput would be affected (overhead). A static scheduling process is suitable for tiny or small projects due to the problems that could be generated by the frame collisions in large-scale applications.

The cluster-tree network is more efficient than mesh networks, since it does not require heavy routing processing. In addition, the one link in a cluster-tree effectively avoids redundant communications. This could be a problem if the single link is disconnected for any reason. Therefore, fault-tolerance and self-healing algorithms should be designed properly.

In the same context, there are some considerations for the design of these algorithms, such as the need to dynamically fashion the re-synchronization process. The dynamic management of duty-cycles should also be considered in cases of failure. In addition, the addresses and time slots should be assigned dynamically. Moreover, these features will enable node mobility.

Table 1.1 shows the differences between the three topologies of the ZigBee based on specific parameters.

Parameter	**Star**	**Cluster-Tree**	**Mesh**
$Scalability$	No	Yes	Yes
$InactivePeriod$	All devices	End Devices	All devices
$Synchronization$	Yes/No	Yes(missing)	No
GTS	Yes	Yes	No

Table 1.1: Comparison of various ZigBee topology features

1.1.3 Cluster-Tree Formation in ZigBee/IEEE 802.15.4

In a low-rate wireless personal area network (LR-WPAN) such as the ZigBee/IEEE 802.15.4 standard protocol, the network is in charge of its own organization (self-organizing). This organization is structured based on the data source destination routing procedure. In addition, by putting some restrictions on the routing procedure, the network topology can be controlled, i.e., topology formation can be fixed.

The cluster-tree formation will be constructed by maintaining a neighbor table for every node in the network. This table saves the neighbors' information on each node. During the message exchange process, each node will extract the corresponding information about its neighbors. This process will be updated within every duty-cycle. In addition, the parent-child relationship is maintained by storing the relationship information in the corresponding frames filed on each node's table. The PANC will send beacons to advertise itself. Then each FFD will join the network upon receiving an acceptance message from the PANC. Each FFD will be authorized to work as a router. Each router in turn sends beacons to advertise itself in order to allow other devices (end devices) to join its cluster, etc.

The updates on every cluster will be maintained by comparing the identity numbers for the connected nodes with the maximum counter; otherwise any new node will be rejected. The rejected nodes can join another cluster through the same process if the nodes are located in the same radio range. Every node in the network must belong to only one router. This status is maintained by storing data on the adjacent clusters in a list on every router in the cluster-tree. Hence, any overlaps will be eliminated, and each node is prevented from joining more than one cluster in the network.

In the cluster-tree-based topology, there is a unique coordinator that is called the PANC. See Figure 1.4.

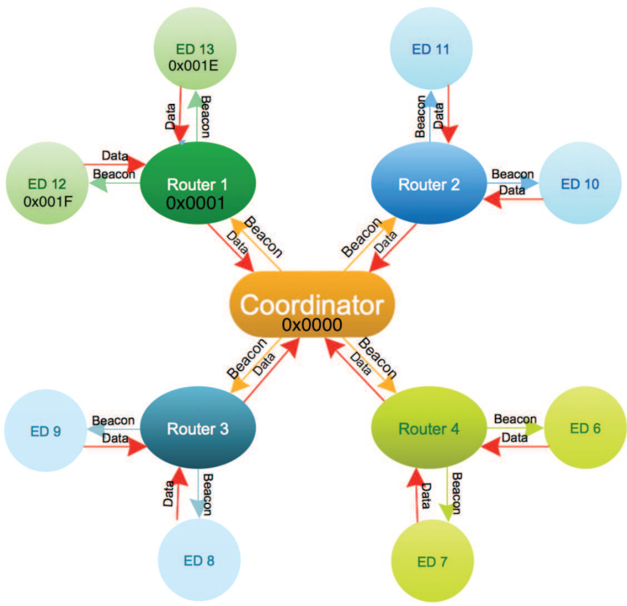


Figure 1.4: Cluster-Tree Topology

1.1.4 Addressing in ZigBee

Addressing in ZigBee is used to identify the sender and the receiver of every message in the same network.

There are two types of addresses in ZigBee, 64-bit and 16-bit. The 64-bit is defined by the manufacturer. It is a MAC address and cannot be modified.

The 16-bit address is a logical, short one and it can be assigned by the PANC to any newly joining device. This address can be changed as needed. But it must be unique for each device in the same network. The 16-bit address is used to identify every device and to allow a relevant device to send and receive data.

Table 1.2 The Ranges of Addressing in ZigBee.

Name	**Range**	**Description**
Channel	11-26	The spectrum of the Radio Frequency
PANID	0x0000-0x3fff	The address of a network within a channel
NwkAddr	0x0000-0xfff7	the network address in the same channel
Endpoint	1-240	The application address inside the relevant node
Cluster	0x0000-0xffff	The address of the object inside the application
Command	0x00-0xff	It is to take an action inside the cluster
Attribute	0x0000-0xffff	The item of data inside the cluster

Table 1.2: The Ranges of Addressing in ZigBee

Tree-based Addressing

The tree-based addressing scheme in ZigBee is used to form the network as well as to confirm the relationship between the nodes. The relationship must be a parent-child one based on the *Cskip* equation. The *Cskip* or the child skip is an address assignment function that is located in the profile of the stack number $0x01$. The name of this profile stack is ZigBee. It is used to compute the depth of the cluster-tree network, which in turn reflects the number of hops starting from the PANC.

In addition, the tree addressing scheme is used to route the messages in the network from up to down and vice versa. The scalability of the network is considered in the tree addressing scheme.

The number of maximum devices in the network of each cluster is set to 32. Every cluster is managed by one FFD, whether a PANC or a router. The reason for this limitation is the 32-bit bitmask. According to the standard, the parameter is defined as $NWK\_MAX\_DEVICE\_LIST$ and it is fixed at 32 devices at a maximum to avoid complication of the network. Moreover, the network must be configured using three important parameters: $MAX\_ROUTERS$, $MAX\_CHILDREN$, and MAX_DEPTH.

Therefore, the settings of these parameters have to follow some rules that will manage the network in a tree-based fashion. For example, the $MAX\_CHILDREN$ must have the same value as that set in the $NWK\_MAX\_DEVICE\_LIST$. This configuration is computed by the *Cskip* equation.

Logically, in each cluster, the number of end devices (children) has to be greater than their router number (parent) since in each cluster there is only one router and several end devices.

For example, suppose that we have the following $Cskip$ setting values:

1. $MAX\_CHILDREN = 4$
2. $MAX\_ROUTERS = 1$
3. $MAX\_DEPTH = 15$

Then, it becomes:

```
#define MAXCHILDREN 0x04
#define MAXROUTERS 0x01
#define MAXDEPTH 0x015


#elif ( STACK_PROFILE_ID == NETWORK_SPECIFIC )
    byte CskipRtrs[MAX_NODE_DEPTH+1]
    ={1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0};
    byte CskipChldrn[MAX_NODE_DEPTH+1]
    ={4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,0};
#endif
}
```

In this case the MAX_NODE_DEPTH will be set to 15 and the parameter $NWK\_MAX\_DEVICE\_LIST$ will be set to 4.

Suppose that we have $nwkMaxChildren(C_m)$, which reflects the number of maximum children allowable in each cluster.

And $nwkMaxDepth(L_m)$ denotes the maximum depth of the entire network. The maximum routers in the network is $nwkMaxRouters(R_m)$.

Then the cluster-tree formation can be computed using the following equation:

$$C_{\text{skip}}(d) = \begin{cases} 1 + C_m(L_m - d - 1), & \text{if } R_m = 1 \\ \frac{1+C_m-R_m-C_m R_m^{L_m-d-1}}{1-R_m}, & \text{otherwise} \end{cases} \tag{1.1}$$

In the case of $Cskip(d) = 0$, the network does not allow newly joined child nodes but they will be accepted as end devices. In other cases, such as $Cskip(d) > 0$, acceptance will occur according to the device functionality, whether it is a FFD or a RFD. The FFD will be permitted to work as a router while the RFD will work as an end device.

The $Cskip(d)$ function is also used to maintain the network addresses based on the functionality of each device. The address assignment mechanism will be managed by designing an algorithm that may set some rules for each device. The address assignment process must be scheduled and linked to time offsets. Thereafter, every device in the network may have an address and time slot.

$$A_n = A_{parent} + Cskip(d) \cdot Rm + n \tag{1.2}$$

In the case where n is

$$1 \leq n \leq (C_m - R_m) \tag{1.3}$$

and the address of the parent is A, the addresses must be unique to all devices in the network, so the PANC may have no more addresses available. Hence, any new device trying to join the network will not be accepted by the PANC. Instead, it can find another network to join or try to join again later. An example is given in Figure 1.5.

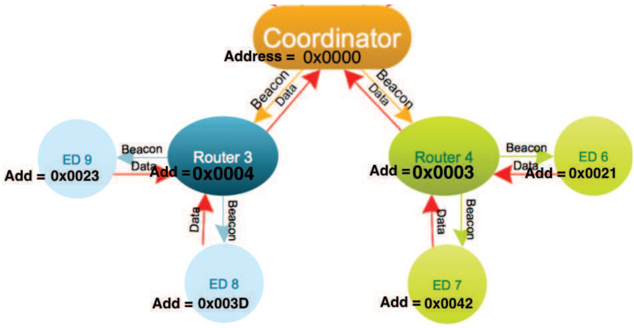


Figure 1.5: Addressing Example in Cluster Tree-based Topology

1.2 Background of Wireless Body Area Networks (WBANs)

1.2.1 On WBAN Platforms

The main goal of this section is to investigate the communication mechanism (process) of each platform in order to find out suitable solutions for organizing the on-body SDs. Thus, we have chosen healthcare filed for this investigation because it is the most important filed.

This section presents the most significant health-monitoring platforms based on WBANs. We focused on the main concept behind these platforms from a technical perspective.WSNs are also useful for monitoring people. This monitoring can include daily monitoring (of elderly or dependent persons) or patient monitoring [10]. Several works deal with monitoring people using this technology. For example, we can cite the studies of [11, 12, 13, 14, 15], which deal with implementing algorithms or software architecture for WSNs used in medical care. There are three kinds of monitoring in medical monitoring: pre-hospital monitoring, in-hospital monitoring, and in-home monitoring. In-home monitoring refers to the daily monitoring of elderly or dependent persons. In this context, the goal is to enable monitoring of an individual in a familiar place. Thus, this consideration is fundamentally a social one due to the fact that in-home monitoring allows an individual to avoid making a trip to a special medical structure. Pre-hospital monitoring, also called in-situ monitoring, consists of monitoring people at an emergency site. Such support can involve a significant number of monitored individuals. Finally, patients can be monitored at a hospital. In-hospital monitoring, as well as in-home monitoring, offer a controlled environment. For example, in these two cases, faulty sensors can be easily replaced.

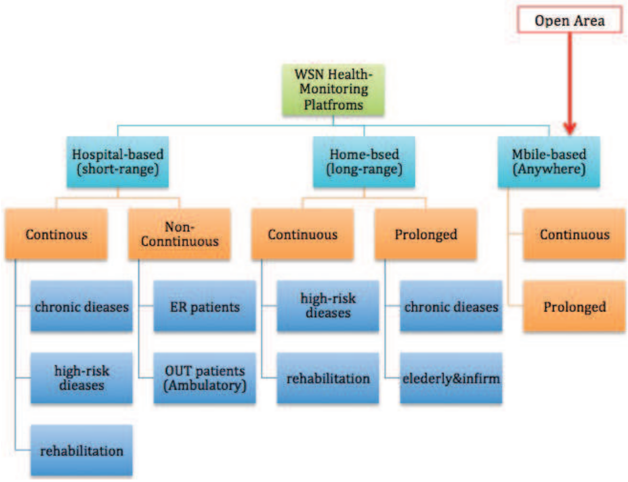


Figure 1.6: Types of monitoring system in medical platforms

Most of the actual WBAN platforms are developed using the three-tier system architecture: client, middleware, and server. Indeed, using the traditional network architecture allows the application of WBANs in the field of healthcare to be extended outside of medical institutions. Such a task could be achieved by enabling a Wi-Fi connection, a GSM system, or a VANET connection using 802.11p for example. See Figure 1.7.

Ensuring the mobility of WBANs helps encourage the use of the mobile WBAN platform, which can be used without a central database (DB) or a back-end server in a medical establishment. Another important issue to consider regarding mobility management is how to ensure and secure the channel of the communication while moving from place to place. There is also the questions of how to handle a given network changing from one communication protocol to another. To transfer medical data securely, a confidential channel of communication is needed, but unfortunately there are currently no perfect solutions that can completely secure network channels. However, there are a lot of existent solutions in this field, such as using cryptographic techniques.

An important consideration is the fact that a given patient should have the freedom to move and the ability to conduct everyday life tasks without any hindrances, including sports and activities such as shopping for instance. One additional system that should be integrated with the mobile WBAN platform is the GPS so that a patient's exact position can be obtained in cases of emergency.

In our proposal for the mobile WBAN platform (TM-WBANP), we provide a healthcare monitoring system that can detect specific indoor and outdoor diseases

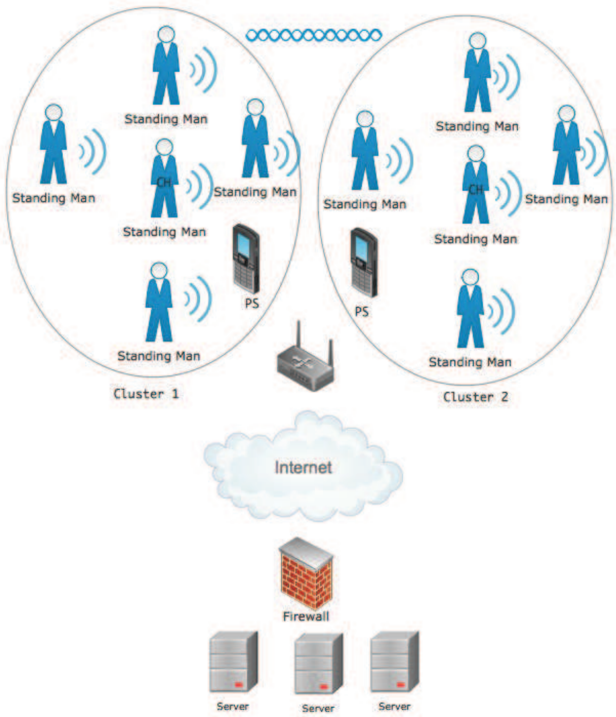


Figure 1.7: A WBAN Platfrom in Clusters

and have integrated a mobility management system (MMS) that should control the different types of wireless connections. The MMS simply works as a switch (a connection manager) in TM-WBANP. It should be noted, however, that centralization of the system is still unpractical in some cases.

1. CodeBlue: It was developed by Harvard University based on an ad hoc WSN [16]. The main objective of CodeBlue is to provide an advanced and high-quality medical platform for emergencies based on the body sensor network (BSN). CodeBlue monitors the vital signs of patients. The BSN collects various medical parameters, and then the data sink (which here is the Personal Assistant Digital (PDA) receives the data on a real-time basis and forwards it to the back-end server (database server) for storing. Then the data can be retrieved by medical professionals if needed. When an abnormal parameter is detected (a threshold has been reached), the immediate alarm system will be triggered. CodeBlue has a solid database that can prevent any real threats. The security of the DB provides encryption and authentication services. Moreover, CodeBlue has an integrated GPS to give the exact position of each patient. See Figure 1.8.

2. AMON (Advanced care and alert portable telemedical MONitoring): This project was developed by the European Union Information Society Technologies (EU-IST) [17]. AMON is a continuous medical monitoring platform. The

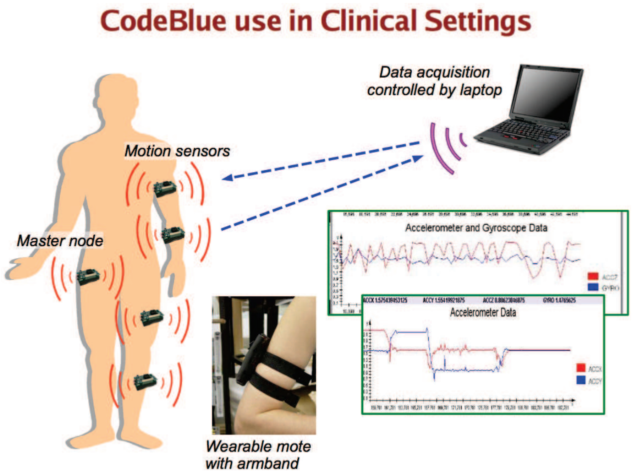


Figure 1.8: CodeBlue use in clinical settings

main goal for the platform's development was to provide an Òadvanced care alert portable telemedical monitoring platformÓ that would serve high-risk patients, e.g., those with a cardiac disease. Patients can wear the system as a wristwatch "wrist-worn". It has the ability to measure multiple vital signs simultaneously in near real time, with a secure channel between the AMON device and the telemedicine center (TMC). The TMC works as a filtering and monitoring service that manages the active patient. AMON is an extendable platform; it can be integrated easily with other SDs. A patient's activity level can be detected and correlated with the patient's vital signs. Moreover, this platform has an online service that analyzes the recorded measurements; each patient can see his results online. Additionally, the patient can receive any results on his smartphone. During the analysis of the collected medical data, the system checks whether any abnormal parameters have been detected. If any are seen, then the alarm system will be triggered automatically. This platform can open a direct line of communication between the patient and medical personnel.

3. Personal Care Connect (PCC): It was developed by IBM [18]. PCC is a home-based platform; it provides a health-monitoring platform for patients to use in their homes. The system collects medical data from the patients in their homes and then sends it to a remote server via a PCC hub that can reliably handle the data even if communication is lost between the hub and the server. In such a scenario, the data can be handled in the SDs for later use. The server processes the obtained data and then the patients can view their results remotely via a graphical user interface (GUI). With regard to security, PCC has been tested using secure HTTPS communication between the hub and the server. PCC can integrate heterogeneous SDs in an efficient way through the use of

an application programming interface (API). The platform is based on Java, which provides ease of use in terms of communication. The interface is used to communicate with other medical devices that support Bluetooth standard communication. For instance, data can be easily aggregated through the use of a Java-based personal digital assistant (PDA) that has Bluetooth.

4. AlarmNet: It was developed by the University of Virginia [13]. It is a residential platform, meaning that it is a home-based platform. AlarmNet is a health-monitoring platform that allows the continuous monitoring of patients in their homes. It uses physiological and environmental sensors. A circadian activity rhythm (CAR) program has been implemented in this platform, and it is used to analyze rhythmic behaviors and daily activities. Results from the CAR can be used to indicate whether there is a particular health problem. The system provides a strong backdrop in terms of security and privacy through its reliance on AlarmGate, which uses sophisticated algorithms. AlarmNet provides the ability to manage power consumption; for example, the automatic sensors turn off and on every two minutes. The GUI between computers and AlarmNet is yet another distinguishing feature. In terms of reliability, the motion sensor can faithfully detect patient movement. See Figure 1.9.

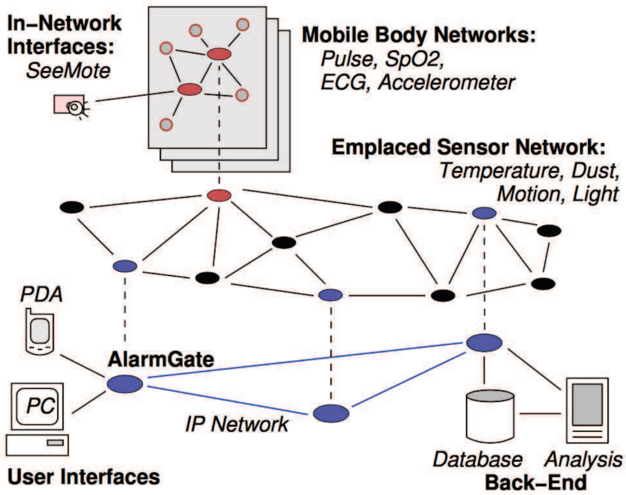


Figure 1.9: ALARM-NET Platform

5. Medical MoteCare: It was developed by the Sydney University of Technology [19]. It is a monitoring platform for the elderly and infirm. MicaZ (Crossbow) is the basis of this platform. Medical MoteCare provides five modules: the Motes (sensor devices), the StarGate (personal server that supports Wi-Fi, GSM, and GPRS), the data acquisition board, the monitor unit (MU), and the network management server (NMS), which is based on CodeBlue software. It works as a logfile. The MU and the StarGate communicate with each other

in order to capture the medical data to be sent to the logfile. The MU delivers data by using a TCP/IP communication protocol. The NMS works as a data analysis system, so the medical data that has been obtained will be analyzed and checked to see whether any abnormal parameters can be detected. In terms of security, Medical MoteCare uses a simple network management protocol (SNMP); it is a management protocol for the Internet that can control devices over the Internet protocol (IP) networks. This platform uses a powerful browser, known as the iReasoning MIB browser, which helps medical professionals manage the attached devices. Medical personnel can view any medical data graphically via the MIB browser in real time.

6. MEDiSN: It was developed by Jeonggil Ko, Tia Gao, Richard Rothman, and Andreas Terzis [20]. It is a heath-monitoring platform that monitors the vital signs of an ambulatory patient. A pilot study was conducted in an emergency room (ER) department to ensure that MEDiSN is applicable in the field. This platform uses relay points (RPs) that work as a wireless backbone to extend network coverage in the ER department as well as to enhance wireless network performance. MEDiSN has five modules: miSense is the end-to-end platform for healthcare-sensing applications, miTag or the patient monitors (PMs) is a WSD for healthcare applications, miNet is a wireless network that transmits miTag measurements for remote configuration by the administrators, and miStore and miView are servers. They store the collected data, which can then be retrieved by authorized users. In terms of security, miTag provides secure communication by encrypting and signing the collected data that is offered by the CC2420 radio. MEDiSN uses a finger clip to measure vital signs. The data collected by the PMs are sent through the RPs to the back-end server remotely via the Internet. The RPs use the collection-tree protocol to facilitate data delivery to any destination over the Internet. The developers distributed a satisfaction survey to gather feedback on the system's usability and they have received a favorable response.

7. MEDIC (Medical Embedded Device for Individualized Care): It was developed by the University of California [21]. It is a health-monitoring platform. MEDIC provides flexibility in terms of system configuration; the administrator or medical staff can do many tasks remotely. It uses a number of physiological sensors, which can be worn on a patient's body. The wearable computer (PDA) has a direct line of communication with the SDs using Bluetooth. The system can extend and integrate other environmental sensors. The wearable computer processes the obtained data and checks for any abnormal parameters. Patients can easily see their results via the GUI on the wearable computer. The remote back-end server receives and stores any collected data. The server allows medical staff/the administrator of the system to remotely conduct data processing. The developers are planning to do a live test of their platform in a rehabilita-

tion department. The system provides real-time results using its physiological sensors while PDAs are used for real-time commination, e.g., by using the built-in Bluetooth feature that is commercially available in PDAs. MEDIC uses the secure shell (SSH) protocol, which establishes a secure channel of the communication between various networking devices.

8. MEMESWEAR: It was developed by the National University of Singapore [22]. MEMESWEAR was inspired by microelectromechanical systems (MEMS technology). The main goal of this platform is to monitor the vital signs of patients and important events, such as an elderly person falling. MEMESWEAR is a wearable monitoring system that uses a T-shirt developed based on the ZigBee/IEEE 802.15.4 protocol. The wearable system collects medical data and sends it wirelessly using the PDA's Bluetooth. Once medical personnel have received the information, it is forwarded and stored in a back-end server over the GSM communication protocol. It is possible to allow family members to access the data as well so that they will be more aware of a patient's situation. MEMESWEAR uses the body area network with two modes: active and passive. The environmental sensors continuously collect medical data without forwarding it until something of note occurs, i.e., no action is taken until an abnormal event is detected. This platform can be used in a home or hospital.

9. iCare: It was developed by the Dalian University of Technology, China [23]. It is a mobile health-monitoring platform. iCare uses WBANs and cell phones to monitor the elderly. It includes an emergency agent to help the elderly in cases of emergency by sending an automatic SMS over the GSM mobile network standard. Family members can be included and granted permission in iCare to be a medical contact in emergencies. The personal health information system exists as the server in this platform. The server works as a medical guide, which means that it can provide real-time healthcare guidance to medical users. iCare can detect the exact position of a patient in order to notify an ambulance using SMS/calling over GSM. This system provides monitoring services for the elderly in real time. It serves as a living assistant to patients while they are in their homes or even outside of them.

10. DexterNet: It was developed by the University of California, USA; Cornell University, USA; Telecom, Italy; the Tampere University of Technology, Finland; and the University of Texas at Dallas, USA [24]. It is an open-source platform for health monitoring. It provides real-time communication. DexterNet provides three-layer architecture: a body sensor (BS) layer (BSL), a personal network layer (PNL), and global network layer (GNL). The first two layer functions are implemented as an Òopen-source software library, which is called the signal processing in node environment (SPINE). Based on SPINE,

the platform can be reconfigured Òon the flyÓ. Any medical data obtained by a SD will be stored on a remote server using the TCP/IP standard. DexterNet supports the communication of multiple PNLs simultaneously with the remote server. The platform is extendable; it can integrate with other kinds of motes. The elderly can benefit from using DexterNet in their homes. Physicians have the ability to monitor their patients remotely by recording and visualizing their movements via a GUI. The algorithm for DexterNet maximizes its flexibility, extensibility, and reusability as an open-source system. See Figure 1.10.

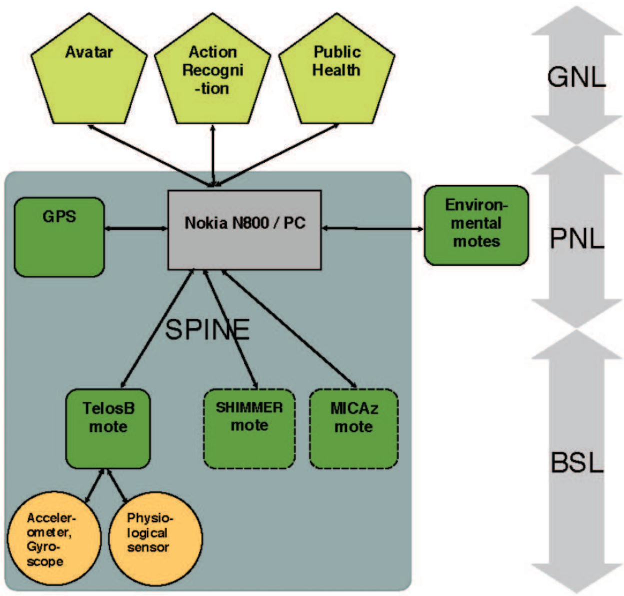


Figure 1.10: The architecture of DexterNet

11. Health Gear: It was developed by Microsoft [25]. It is a wearable heath-monitoring platform. It monitors vital signs and analyzes the physiological signals of patients. A smartphone is used to connect wirelessly with the physiological sensors through Bluetooth. The cell phone presents the collected data and performs data processing and analysis. Health Gear is mainly focused on monitoring the vital signs of people who are sleeping. An algorithm is used for the detection of sleep apnea events. Health Gear was developed as a mobile application, and it has all the possible functionalities such as data reception, data analysis, data display, and data storage. It shows the obtained data in real time using smartphones. This platform detects abnormal measurements and the exact position of a patient by employing GPRS technology. After conducting a usability survey, Microsoft has gathered a favorable response. Patients were satisfied with the system's functionality, but more work must be done to enhance the level of comfort experienced when it is worn.

12. MobiHealth: It was developed by the University of Geneva, Switzerland, and the University of Twente, the Netherlands [26]. It is a mobile health-monitoring platform. It monitors vital signs remotely using a highly customized tele-monitoring and tele-treatment system. MobiHealth was created based on a body area network and mobile healthcare (m-health). It allows a variety of medical sensors to be connected wirelessly and provides real-time transmission of the measured parameters, thus giving patients feedback in real time as well. MobiHealth provides two kinds of communication: intra-BAN is for short-range communication between entities using Bluetooth, while extra-BAN is for long-range communication between a mobile base unit (MBU) and a back-end system (BESys) using (3G-UMTS). This platform is dedicated to monitoring patients with high-risk chronic diseases in their homes. The developers are working on security and privacy considerations, which will soon be implemented. Measures must be put in place for the authorization and authentication of medical users as well.

13. MobiCare: It was developed by the University of Cambridge, UK, and the University of Wisconsin, Madison, USA [27]. It is a mobile health-monitoring platform. The system architecture is divided into three parts: the BSN, the BSN manager and the back-end infrastructure. It provides a wide range of functionalities, such as the ease of reconfiguration and programming for services. MobiCare features some important health-related services, including devices and sensors that let the system remotely install, reconfigure, self-activate, or self-repair any other services or applications. The medical devices provide security and reliability with remote dynamic upgrades and even update services. Registration and reconfiguration of the BSs and health data services can also be done remotely. MobiCare uses Bluetooth for short-range communication and GPRS/UMTS for long-range communication.

14. Sensium[a] SoC (System-on-Chip): It was developed by Toumaz Technology Ltd., Oxfordshire, UK [28]. It is a SoC health-monitoring platform. Sensium provides an ultra lower-power system using SoC, which enhances quality in the healthcare sector and reduces overall cost. The primary objective behind the development of Sensium was to monitor patient vital signs, but SoC can be interfaced to any physiological sensor to achieve continuous monitoring. By factoring in SoC as a key factor, real-time vital sign monitoring has been achieved. This platform is used as a tiny component in the wireless body sensor network (WBSN) platform; it is called the LifePebble.

15. Body Sensor Network Development for Pervasive Healthcare (BSN-DPH): It was developed by IBHE, Key Laboratory for BIHE, SIAT, and CAS, Shenzhen, China [29]. It is a highly integrated and low-power platform. The platform is

divided into four parts: the node board works as a BSN, the base station, the battery board, and a number of a prototyping boards to debug the system if needed. The BSN is characterized by its ease of use. The developers only created two BSNs for the initial run of this platform. PPG is being used to measure the pulse rating and oxygen saturation. They have designed their own BSN software architecture by defining a self-contained BSN protocol that includes a PHY layer, a MAC layer, and an LLC layer. Reliability is achieved using three BSN nodes placed on different parts of a patient's body that use a five-second time step. The BSN-DPH was specifically designed to be small and stackable.

16. LOBIN: E-Textile and Wireless-Sensor-Network-Based Platform for Healthcare Monitoring in Future Hospital Environments. It was developed by Carlos III University, Madrid, Spain [30]. LOBIN was developed under the LOBIN project. It is a remote health-monitoring platform that can monitor physiological parameters and has the ability to track a given patient's position. It is a hospital-based system. LOBIN uses a wearable T-shirt created from e-textile that has some sensors attached to measure physiological parameters in real time; the collected data is then transmitted via the hospital's WSN and stored in the management server. A number of tests have been conducted to make sure the platform is capable of serving the maximum number of patients. Such tests reflect the QoS of the system. LOBIN has been validated through its use in the Cardio Department at La Paz Hospital, Madrid, Spain.

17. Efficient Patient Monitoring for Multiple Patients using Wireless Sensor Network (EPMMP): It was developed by Amrita Vishwa Vidyapeetham, Amritapuri [31]. It is a health-monitoring platform for multiple patients based on a WSN. Each patient in the hospital has a unique ID number for easy identification by medical personnel. The system was developed to monitor multiple patients at the same time, because it is difficult for healthcare providers to accomplish such a task with a non-automated infrastructure. This system is capable of monitoring multiple parameters simultaneously and makes it possible to compare results against threshold limits. The collected data is then transmitted remotely to medical personnel. The system is able to send the medical data to a doctor's mobile phone, and it can be accessed by authorized people via the Internet. EPMMP makes it possible to continuously monitor and collect medical data depending on the risk factor of the patient in question, e.g., a high-risk patient should be monitored continuously whereas a low-risk patient can be monitored intermittently. This system can reliably serve many patients at the same time and makes indoor monitoring easy.

18. Home Healthcare Platform based on a WSN (HCP-WSN): It was developed

by Zhongkai University and South China University [32]. It is a remote health-monitoring platform based on a wireless biomedical sensor network. The developers used ZigBee/IEEE 802.15.4, which yields low power usage at low cost. The HCP-WSN has the flexibility to be used as a home-based platform. It can be remotely accessed using the Internet via a PDA. It is lightweight and portable, using smaller versions of SDs. This platform monitors vital signs through a number of biomedical sensors; environmental sensors and wearable SDs are used to track a patient's daily activities. The HCP-WSN supports integration with other SDs, i.e., it can be integrated with new SDs at low cost. This platform continuously monitors a given patient in real time using both environmental and physiological sensors. The HCP-WSN provides flexibility with regard to installing and adding or removing sensors, which means that it is self-organizing and reconfigurable.

19. Home-based Healthcare Monitoring over a Wireless Biomedical Sensor Network (HbHM-WBSN): It was developed by the University of Technology Malaysia [33]. It is a home-based health-monitoring platform. It provides continuous monitoring for patients in their homes or while they are outside, e.g., in a home garden. The collected data is then sent to local personal computers (PCs). After that it is forwarded via the Internet to the Hospital Health Monitoring Center (HHMC). Medical professionals can monitor their patients remotely using the HHMC; the data can also be seen by doctors who may need to take immediate action in the case of an emergency. The HbHM-WBSN provides a real-time monitoring system.

20. Patient Vital Signs Monitoring Using a Wireless Body Area Network (PVSM-WBAN): It was developed by Rutgers University, Piscataway, New Jersey [34]. It specializes in monitoring a patient's vital signs in pre-hospital or hospital environments. The developers have proposed a new interference-aware WBAN system that can monitor vital signs continuously. This platform supports the monitoring of multiple patients. The QoS is applied in terms of medical data transmission. The developers sought to maximize the reliability of a transmission carrying data on the vital signs of patients. They propose a two-tier networking architecture: inter-BAN (high level) and intra-BAN (low level); the two tiers enhance overall performance and the quality of communications between the sensors in different parts of the patient's body, e.g., by choosing the best and most suitable routing path based on route quality, which also enhances the quality of low radio frequency (RF) interference as well.

1.2.2 Classification and Evaluation of WBAN Platforms

Classification of WBAN After conducting our survey, we noticed that most of the previously proposed solutions are quite similar. They have followed the typical three-tier network architecture (client, middleware, and server). Overall, they focused on location-based diseases, the monitoring of vital signs, mobility (ease of use, extendibility, real-time operation), Bluetooth (short-range), GPS/GSM/3G and TCP/IP (long-range), and how to deliver medical care based on QoS (reliability, integrity, and security). QoS is a challenging area of study and it is not easy to achieve a satisfactory level of QoS in medical care through the use of WBANs, as we mentioned above. The mobility to use a WBAN anywhere still needs more research, and we came to this conclusion in our previous studies. However, we have designed a hierarchical classification of some platforms based on the aforementioned factors.

CodeBlue has been built to deliver high-quality medical care to the patients experiencing an emergency while as iCare as well as DexterNet are dedicated to the elderly who are residing in their own homes. AMON focuses on providing continuous medical care to high-risk patients. IBM PCC and AlarmNet concentrate their solutions on home-based patients. Medical MoteCare has dedicated its solution to the elderly and infirm, and the developers of this platform have taken advantage of CodeBlue software in terms of how they built their own server. MEDiSN is used in ER departments to allow ambulatory patients to monitor their vital signs. MEDIC has not yet been tested in a real-life scenario, but the developers intend to use it in a rehabilitation department. MEMESWEAR has been developed to monitor vital signs and detect when elderly people fall. Health Gear is exceptional platform that focuses on monitoring vital signs while patients are sleeping, regardless of their location. MobiHealth and MobiCare generally have the same idea, which is premised on the platform's mobility and continuous medical care. The BSN-DPH developers did not mention the location for their platform's use; instead, they concentrated on designing a miniature and stackable system. LOBIN is a wearable T-shirt that uses an e-textile system intended for a hospital environment. EPMMP was designed to monitor multiple patients in a hospital. HCP-WSN and HbHM-WBSN are home healthcare platforms for the continuous monitoring of patients. PVSM-WBAN was developed to continuously measure a patient's vital signs in pre-hospital and hospital environments. Finally, there is no existent solution for a WBAN that can be used anywhere, e.g., where the patient can carry a BSD around while shopping or walking to a train station.

Evaluation of WBAN Based on the provided WBAN platforms, this section presents some important measurements that can be used to evaluate these platforms. Reliability, availability, usability, QoS, and security are the most important issues in designing and evaluating the WBAN platforms.

Reliability: Reliability is one of the most important requirements in WBAN platforms. WSNs designers must be careful in choosing the right network topology and routing protocols in order to ensure the reliability of the WBAN platforms. However, the tools and applications involved in designing and analyzing a WBAN are very important because of the sensitivity of the medical data and privacy requirements in the medical care field. For instance, it is important to use sophisticated routing algorithms that can improve the reliability of WBAN platforms.

Availability: Medical data must be available when it is needed. For instance, the failure to retrieve the medical data of a particular patient in a timely manner might affect his life. Hence, WBAN platforms must support information availability. The system has to cope with instances of failure and malfunctions efficiently. Communication between sensors through gateways, sending or receiving medical data, and storing it in or retrieving it from database servers must always be possible. Adapting and implementing a SHR protocol and self-reconfiguration SDs can improve the availability of WBAN platforms.

Usability: Some of the WBAN platforms are wearable or implantable. These systems should be easy to use and comfortable. Usability in WBAN platforms is a critical factor since dealing with patients calls for easy-to-use systems. Thus, in order to get accurate results, patients should have positive interactions with these systems. For instance, conducting a survey to collect patient feedback or gauge satisfaction in terms of system usability is an important metric.

Quality of Service: The QoS in WBAN platforms is also an important issue in wireless networking in general. Reliability, availability, and security are considered to be categories underneath QoS, i.e., without QoS there would be no guarantee of high-quality service being provided. By enabling QoS in WBANs, we can provide excellent medical services. But QoS in a WSN in general is still not enough because of the redundancy in distributing the sensors, the routing protocols, and MAC along with the short range of the RF. A WSN is not an end-to-end communication module, which means many consideration must be taken into account in this area in order to achieve high-level QoS with WSNs. In WBAN platforms and when applying QoS during design stages, we have to consider the nature of the medical environment and the CAR, which refers to the daily activity of the human body. Using environmental and physiological sensors on the human body to obtain medical data or to detect specific events requires the QoS to guarantee data delivery.

Security: Security is a vital issue in WSNs in general and using advanced algorithms might help prevent any real threats or attacks between the SDs. Security is divided into five parts: (1) integrity, (2) availability, (3) confidentiality, (4) authentication, and (5) authorization [3]. Ensuring that these considerations are included is important for a WSN. Using a secure routing algorithm based on identity-based cryptography (IBC) is a good solution. As we stated before, the sensitivity of the medical data and privacy in the implementation of WBAN platforms are quite important issues. Medical quality standards require that the security of a patient's medical records be ensured.

Based on the provided solutions, we have evaluated all the studied platforms as shown in table 1.3.

Table 1.3: WBAN Platforms Evaluation

No	WBAN	Reliability	Availability	Security	Usability	QoS
1	CodeBlue	yes	no	yes	no	yes
2	AMON	yes	no	yes	no	no
3	PCC	yes	no	yes	yes	no
4	AlarmNet	yes	no	yes	no	no
5	MoteCare	yes	no	yes	yes	no
6	MEDiSN	no	no	yes	no	no
7	MEDIC	yes	no	yes	no	no
8	M.SWEAR	no	no	no	no	no
9	iCar	yes	no	no	no	no
10	DexterNet	yes	no	no	yes	no
11	HealthGear	yes	no	no	yes	no
12	MboiHealth	yes	no	no	no	no
13	MobiCare	yes	no	yes	no	yes
14	Sensium	yes	no	no	yes	yes
15	BSN-DPH	yes	no	no	yes	no
16	LOBIN	yes	no	no	no	yes
17	EPMMP	yes	no	yes	yes	no
18	HCP-WSN	yes	no	no	yes	no
19	HbHM	yes	no	no	no	no
20	PVSM	yes	no	no	no	yes

1.2.3 On Medium Access Control Protocols for WBANs

The WBANs require flexible communication process that can manage duty-cycles efficiently. In some cases, the WBANs should support high priority events to report critical situation for a particular patient. For example, detecting a heart failure for a particular patient require a time sensitive application which supports priority events in real-time. The communication of on-BSDs must be scheduled to avoid packet collisions. The communication management (scheduling process) has to control idle listening and overhearing. Furthermore, minimizing idle listening and overhearing will improve the communication quality. In addition, packet overhead will be eliminated by minimizing waiting time for a not ready SD to send. For example, some devices do not have data to send, in this case they should not be synchronized frequently if exist.

There are some MAC protocols, which are designed for WSNs in general, but they could be also used for the WBANs. The IEEE 802.15.4 has been considered as one of most suitable protocol for the WBANs for many reasons. It supports low data rate applications such WBANs. In addition, it supports low-power communications. Moreover, the beacon-enabled mode is more appropriate than the non-beacon for these applications. The time critical applications need an always synchronization process which is in this case supported by beacon frames. The exchanging packets process between on-body devices will be controlled by using the control packets such as beacons and acknowledgment packets.

According to ZigBee/IEEE 802.15.4 standards, there are two main approaches. The first one is the time division approach. In this approach, each device can only communicate within its time slot. The second approach is the beacon only period. In this approach, the PANC and all of its router children must send their beacons within the first period of the SuperFrame. Then after that all other devices can communicate.

There are some MAC protocols that are designed specifically for WBAN applications such as A Heartbeat Driven MAC protocol (H-MAC). It is based on the time division multiple access protocol (TDMA). The authors developed this protocol for a start topology. The heartbeat rhythm information is used to control the energy consumption. Thus, the synchronization process is maintained by using this mechanism. The time slot assignment process for each device is maintained by the main coordinator to avoid packet collisions.

In other proposals such as A Reservation-based Dynamic TDMA Protocol (DTDMA) the authors focused on a periodical traffic. This protocol is designed based on the TDMA. Every node in the network will be assigned by a time slot based on the buffered information on the coordinator. In addition, this protocol uses beacons to control the time slot assignment process by exchanging the control packets within the network. In this protocol the CFP comes before the CAP to enable the guaranteed devices to send their data first.

Based on these studies, there are some drawbacks in terms of guaranteeing a collision-free environment in beacon-enabled mode networking. Specifically, the scheduling process that could classify sensory data being generated from different SDs must be considered.

1.3 Conclusion

The study that is presented in this chapter aims at investigating the related works in technical perspective way.The WSNs is a very large domain to cover, because of a lot of protocols and standards that can be used. In addition, the WSNs have also subdomains such as WBANs which is designed specifically for healthcare applications. The interoperability between the network SDs that are interconnected wirelessly is demanded for some purposes, such as monitoring patient remotely, agriculture monitoring, forest fire monitoring (Fire Detection), etc. The interaction between these SDs would also help in detecting some debases by perfuming a special computation algorithm. Furthermore, to understand this interoperability a deep study on the standards and the protocols is required. Logically, the sampling data that are obtained from these devices will indicate certain issues. These data will be transformed into some information to provide the intended services. The communication between these devices need to be scheduled. Each device transmits data, the transmission must be organized to avoid collisions. The transmission process is maintained in duty-cycles. The management of every duty-cycle has to be well managed. Thus, the network performance to the end-to-end communication will be improved. This will increase the throughput and eliminate overheads. In addition, waiting time will be minimized in every duty-cycle.

Next chapter, a mobile WBAN platform is designed by detailing its architecture as well as its requirements and considerations. Further investigation is done to check out communication drawbacks in the WBANs by proposing an epidemic tracking process in urban areas. This study has shown the shortcoming of the communication level, which encourages us to propose three different task scheduling techniques.

1.4 Résumé du Chapitre 1

Ce chapitre a pour objectif de présenter les travaux connexes réalisés tout d'abord sur les réseaux de capteurs sans fil (Wireless Sensor Networks WSNs) en général et par la suite sur les réseaux Wireless Body Area Networks (WBANs) plus spécifiquement. Il permet ainsi de présenter les fondements nécessaires à la compréhension du reste de manuscrit de thèse.

La première partie de ce chapitre est consacrée à la présentation des réseaux de capteurs sans fil. En effet, il comporte une présentation détaillée des protocoles IEEE 802.15.4 et sur la couche du Standard Zigbee. De plus, les couches réseau, MAC et physique sont détaillées pour ces types de réseaux. Par la suite, les différentes topologies possibles dans Zigbee sont décrites. Ces dernières sont les topologies : étoile (Star), arbre (Cluster-Tree) et maillée (Mesh). Par la suite, l'accent est mis sur la formation de ces topologie avec l'explication de l'assignation des adresse du PANC aux routeurs et aux noeuds fils surtout dans la topologie étoile qui nous intéresse plus particulièrement dans ces travaux de thèse.

La deuxième partie se concentre sur les fondements des réseaux WBANs. Il s'agit tout d'abord de présenter un large panel de plateformes WABNs déjà existantes : CodeBlue [16], AMON [17], Personal Care Connect (PCC) [18], AlarmNet [13], Medical MoteCare [19], MEDiSN [20], MEDIC [21], MEMESWEAR [22], iCare [23], DexterNet [24], Health Gear [25], MobiHealth [26], MobiCare [27], Sensium [28], BSN-DPH [29], LOBIN [30], EPMMP [31], HCP-WSN [32], HbHM-WBSN [33] and PVSM-WBAN [34]. Ces plateformes ont été classées et évaluées selon des critères importants : la fiabilité, la disponibilité, la facilité d'utilisation, la qualité de service et la sécurité.

Le résultat de cette comparaison est présenté dans la table 1.4.

Enfin, ce chapitre résume les principales caractéristiques spécifiques des protocoles MAC qui sont utilisés dans les réseaux WBAN.

Table 1.4: Evaluation des Plateformes WBAN

No	WBAN	Fiabilité	Disponibilité	Securité	Facilité d'utilisation	QoS
1	CodeBlue	oui	non	oui	non	oui
2	AMON	oui	non	oui	non	non
3	PCC	oui	non	oui	oui	non
4	AlarmNet	oui	non	oui	non	non
5	MoteCare	oui	non	oui	oui	non
6	MEDiSN	non	non	oui	non	non
7	MEDIC	oui	non	oui	non	non
8	M.SWEAR	non	non	non	non	non
9	iCar	oui	non	non	non	non
10	DexterNet	oui	non	non	oui	non
11	HealthGear	oui	non	non	oui	non
12	MboiHealth	oui	non	non	non	non
13	MobiCare	oui	non	oui	non	oui
14	Sensium	oui	non	non	oui	oui
15	BSN-DPH	oui	non	non	oui	non
16	LOBIN	oui	non	non	non	oui
17	EPMMP	oui	non	oui	oui	non
18	HCP-WSN	oui	non	non	oui	non
19	HbHM	oui	non	non	non	non
20	PVSM	oui	non	non	non	oui

Chapter 2

A Mobile WBAN Platform

Contents

This chapter aims at investigating deeply the communication level of a WBAN Platform. It describes a mobile WBAN Platform in details by presenting the requirements and the considerations of this model. Thereafter, an example is given to clarify the mechanism of this model. In addition, an extended process is proposed to include epidemic tracking in urban areas. It is a new strategy for supervised learning in the detecting and describing of instances from different stages of disease in the case of epidemics. This study reveals some drawbacks in terms of task scheduling process. The on-body devices need to be scheduled efficiently to avoid packet collisions.

2.1 Platform Architecture of WBAN

General Aspects

A medical protocol is usually described as follows: a disease is diagnosed if a set of symptoms is verified. These symptoms can be verified thanks to blood analysis results or by observing patient behavior (vomiting or headaches, for example). In other cases, these symptoms rely on the fact that some health parameters (heart rate, electrocardiogram, temperature, or blood sugar) have reached abnormal values. Let us look at a case of gastroenteritis as an example. A patient has gastroenteritis if the following symptoms are verified: a) diarrhea, b) fever, c) headaches, d) vomiting, e) loss of appetite, and f) abdominal cramps. In this case, only one symptom (fever) can be sensed by a transducer. We focus on such symptoms and intend to suggest various diseases that are associated with them. Even if the example we presented is quite simple, our model is designed to handle more complex diseases. The health parameter values are usually sensed by transducers that are embedded on the patient and linked to a specialized engine. The first works that have been studied the use of a WSN in medical care relied on a centralized approach. In this scenario, a patient is equipped with a set of motes (a sensor linked to transducers) and each mote measures some health parameters. All the collected values are sent to a base station that performs all computations, analysis, and detections. This centralized paradigm leads to some drawbacks. First, routing mechanisms have to be implemented in order to ensure that all the information collected by the sensors is transmitted to the BS. Second, BS failure implies that no detection is possible anymore. In a fine-grained multi-hop network, which is highly prone to congestion, it remains difficult to ensure the real-time detection needed in an intensive care unit. In the context of in-hospital monitoring, we suppose that each WBAN transmits the data to a wired network that is then responsible for forwarding it to the base station.

As shown in Figure 2.1, communication between all body sensors (BSs) will run in order to collect information on the number of people suffering from a given disease and the number of people who could have the same disease in the near future. The aim is that each body sensor network (BSN) that diagnoses a disease or an "almost disease" will send the data to the neighborhood. Each BSN will then aggregate its data with what has been received. Step by step, the data will move until a base station is found in the neighborhood. At that time, the collected data is recorded.

Next, we describe the general architecture of TM-WBAN. The system architecture of the WBAN platform usually consists of three tiers, as follows:

- 1st tier is the client (Sensor Device): A wearable or implantable SD placed on the patient's body.
- 2nd Tier is the middleware (Personal Server): The personal server, it is computer software that could be installed on a PC or a smartphone.
- 3rd Tier is the back-end server (Database Server): The database server is located in the medical institution.

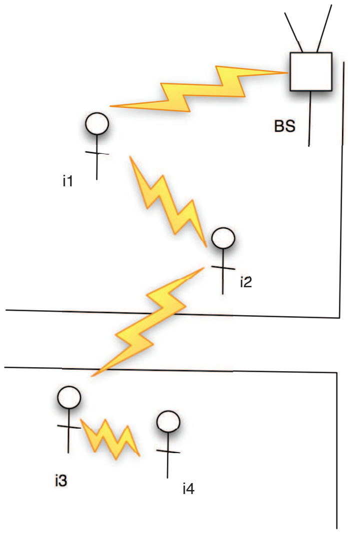


Figure 2.1: Communication between body WSNs

Moreover, in our proposal, we integrate a MMS to enable the mobility of TM-WBAN so that it can be used indoors and outdoors. There are a number of challenges in the WBAN that should be taken into account:

- It is small scale and will only work in the human body.
- There are very few sensors and they depend on the status of a given patient.
- Each sensor has a node with its own task.
- It has a reduced network size that only includes the human body.
- Sensory data must be protected, further emphasizing the importance of data protection.

The system architecture of the WBAN platform can operate with only two tiers (the client and the personal server). This decision depends on the goal of the WBAN platform being developed. Sometimes the central database is unnecessary. We will provide an example in the last section of this paper where a WBAN is used without a central database.

In the rest of this section, we will discuss each tier in greater detail to illustrate the use of a WBAN platform, as shown in our flowchart in Figure 2.2. As we stated before, the system architecture of the WBAN platform generally consists of three tiers (the client, middleware, and the server).

The First Tier

Hardware requirements: The WBAN node, which is a wearable or implantable SD placed on different parts of a patient's body is called a mote. The SDs must be

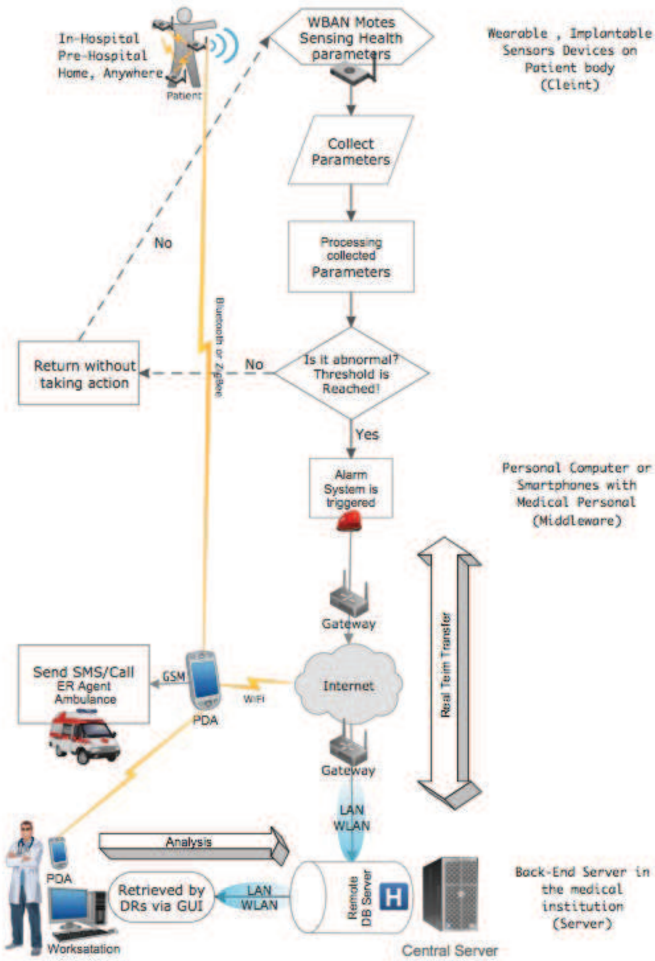


Figure 2.2: The Wireless Body Area Network Flow Chart

lightweight, easy to use, and comfortable in order to satisfy patient needs and enhance the overall performance and accuracy of medical data collection. Using motes that are commercially available and compliant with IEEE 802.15.4 is common practice when developing a WBAN platform such as Shimmer, Mica2Dot by Crossbow, Telos Mote by Moteiv, etc. For example,

> *"Shimmer is a small wireless sensor platform that can record and transmit physiological and kinematic data in real-time".* [35]

However, motes should provide high performance with regard to reliability and multi-hop networking.
There are two main types of sensors: physiological and environmental sensors. Physiological sensors collect data on vital signs such as blood pressure, heart rate, and oxygen saturation. Environmental sensors gather information on temperature, light, and location. Using both physiological and environmental sensors could be very useful in the ubiquitous monitoring of a high-risk patient and the elderly.

Software Requirements Software requirements: Each SD has a number of requirements, as seen below.

- The SD should be implemented in a TinyOS environment.
- A proper communication protocol must be chosen.
- Power must be efficiently managed.
- Collected data (events) must be efficiently managed as well.

The SDs must collect medical parameters from the sensors that are located on different parts of a patient's body. The data should then be correlated using the time synchronization feature, thus facilitating on-sensor data processing and communications through the transmission of the measured data to a personal server. After the data processing or event processing step, abnormal events will be marked by the detection algorithms. These detected event occur in precise time slots, so event messages will be created and linked to an accurate time stamp. Then the information will be ready for real-time transmission. Transmitting the event messages into packets requires a unique sequence number in each frame for the retransmission of the data to its destination. ACK messages can also be used to ensure the data has been delivered to its destination. However, the buffering of an event message's on-sensor memory is dependent on a specific SD's capabilities.
The capability of each sensor depends on its type. Other factors to consider are the lightweight nature of a WSN and its limited resources, such as its limited memory, low power level, and the short range of WSN radios.
Most SD hardware architecture consists of an A/D convertor, a microcontroller, a transceiver, and memory. For instance, the Mica2Dot specifications are as follows:

uProc: 4Mhz, 8 bit Atmel RISC; radio: 40 kbit 900/450/300 MHz or 250 kbit 2.5GHz (MicaZ 802.15.4); memory: 4 K RAM / 128 K Program Flash / 512 K Data Flash; and power: 2 x AA or coin cell [36].

Based on the WBAN data flow, there are a number of devices that must be properly programmed in order to meet medical standards such as HIPPA. Moreover, there are a number of considerations and standards that must be implemented in the development of such a platform, as follows:

- Availability
- Usability
- Security and privacy
- Quality of Service

In terms of software requirements, there are five important requirements in our proposal, described as follows:

- To process the collected medical data on a sensor in real time.
- To manage the different types of connections by integrating a MMS.
- To prolong a sensor's battery life.
- To manage the database, whether remotely or on-sensor by using a (DB) management system.
- To provide a comfortable and easy-to-use platform.

Real-time data processing is an important functionality in our mobile WBAN platform. When a patient interacts with the BSDs, medical parameters should be automatically collected. The BSD can respond to many queries from different medical facilities at the same time. For example, a BSD can collect specific parameters that have been requested by a doctor and then send them back to be displayed on the doctor's GUI. Normal medical parameters are sampled periodically and stored in the database server. Abnormal medical parameters are sampled instantly and sent immediately to medical facilities as emergency packets associated with a time stamp through the reservation of priority time slots and real-time reporting. Emergency and on-demand queries have higher priorities than normally sampled (periodic monitoring) data.

To manage the different types of connections in our platform, we plan to develop a MMS that is able to gather signals from the BSDs and communicate with the Internet over a personal server (PS), which is smartphone-mobile-data-enabled (3G/4G), in order to expand the range of the WBAN platform and enable patient mobility as well as get exact patient positions via GPS for outdoor monitoring. Our proposed platform contains several BSDs and a personal server (smartphone) with Java-based software installed on it to manage two-way queries from the human body to the medical facilities and vice versa. It also includes the DB server, where each doctor can get direct access to his patient through a smartphone or via his PC terminal, which will typically be located at a medical facility. There is also a need to include direct access to the ER department through the Ambulance-VANET-enabled feature in case of critical events. In this case, we need to have three kinds of networks: one between the BSDs and the PS, another between the PS and the medical facilities (the base station), and the last between the base stations themselves. We need three types of software: for the BSDs (TinyOS), for the PSs (Java-based, as most of today's smartphones are Java-based) for the patient's and doctor's smartphones), and for the base stations (the PC terminals), which could be Java-based as well. To enable the easy and efficient processing, interpreting, analyzing, displaying, and storing of the gathered data.

We recommend that our platform be implemented in a TinyOS environment, because it is an open-source and lightweight operating system. TinyOS is an operating system that was designed specifically for a WSN; it supports ad hoc networks and time synchronization. TinyOS consists of nesC components. NesC is a C-embedded programming language. In addition, TinyDB is a WSN data model and query language that stores and buffers a sensor's data temporally to execute query commands in the WBAN. For example:

```
SELECT MAX(temp)
FROM sensor_table
WHERE temp > threshold_limit
SAMPLE PERIOD 1m
```

[37]

The above SQL operation selects the maximum temperature from the sensor table if the threshold limit is reached within the period of one minute. In Figure 2, the idea of implementing a TinyDB in the WBAN platform is illustrated. In terms of the back-end server (DB Server), there will be an integral database based on a standard MySQL that is able to manage the different types of queries and maintain our platform configurations as well as the medical records of each patient and location, including tables of the relevant BSDs that have been identified with IDs. Security and privacy roles should also be supported to prevent unauthorized logins.

Power consumption is one of the main concerns during the WBAN development. Taking into account the implementation of the power management mechanism, which can reduce power consummation, is highly recommended to overcome the power limitation associated with using WSN technology. For example, by turning off the SD radio while it's inactive, battery lifetime could potentially increase, i.e., by controlling the standby and sleep of the sensor's radio [38]. This feature can be used by choosing the suitable MAC layer.

We intend to design a small form factor for the BSDs. This is an important issue in our platform in order to enable patient mobility when an individual is

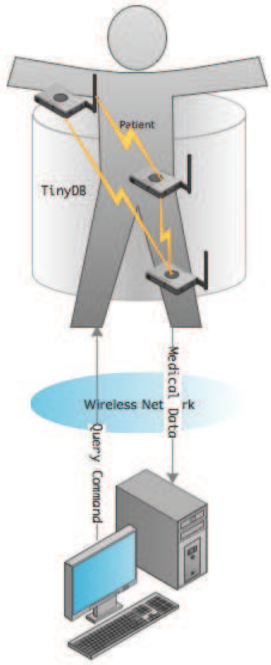


Figure 2.3: A TinyDB of the mobile WBAN platform

comfortably equipped with easy-to-use BSDs. Such a scenario will certainly improve the platform"s usability as well as the overall results that are collected from the creation of such a system.

The Second Tier

Hardware requirements Hardware requirements: The PS could be any PDA device, such as a smartphone or tablet with the specifications mentioned below:

- Integrated GPS.
- High system memory (RAM).
- High internal storage.
- High battery capacity in active and standby mode.
- Powerful system processor.
- Powerful graphic processor.
- Support the network management system.

Software Requirements Software requirements:

- To develop a friendly GUI.
- To process the collected data (events) efficiently.
- To capture the sensory data in real time.

However, the PS is an application or a piece of computer software that could be installed either on PCs or smartphones. In general, the PS is implemented using the Java programming language because of its flexibility and because most smartphones are Java-based. The PS should be interfaced and have a direct connection with the SDs and the back-end server in order to receive and forward medical data in real time. Once the PS receives event messages from the SDs, immediate action should be taken by comparing the medical data with the threshold limit to alert medical personnel of any critical developments. An SMS should also be sent or an ER agent should be called regarding the patient's situation. Moreover, the medical data should be forwarded to the back-end server. All of these tasks must be done in real time.
Developing an efficient GUI to visualize the aggregated data is strongly required in the PS of the WBAN platform. For example, displaying the heart rate graphically to medical personnel is very important in monitoring a patient's heartbeat. The PSs must support the ability to monitor many patients and receive multiple medical parameters from various patients simultaneously. Also, the PS should be able to wirelessly configure the parameters from the sensors for efficient patient monitoring. Each patient should have the ability to monitor himself by accessing a user-friendly interface on his PDA; this feature is ideal for home-based patients, who often need to monitor themselves continuously for a long time. See Figure 2.
However, the communication protocols in the WBAN platform must be secured and protected. In such a platform, the system architecture consists of many protocols, such as TCP/IP and cellular along with IEEE standards such as ZigBee/IEEE 802.15.4, Bluetooth, Wi-Fi 802.11). The use of all these protocols in WBAN platforms came about from the disadvantages associated with the radio frequency of a WSN based on IEEE 802.15.4. Enabling these communication protocols has the advantage of extending the range of the WBAN outside of medical institutions. For example, a patient can wear a smart-shirt and holds his PDA that is interconnected wirelessly. The PDA would need to be 3G-enabled so that once a health problem occurred, the smart-shirt would transmit an event message to patient's PDA to alert the remote medical personnel of the problem in real time.

The Third Tier

Hardware requirements Hardware requirements: The third tier is the database server or the back-end server that could be located centrally in the main building of

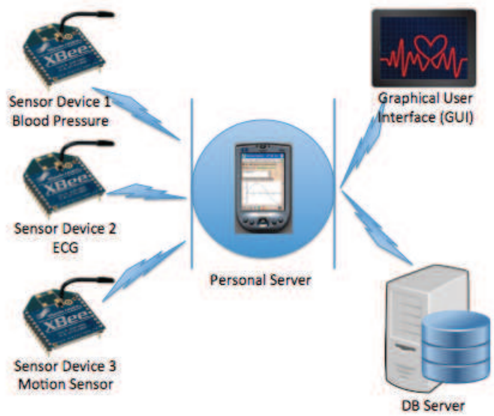


Figure 2.4: A WBAN platform

a medical institution. Some technical specifications should be allocated in the DB server:

- High system memory (RAM).
- High Internal storage.
- Powerful system processor.
- Powerful Graphic processor.
- Support the network management system.

Software Requirements Software requirements: The back-end server must consist of an integral database. The central database server in a WBAN platform is not necessary in some cases depending on the purpose of the WBAN platform. However, having a central database server is very useful to link the sensor's data with a patient's medical record. The authorized medical personnel can access the medical records remotely to examine and analyze the aggregated data of a relevant patient. The back-end server receives and stores the sensor's data through the PSs remotely. Usually, the back-end server is located in a medical institution, for example, in the central hospital of a city.
To relay the medical data in a database some security rules must be implemented. Not all medical staff should be allowed to gain access to the medical records. For example, doctors should be able to access all possible functions related to their patients, while nurses should be able to monitor their patients without having the right to access or modify medical records.

2.2 Mobility Model in WBANs

Movement from one place to another and the position of a given patient influence WBAN mobility as well as the communication between the BSDs and remote medical facilities.

Additionally, the WBAN network is limited to the human body, which means that in a small-scale network only one main coordinator is centralized enough to be placed on the human body with several BSDs, all of which are interconnected wirelessly. The star-topology is the ideal model because of the centralization of the WBAN. The type of communication is one-hop from the PAN to the BSDs and vice versa.

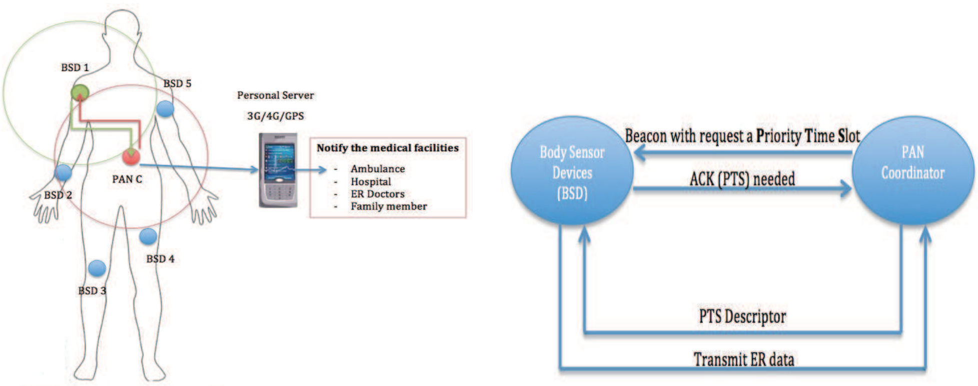
Also, to choose the right mobility model we have to know the exact scenario of the healthcare application. For instance, in our scenario, we want to apply the mobility model indoors and outdoors.

However, here we explain the use of the mobile WBAN platform in order to illustrate the main idea behind the proposed system architecture. Each patient is equipped with one or more biomedical sensor devices (motes). These motes are interconnected in a wireless network (Bluetooth/ZigBee). It is critical to use both physiological and environmental sensors when monitoring the activity of daily living (ADL) of various patients, especially those who are at high risk of experiencing an emergency or elderly patients who need ubiquitous monitoring.

The collaboration between motes would be useful in detecting certain diseases by processing of the data collected from the measured parameters. Event detection is usually performed inside the processor of the mote. Furthermore, the motes must transmit any abnormal parameters (detected events) in real time to the servers in order to alert relevant medical personnel and an emergency agent immediately. To achieve reliability when using the WBAN platform, online monitoring is required. This can be achieved through the visualization of the medical data of each patient via a GUI. Adapting the tracking system of a WBAN is also necessary in order to detect the exact position of a patient who is experiencing an emergency.

The mobile WBAN platform is mainly working as an ER agent to help those people who need immediate healthcare, but it can also be used to monitor patients continuously whether they are in a hospital, visiting multiple hospitals, or in their homes. To clarify the mechanism of the WBAN platform, a brief scenario is given as follows:

The motes sense the medical parameters such as the vital signs of a given -patient; the measured parameters will be processed on-sensor to check and detect the abnormal parameters. Then the sensor transmits the detected events to a PS in real time to alert medical personnel and the ER agent about the patient's situation by sending an SMS or calling them over GSM/3G/UMTC networking standards. The vehicular ad hoc network (VANET) can be used in the ambulance to support our mobility by creating a mobile network between the hospital and the patient in the ambulance. Then the medical data will be forwarded and stored to a remote back-end server over a secure TCP/IP connection standard. The back-end server must be



(a) To transmit the medical parameter (b) To report abnormal event by reserving PTS

Figure 2.5: The communication between the PANC and The BSD no.1)

able to maintain its integrity so that the database will be secure. Medical personnel can remotely retrieve a patient's medical data via a GUI through his PDA in order to examine and analyze the medical data. See Figures 3, 6, and 7.

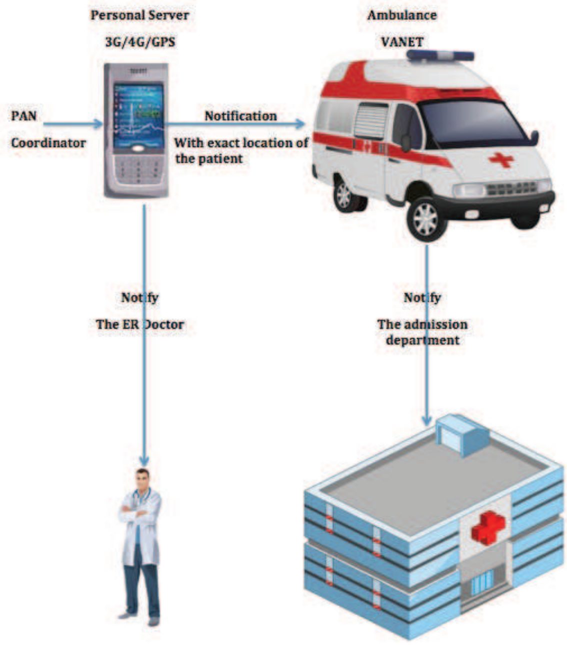


Figure 2.6: The ER model of the WBAN platform

As an example of an ER model that focuses on the ER communication of the mobile WBAN platform, once the abnormal data is detected, the BSD1 requests a priority time slot (PTS) by sending an acknowledgment packet to the PANC's beacon. In turn, the PANC will reply to the PTS descriptor and open direct access to BSD1 so that its ER data can be sent to its PTS. See Figure 4.

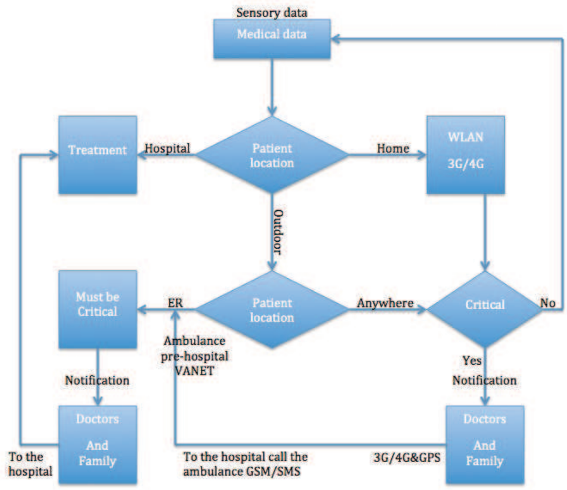


Figure 2.7: The mobility model based on the patient location

2.3 Disease diagnosis and epidemic tracking

In this section we give an example of how our WBAN platform could be used to deal with a serious disease. We will use only a two-tier (client-server) architecture in this example, i.e., only SDs and a base station will be considered. Later, we will enhance our diagnosis procedure by using classification methods in order to solve very sensitive epidemic diseases.

2.3.1 Disease Diagnosis

We chose influenza to illustrate the proposed platform. According to the American Lung Association:

> *"Influenza, or [the] flu, is a serious respiratory illness. It is easily spread from person to person and can lead to severe complications, even death."* [39]

Therefore, we chose this disease because it is a serious one and widely spread throughout the year. Moreover, it could lead to death in some cases. The symptoms of influenza are listed below:

- High temperature that can reach (40¡C)
- Feeling of general sickness, i.e., the patient cannot do daily activity properly
- Headache
- Red eyes with illness experienced when opening and closing them
- Runny nose with sneezing and a dry cough
- A rise blood pressure in some cases

Numerical Results

If all three symptoms accrued in the same time we can say this person is infected
Proof:

$$M_{i=3} \ = \ \{M_0 \ , \ M_1 \ , \ M_2\} \tag{2.1}$$

$$\sum_{i=0}^{2} M \ > \ 2 \ , \ infected \tag{2.2}$$

$$M_{(\ a,\ b\ )} \ = \left\{ \begin{array}{cc} 1,\ if\ M \ > \ 2, & infected \\ 0, & otherwise \end{array} \right\} \tag{2.3}$$

$$Person_i \ = \left\{ \begin{array}{cc} 1,\ if\ M \ > \ 2, & infected \\ 0,\ if\ M \ < \ 2, & not\ infected \end{array} \right\} \tag{2.4}$$

For example: $Person_{\ i}$, *has* 3 *motes*

$$if \ \sum_{i=0}^{2} M \ > \ 2 \ , \ infected \tag{2.5}$$

$$M_0 \ ; \ (1) \ \mathrm{S}kin\_temp \ = \ 40\ ^0\mathrm{C} \tag{2.6}$$

$$M_1 \ ; \ (2) \ Body\_motion \ < \ 1 \tag{2.7}$$

$$M_2 \ ; \ (3) \ blood\_\mathrm{pressur}e \ = 90 \ \frac{\mathrm{C}}{mn} \tag{2.8}$$

$$M_{i=3} \ = \ \{M_0 \ , \ M_1 \ , \ M_2\} \tag{2.9}$$

The input values are converted into the binary codes; 0=off and 1=on.

$$M_0 \ = \ 40\ ^0\mathrm{C} \ = \ 1 \tag{2.10}$$

$$M_1 \ =; 1 \ = \ 1 \tag{2.11}$$

$$M_2 \ = 90 \ \frac{\mathrm{C}}{mn} \ = \ 1 \tag{2.12}$$

So, the result of this patient is as follows:

$$\sum_{i=0}^{2} M = \ 1+1+1 = 3, \ so\ this\ person\ is\ infected \tag{2.13}$$

So, if we have four patients, as follows: $P_4 = \{P_0, P_1, P_2, P_3\}$. Table 2.1 shows the diagnosis results.

Motes	Patient0	Patient1	Patient2	Patient3
$M0$	0	1	0	1
$M1$	0	1	0	1
$M2$	0	1	0	1
Sum	0	3	0	3
Result	not infected	infected	not infected	infected

Table 2.1: Table of infected patients

According to these symptoms, we can identify the patient who is infected by using three sensors to see (skin temperature, body motion, blood pressure) whether all three symptoms have occurred. If they have, we can say that a person is infected as a part of the initial diagnosis. We will use a mechanism to compute disease detection. A factor such as population protocol can be considered by performing special computations that make it possible to detect the disease [40]. Once the event is detected, the event message will be sent to the base station to alert medical personnel that a specific person could be infected.

The scenario will be as follows: four volunteers are ready for an infection test. Each person is equipped with three SDs that are interconnected wirelessly, and the GPRS network is enabled on the volunteers' PDAs to detect their positions if needed. The base station is connected wirelessly with the SDs to aggregate and analyze the medical data that will be received from the SDs.

1. The result of the 1st volunteer: Skin temperature is 37.5¡C. The motion sensor signals reflect that the ADL is normal, and the blood pressure (BP) signal is 80 mm Hg. So we can say this person is not infected.

2. The result of the 2nd volunteer: Skin temperature is 40.1¡C. The motion sensor signals reflect that ADL is abnormal, i.e., the volunteer is not moving as usual. Moreover, her BP signal was between 90-95 mm Hg. So this person is initially believed to be infected and has to be checked by the doctor.

3. The result of the 3rd volunteer: Skin temperature is 36.7¡C. The motion sensor signals reflect a normal ADL, and the BP signal was between 70-74 mm Hg. So this person is not infected.

4. The result of the 4th volunteer: Skin temperature is 40.0¡C. The motion sensor signals reflect an abnormal ADL, and the BP signal was between 90-92 mm Hg. So this person is initially believed to be infected and has to be checked by the doctor.

As we have seen from this example, the possibility of applying such a system on a wide physical area to provide early detection in the infection area using a GPRS network and the movement of influenza during the year is very useful. This project could help in projecting the likeliest infected areas, thus making it possible to provide infected people with any necessary vaccines.

2.3.2 Epidemic Tracking

We extend the diagnosis process to include epidemic tracking in urban areas. We propose a new strategy for supervised learning in the detecting and describing of instances from different stages of disease in the case of epidemics.

We will provide a description of the status for each detected instance (e.g., by considering a healthy patient; an incubating, ill, or very ill patient; a recovering patient; and a cured patient). First, we associate a binary classifier to each sensor in order to detect whether a patient is ill or not. Then we collect all detected instances via the servers in order to cluster them using fuzzy algorithms. Finally, data-mining techniques can be used to interpret the semantics of each cluster that has been built.

Binary Classifier

Given a family of healthy/disease parameters, such as temperature, pressure, heart rate, ..., we denote $\mathbf{x}$ a parameter vector and x^j its j'th coordinate. We will build a binary classifier, $C(\mathbf{x})$, trained from a set of labeled parameter based on previously collected examples. In order to build the most efficient, robust, and fast classifier possible, we will investigate two well-known approaches: the geometric support-vector-machine (SVM) method and the probabilistic one based on logistics regression. In the following sections we describe each of the classifiers.

SVM classifier

For each patient from the learning sample, the label, denoted Y, is a random variable with a value 1 if the patient is ill and -1 if the patient is healthy. The parameter vector $X \in \mathbf{R}^d$ is a random vector of dimension d, the number of parameters.

We observe each patient $i = 1, \ldots, n$ from the learning sample of size n both Y and X, and we denote y_i and x_i, respectively, for the values of Y and X observed with regard to the individual i. Following [41], the SVM algorithm consists of estimating the parameters of the decision function $C(x)$ during a learning phase from the learning sample data $(x_1, y_1), \ldots, (x_n, y_n)$. After training, based only on the observed values x of its parameter vector X, the classification of a new patient (to the group of ill patients or to the group of healthy patients) is predicted according to the following decision rule.

$$\begin{array}{lll} y = 1 & if & C(x) > 0 \\ y = -1 & telse. & \end{array}$$

The classifier $C(\cdot)$ takes the form

$$C(x) = \sum_{i=1}^{n} w_i \varphi(x) + b, \tag{2.14}$$

where φ are some transformation functions to be appropriately selected, and the w_i and b are real parameters to be estimated from the learning sample. The estimation of these parameters is based on maximizing the margin between the class boundary and the training parameters provided that this separation exists for an appropriate choice of functions φ. These issues are handled efficiently within our platform and the efficiency-robustness of the corresponding classifier will be compared with the following probabilistic one.

A binary classifier through logistic regression:
In the logistic regression [42], the label Y is considered a random variable, taking a value of 1 if the patient is ill and 0 if the patient is healthy. The parameter vector $X \in \mathbf{R}^d$ is a random vector of dimension d, the number of parameters. The logistic regression is used to model the conditional probabilities

$$P(Y = 1/X = x)$$

and

$$P(Y = 0/X = x)$$

for a given value $X = x$ of the parameter random vector X. The model provides

$$P(Y = 1/x) = \frac{\exp\{a + b^t x\}}{1 + \exp\{a + b^t x\}}$$

and

$$P(Y = 0/x) = \frac{1}{1 + \exp\{a + b^t x\}}$$

where $b^t x := \sum_{j=1}^{d} b_j x^j$ is the scalar product of b and x, and $a \in \mathbf{R}$ and $b \in \mathbf{R}^d$ are the parameters of the model to be estimated from the training data $(x_1, y_1), \ldots, (x_n, y_n)$. We will use the maximum likelihood approach in order to estimate these parameters, which are known to be efficient and robust under incomplete specification. We denote $\widehat{a}$ and $\widehat{b}$, the obtained estimates of a and b, from the training data.

Based on the values of its feature $X = x_i$, the affectation of a new patients i to the group of ill people or to the healthy group is done through the following decision rule:

$$\begin{array}{lll} Y = 1 & \text{if} & \frac{\exp\{\widehat{a}+\widehat{b}^t x_i\}}{1+\exp\{\widehat{a}+\widehat{b}^t x_i\}} > 1/2, \\ Y = 0 & \text{otherwise.} & \end{array}$$

Fuzzy C-Means Clustering

Fuzzy clustering methods have proved their efficiency and their accuracy in different frameworks. In fuzzy clustering, the Fuzzy C-Means (FCM) algorithm is the best-known method [43]. FCM generates a fuzzy partition based on the minimization of the following functions:

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^m (d_{ij})^2 \tag{2.15}$$

where N is the data size, C is the number of clusters, m is the fuzzyfier (generally between 1.5 and 2), d_{ij} is the distance between data ν_j and the prototype of cluster i, and u_{ij} the membership degree of data ν_j to cluster i.

The solution of J is reached by updating the new membership degree (u_{ij}) and the new prototypes (β_i) in the following way:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{(m-1)}}} \tag{2.16}$$

and

$$\beta_i = \frac{\sum_{j=1}^{N} (u_{ij})^m . \nu_j}{\sum_{j=1}^{N} (u_{ij})^m} \tag{2.17}$$

In our platform, ν_j represents a vector that includes all the parameters measured by the BS, while FCM provides a partition. After this step, it is possible to categorize the patient depending on his symptoms.

Decision Stage

With "hard" clustering, algorithms divide a data set into separate clusters where each element belongs to a single cluster. However, in fuzzy approaches, clusters are not created. Instead, each element belongs to all the clusters according to a membership degree. So, the minimization of the fuzzy function J leads to partitions characterized by the membership degree matrix, the affinity, and the relations between all the data and clusters. Thus, after obtaining the optimal fuzzy partition according to functional J, a "defuzzyfication" step is needed. Finally, the data is assigned to the cluster that has the most important membership degree.

2.4 Conclusion

We have illustrated the application of the WBAN platform theoretically by studying some of the significant projects conducted with WBANs, and we have proposed a mobile WBAN system architecture. We applied the proposed system by providing an example of how it can be used. To apply such a system in a real-life scenario requires more investigations in terms of proper hardware use and the development of high-quality software that is compliant with medical quality standards. Moreover, we have to take into account the importance of the security, privacy, confidentiality, and integrity of the medical data.

Finally, we are working to identify a more practical mechanism that has the ability to detect an infection area and the movement of a given disease in a wide area. This requires a smart interoperability model. In facet, there are still some issues in this area need more investigation. Thus, in next chapters we propose three different scheduling approaches to solve the drawbacks that are caused by the non organized communication between the devices. The interaction between these devices have to be scheduled efficiently. For these reasons, the following chapters will overcome these issues.

2.5 Résumé du Chapitre 2

Ce chapitre présente d'abord les différents pré-requis nécessaires et les exigences à prendre en compte dans la conception d'une plateforme WBAN mobile. Pour finir, nous avons proposé une plateforme complète de détection et de suivi de maladie épidémique en cas de crise.

Une plateforme WBAN peut-être décomposée en trois parties principales :

1. Premier niveau : c'est le client (Capteur ou Sensor Device). un SD portable ou implantable placé sur le corps du patient.

2. Deuxième niveau : c'est le middleware (serveur personnel). Le serveur personnel est un logiciel informatique qui pourrait être installé sur un PC ou un smartphone.

3. Troisième niveau : c'est le serveur back-end (Database Server). Le serveur de base de données se trouve au sein de l'institution médicale.

Ces différents niveaux sont décrits ainsi que les pré-requis matériels et logiciels nécéssaire pour le bon fonctionnement de chaque niveau. En se basant sur ces pré-requis, un modèle de mobilité est détaillé pour la plateforme proposée. Ce modèle permet de gérer le déplacement du patient au sein de l'unité médical ou entre sa maison et l'hôpital.

Dans la deuxième partie de ce chapitre, nous avons proposé une plateforme complète qui permet le diagnostique et le suivi de maladie dans le cas d'une crise sanitaire. Cette plateforme est basée sur une architecture à deux niveaux (client-serveur). Pour le diagnostique de la maladie, un mécanisme de surveillance de différents types de capteurs physiologiques permet d'extraire un ensemble de conditions pour la détection de la contamination ou pas de l'élément. Pour le suivi de la propagation de la maladie, nous avons mis en place une classification des différents patients en patient en bonne santé, patient infecté, patient malade, patient très malade, patient en voie de rétablissement et patient guéri. Une fois que les états des différents membres sont classifiés le suivi du déplacement de la maladie devient possible. Ceci permettrait d'éradiquer plus facilement une maladie épidémique comme la grippe.

Chapter 3

Task Scheduling for Wireless Sensor Network

Contents

3.1 Introduction

3.1.1 Definitions

Task Scheduling in WSN is about deciding which node will be allowed to access the medium first. Two main scheduling requirements are needed to schedule nodes' tasks in the network: time slots and addressing assignments.

The wireless body area network (WBAN) is the most critical field in wireless sensor networks (WSNs). It must be a self-organizing network architecture, meaning that it can manage all the network architecture requirements efficiently. The WBAN usually contains at least two or more body sensors (BSs). Each BS sends or receives packets from or to the personal area network coordinator (PANC). The PANC is responsible for scheduling its child nodes' tasks. Scheduling tasks in the WBAN require a dynamic and adaptive process in order to handle emergency cases that can occur in any related-patient. To improve the most important parameters of the WBAN, such as energy consumption, throughput, duty-cycle and packet delivery.

3.1.2 Discussion

To schedule some tasks, the typical scenario consists of putting them in a queue; in this case, the traditional scheduling of first come, first served (FCFS) will be used. In some cases, FCFS is not applicable. For example, some nodes may have different priorities or sometimes may not have tasks to be scheduled. In such cases, the FCFS algorithm is not the best choice. Why? Because it could maximize the waiting time and minimize the throughput. Fairness could also be affected. Fairness is about sharing the medium among the nodes in the same equitable way. For this reason, in some applications, there is a need to have some exceptions.

3.1.3 Scheduling Assumptions

Let us suppose that there is a cluster-tree topology that has only one main coordinator, some routers, and many end devices. They are inter-connected wirelessly based on ZigBee/IEEE 802.15.4 protocol. In this network, some nodes have sampled some environmental data and are ready to send their data. On the other hand, in the same network, some of them do not have any data to send. In this case, we have two kinds of nodes: ready nodes and non-ready nodes. Furthermore, the ready nodes may have some urgent data that must be sent instantly, and some normal data that could be postponed to be sent later.

In static scheduling, algorithm nodes must be fixed in one position, meaning that this does not support mobility. The traditional way of communication in such an algorithm is to run the first task and to find the next etc. Sometimes, there may be high-priority tasks in the same queue, but it should be run under FCFS algorithm. The problem of this algorithm comes when computing the process time and repeating the same cycle periodically without changing the time duration. Repeating the same duty-cycle every time could increase the latency and reduce the packet delivery, since some nodes may have no data to send. See Figure 3.1.

Figure 3.1 represents an easy example of the scheduling algorithm when the arrival time in the queue before scheduling is different from the service time after scheduling each task. For instance, the highest priority task in red at arrival time was second in the queue order, but when starting to serve them, it comes first in terms of service time.

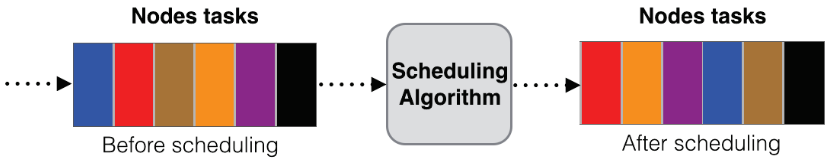


Figure 3.1: Scheduling Queue

For this reason, it is important to design a smart scheduling algorithm that can manage the network traffic with a waring of the stated situations.

3.1.4 Scheduling Metrics

Scheduling metrics are used to measure the quality standard of certain scheduling algorithm to know whether it is meeting the scheduling goals or not. Briefly, it is an output measurement.

There are three important metrics of scheduling: waiting time, throughput and fairness. These metrics reflect scheduling efficiency and measure scheduling quality. Thereafter, we need a scheduling analysis program to have some outputs to check the feasibility of the relevant scheduling process. Using the analysis program can help us determine and highlight the potential problems in the scheduling algorithm. In addition, it allows us to analyze and compare the different scheduling algorithms based on the scheduling metrics.

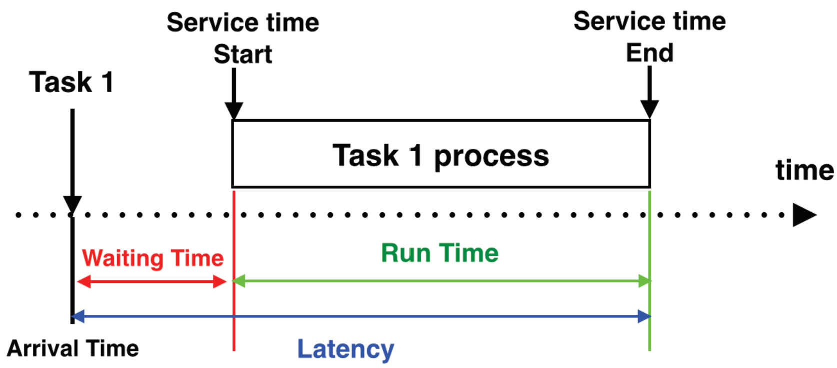


Figure 3.2: Scheduling Metrics

As seen in Figure 3.2, $task_1$ arrived at $time_i$ and it was waiting $time_{i+1}$. Then, the scheduler has started to serve it at $time_s$. The period between arrival time and serving time is called waiting time. Thereafter, the task is ruining for $time_r$ until is done. The entire time from arrival time until service time is completed is called the latency. In the same context, every task in the network will be processed in the same way. This means that if the waiting time at every task is long, the entire latency will consequently be huge. This problem can be solved by reducing the gaps of the waiting times in every duty-cycle, as well as managing duty-cycle dynamically and adaptively.

3.1.5 Scheduling Policy Goals

The goal of scheduling is to organize the network traffic properly, taking into account the different priorities of the network traffic (events). In addition, we design three algorithms that support the ready-to-send and not-ready-to-send nodes.

According to these considerations, the network traffic utilizing is focused on minimizing response time, maximizing throughput and sharing medium among node equitably, i.e., fairness.

In the same context, to maximize the throughput, the resources of the network must be used efficiently. Also, minimizing the overhead will lead to maximizing of the throughput.

There are a number of serving algorithms that can be applied on scheduling processes, such as Static, which is based on a First come first serve (FCFS) model, i.e. serving items are done sequentially. This algorithm runs all tasks until done in a queue fashion, meaning that it keeps the medium busy until the time slots finish.

Suppose that we have three nodes: $node_1$ requires 2 time slots, $node_2$ requires 4 time slots, and $node_3$ requires 3 time slots, respectively.

N_1 starts at 0 until 2 ts, N_2 starts at 2 until 6 and N_3 starts at 6 until 9.

Where waiting time for $N_1 = 0$ and waiting time for $N_2 = 2$ and waiting time for $N_3 = 6$. Hence, the average waiting time $= (0 + 2 + 6)/3 = 2.6$.

Where completion time for $N_1 = 2$, Completion time for $N_2 = 6$ and Completion time for $N_3 = 9$. Hence, the average completion time $= (2 + 6 + 9)/3 = 5.7$

Our dynamic proposals are based on our algorithm, which is called First Ready First Serve (FRFS). They are considered as proactive algorithms, meaning that the decision-making will be done according to a set of predictive schedules based on the traffic pattern. This allows the ready-to-send nodes and the priority nodes to send their data before the others.

The dynamic algorithm is served as First - Ready - First – Serve; in this algorithm, the first-ready-first-request will be served first, wherein the Dynamic-Priority algorithm is served as First - Priority - First - Served and in this algorithm (the first-ready-first-priority), regardless of its req number. The node will be served based on two assumptions: first, it has to be a ready-node, and second, it has to be flagged as a prioritized node. For example, if $node_1$ is $ready - node$ and has req_1 and $node_2$ is also $ready - node$ and has req_2 and flagged as prioritized node, then in this case, $node_2$ will be served before $node_1$. As seen in Figure 3.3, the packets (events) arrived at the scheduler port, then the classifier algorithm will organize them in a number of levels. The highest level is the most prioritized events and the lowest level is the lowest priority events.

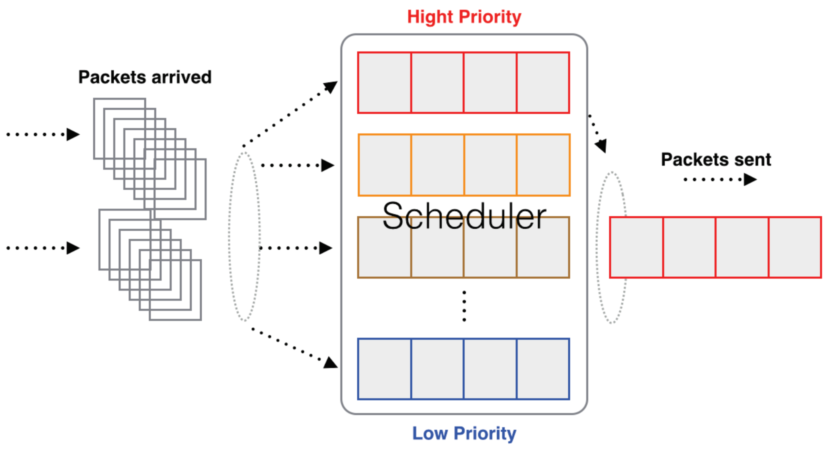


Figure 3.3: Scheduling Classifier

3.2 Scheduling Approaches over Cluster-Tree Networks

Scheduling over a cluster-tree network based on ZigBee/IEEE 802.15.4 require a synchronization process. The synchronization process is done via beacon frames. Applying some rules on this process is very challenging. Some collisions could occur if beacon frames are not scheduled properly. Cluster-tree networking consists of one main parent node, some parent-child router nodes, and leaf nodes. The relationship is computed by using $Cskip$ function, as stated in Chapter One. Furthermore, this relationship is maintained by using the association and disassociation services.

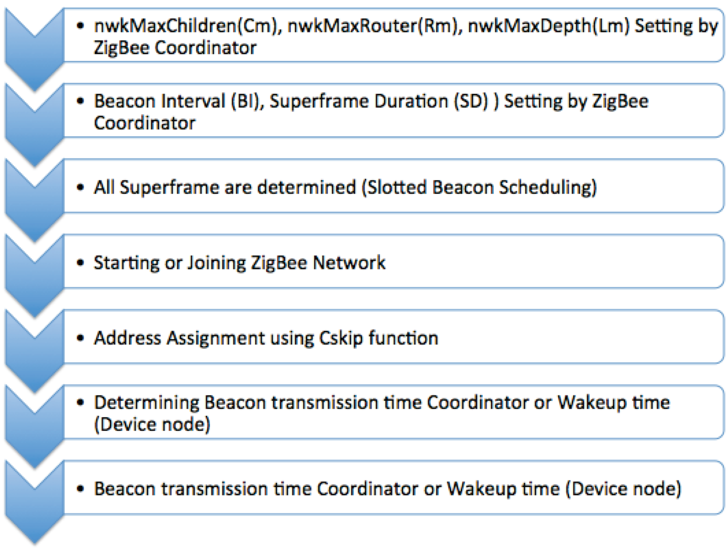


Figure 3.4: Cskip process

In the *Cskip* flowchart shown in Figure 3.4, there are a number of steps for managing the cluster-tree network. The first step is to set the number of maximum children in each cluster, the maximum number of each child router in the network, and the maximum number of depth that can be allowed in the network. These three parameters: Cm, Rm and Lm are set by the PANC. Thereafter, the PANC and its Router children will synchronize the network by using beacon interval (BI), SuperFrame Duration (SD). The PANC is the only device in the cluster-tree network that is responsible for setting the BI and SD. As stated previously, the cluster-tree works in beacon-enabled mode, which means that it is a slotted beacon scheduling process. Then, the network is ready to accept new joining devices by assigning them addresses using *Cskip* function. Each SuperFrame in every duty-cycle is divided into two main parts, active and non-active periods. Both the active and non-active portions of the medium will be maintained by using the beacon frames. Finally, the arrangement of the packet transmission must be scheduled to avoid collisions.

The slotted beacon scheduling process requires the definition of a SuperFrame that is constructed and managed by the PANC. Every SuperFrame is bounded by two beacons. The period between them is called BI. The SD represents only the active period. The active period is divided into 16 time slots equitably. Within this period, every device in the network can send its packets. The transmission of these packets has to be organized in time slots, i.e., every device has to send packets at a different time than the others to avoid collision. Based on the standard of ZigBee/IEEE 802.15.4, the inactive period is optional when $BI > SD$.

Within this period (the optional inactive period), every node can turn off its transceiver to avoid the unneeded energy consumption. To determine both the SD and the BI, there are two important parameters: first is the SuperFrame Order (SO) and the second is the Beacon Order (BO). They are defined as follows:

$$\left.\begin{array}{l} BI = aBaseSuperFrameDuration \cdot 2^{BO} \\ SD = aBaseSuperFrameDuration \cdot 2^{SO} \end{array}\right\} for 0 \leq SO \leq BO \leq 14 \tag{3.1}$$

$$Duty - Cycle = SD \div BI = 2^{SO-BO}, \tag{3.2}$$

The (BO) represents the SuperFrame length, whereas the (SO) represents the active length of the SuperFrame.

As seen in Figure 3.5, forming the network is maintained by exchanging the management messages (control messages) between the OSI layers of the ZigBee/IEEE 802.15.4. The PANC request forms the network via the ZigBee NWK from the MAC layer of 802.15.4. The PANC will receive confirmation messages from the MAC layer of 802.15.4 after performing energy scans and active scans to select an ID of PANC and channel.

Figure 3.6 shows the mechanism of the association process between Routers and end devices using beacon frames. The Routers send beacon frames to advertise themselves; then, the end devices will join the network according to the acceptance and confirmation messages from their parent routers.

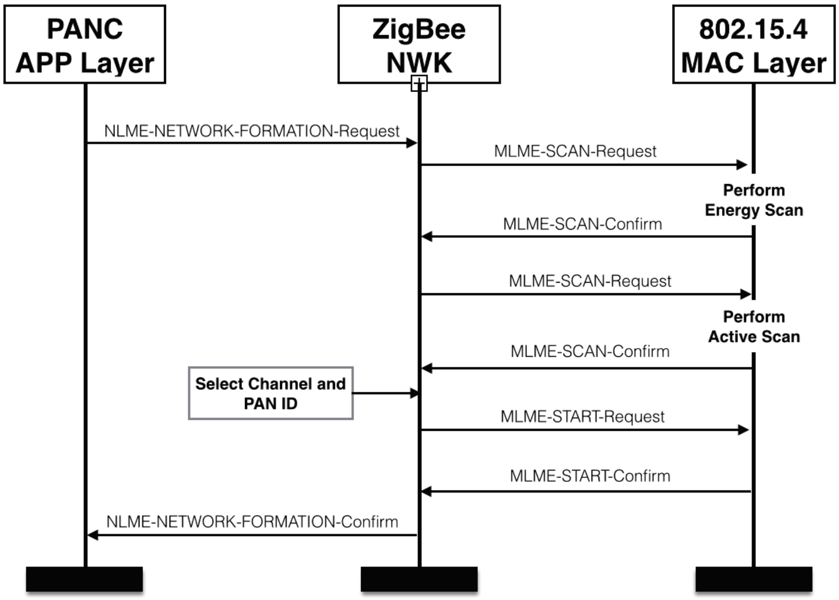


Figure 3.5: The network formation process

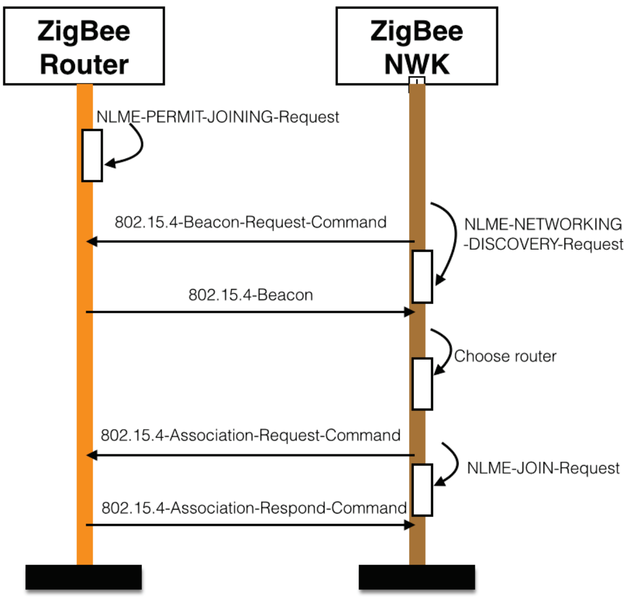


Figure 3.6: The association process

Based on these services and processes, we designed our algorithms. Hence, our approaches have been developed to work over cluster-tree networks.

3.3 Semi-Dynamic Scheduling Approach (SDS)

3.3.1 Design

This section presents Semi-Dynamic Scheduling (SDS) process for Wireless Sensor Networks (WSN). We combined two types of scheduling in this study: the static scheduling for coordinator nodes and the dynamic scheduling for leaf nodes. We consider a cluster-tree topology in our study with one main coordinator Personal Area Network Coordinator PANC, several child routers and several end devices (leafs). We have several WSN nodes that are identified as $s_1, s_1, ..., s_n$. Each one of those sensors will do its task within its time offset (periodically) BI_i. To accomplish its task ($SD_i \leq BI_i$), it needs an amount of time SD_i. In order to do so, each cluster-tree is composed of a unique coordinator, some routers, and several end devices. The job of the main coordinator is to establish the network, distribute the time offset scheduling and, since we work on the ZigBee/IEEE 802.15.4 beacon-enabled mode, the ZigBee Coordinator will synchronize all its children by sending a periodic beacon. The purpose of that is to allow each mote to talk within its time offset to avoid collisions as shown in Figure 3.7. The router's task is to synchronize its children by receiving data packets and forwarding them to its parent until the final destination, which is the ZigBee Coordinator.

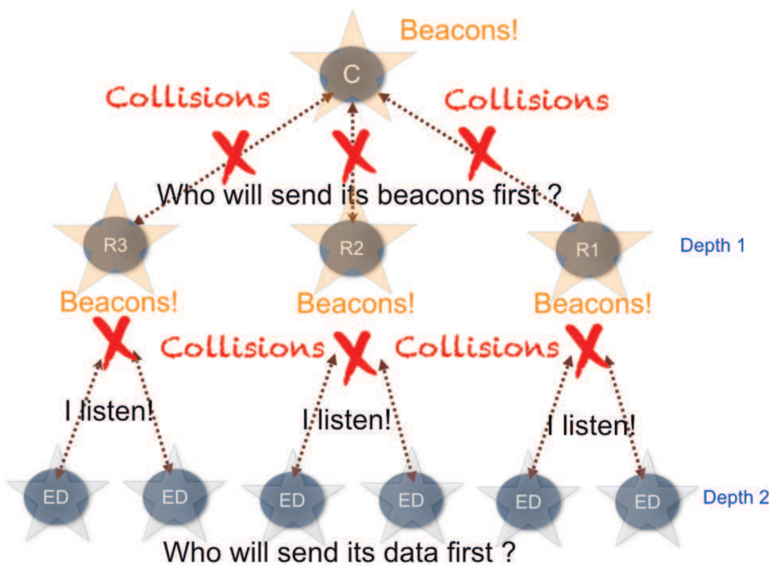


Figure 3.7: Beacon Collisions

We consider that there are two main types of scheduling. The first one is static scheduling based on the time division approach multiple access (TDMA). The second is the first part of our proposals, which is dynamic preemptive scheduling based on the TDMA. We assume that each type is important; hence, we could merge both of them to improve the performance of the network. See Figure 3.8.

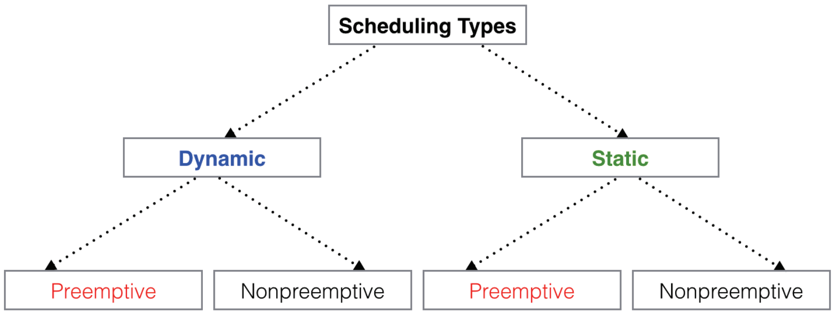


Figure 3.8: Scheduling types

Some advantages and disadvantages of static and dynamic scheduling are as follows: Static Scheduling:
Advantages:

- Fix the topology (Cluster-tree)
- Increase performance.
- Decrease packet loss ratio

Disadvantages:

- High latency
- Waste energy

Dynamic Scheduling:
Advantages:

- The mobility of nodes.
- Energy consumption
- Scalable

Disadvantages:

- Shape of topology unknown
- Rescheduling process waste memory.

We assume that the routers will have static addresses and static time offsets. Let's suppose that the number of routers are usually less than the number of end devices, and that each router or (FFD) Full Function device is capable of serving a number of end devices or (RFDs) reduced function devices. In that context, we assume that the end devices will be serving dynamically. Scheduling the end devices dynamically will reduce waiting time for those nodes that have ready-to-send sensory data that needs to be sent before the others. We also consider that not all the nodes always have data to send (i.e., not necessary to have a predefined time offsets for all nodes). Theoretically, these assumptions will minimize waiting time and maximize throughput by saving some time that has been lost in the not ready-to-send nodes. According to these assumptions, we have started by designing a scheduling algorithm that will schedule the set of router nodes statically and the set of leaf nodes dynamically.

We propose a Semi-Dynamic Scheduling approach (SDS) for WSNs, which performs self-generation of the source address assignment by the PANC to its FFD children, which have been accepted to behave as routers, and each one of them will have a predefined time offset to send its packets based on the traffic pattern. Each end device can send its packets to the coordinator through its parent router once it is ready-to-send without needing to wait until the next duration. In static scheduling, the source addresses and time slots are already pre-defined for each node in the network.

The next part of this section presents the static scheduling algorithm for the routers and the dynamic scheduling algorithm for the end devices. In the static model, the time slot assignments are predefined depending on the sensor address. To explain the difference more, we show a part of the two algorithms. The first one presents the static model and the second presents the dynamic one.

Algorithm 1 Static Scheduling Algorithm

pkt = receive(beacon_scheduling_pkt); ▷ Receiving Scheduling Request
switch(pkt -> source_address) {
    case 0x0001:
        Allocate SD_1 Time Slots from 0 to SD_1-1;
        break;
    case 0x0002:
        Allocate SD_2 Time Slots from SD_2 to SD_2+SD_1-1;
        break;
    ...
    case default:
        break; ▷ Not Scheduable because the sensor ID was not configured
}

In the algorithm 1, when the coordinator receives a request from a router, it will respond depending on the sensor ID. A list of predefined IDs is already stored in the coordinator. For example, if the source address is 1, then the coordinator will assign the first time slots to this node. The allocated time slots are related to the need of each node and are equal to SD_1. Then, the same procedure is repeated for Nodes 2, 3 and so on. If the node ID is unknown, then the coordinator will refuse the association and the node will not be able to send beacons. Therefore, this will ensure that different nodes will send beacons alternatively.

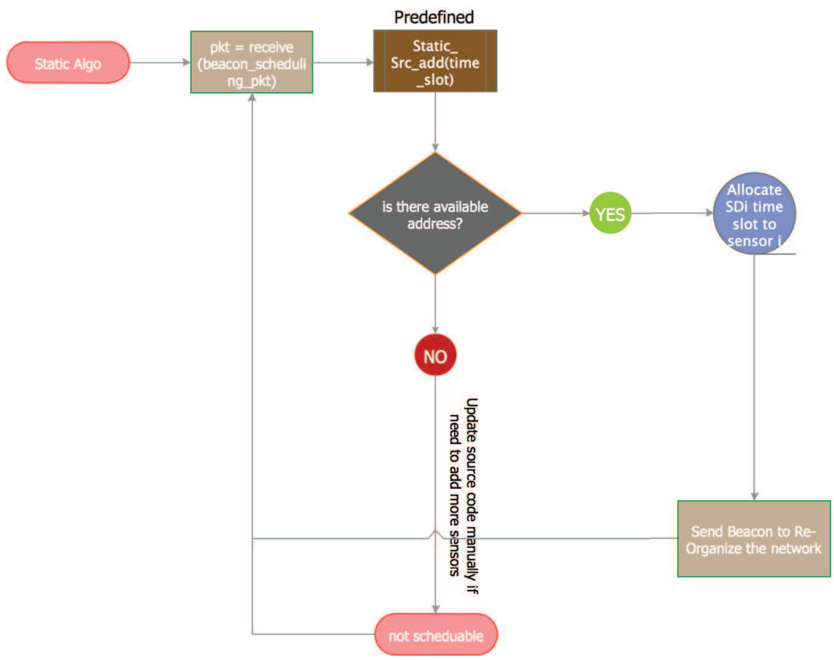


Figure 3.9: Static Algorithm

Algorithm 2 Dynamic Scheduling Algorithm Initialization

$Scheduable = False$; ▷ Initialization of scheduling result
$BI_{minor} = 8$; ▷ Initialization of minor Beacon Interval
$BI = 4$; ▷ Initialization of number of Beacon Intervals
$BI_{major} = BI * BI_{minor}$; ▷ Initialization of major Beacon Interval
$T_s[BI] = \{0, BI_{minor}, 2 * BI_{minor}, ..., (BI - 1) * BI_{minor}\}$; ▷ Initialization of First Available Consecutive Time Slots
$A \leftarrow \{BI_i = 2^{BO_i}\}, \ \forall\, i, \ 1 \leq i \leq P$; ▷ The set of beacon intervals
$BI_{min} = 2^{BO_{min}}$; ▷ The minimum beacon interval
$\overline{A} \leftarrow SortBeaconInterval(A)$; ▷ Organizing in the increasing beacon order

The algorithm 2 represents the first steps of the dynamic scheduling algorithm initialization. To illustrate our approach, an example of sensors requirements is represented in Figure 3.11. We set here the BI_{minor} to 8 and the $BI_{major} = 4 * BI_{minor}$ to 32, as shown in Figure 3.12. At the beginning, the first available Time Slots are set to $T_s[4] = \{0, 8, 16, 24\}$. Then, the coordinator receives all the sensors requirements stored in Table $A = \{8, 32, 16, 32, 8\}$, as depicted in Figure 3.11. After that, we compute the minimum of the Beacon intervals in $BI_{min} = 8$. Finally, we sort the requirements in Table $\overline{A} = \{8, 8, 16, 32, 32\}$ with increasing beacon intervals. This will make the scheduling easier.

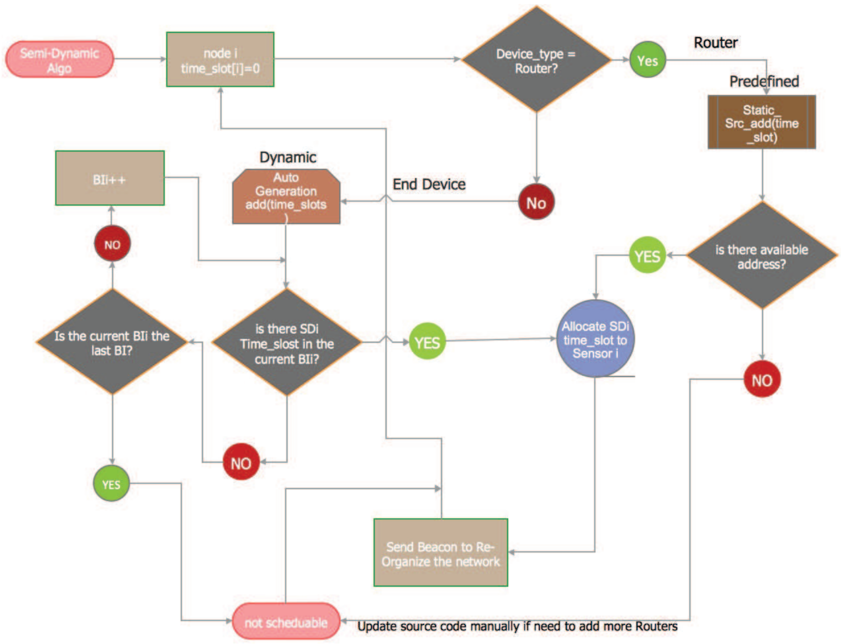


Figure 3.10: Semi-Dynamic Algorithm

The algorithm that is presented in Figure 3.9 shows that the PANC receives a request from an R router, and then the request will be confirmed only according to a pre-defined address list. Usually, the PANC in a static system contains a list of addresses. Every new joining node will be assigned by a pre-defined network address with the requested time slots. Note that the time slots will not be only provided upon the node request, but based on the schedule availability on the PANC. In this case, if the requested time slots are greater than the pre-defined slots to a particular node, the PANC will assign only the scheduled slots. In case of unavailable time slots, the PANC will refuse the request. Also, if the undefined node needs to join the network, it will be neglected by the PANC.

Figure 3.10 presents our algorithm of the semi-dynamic scheduling approach. It shows two phases: the first phase is to initialize the scheduling process by the PANC. Secondly, once the network is constructed and synchronized, every sensor node (SN) in the network will be assigned a unique address. The time slot assignment will be generated automatically by the PANC according to the traffic pattern. An example of the scheduling process is given to illustrate the approach. As shown in Figure 3.11, a set of beacon intervals (BI) are configured for the minimum period $BI_{minor} = 8$. In the example, there are four BIs $BI_{major} = 4 * BI_{minor}$. The total is 32, which represent the major BI = 32 time slots. Every eight time

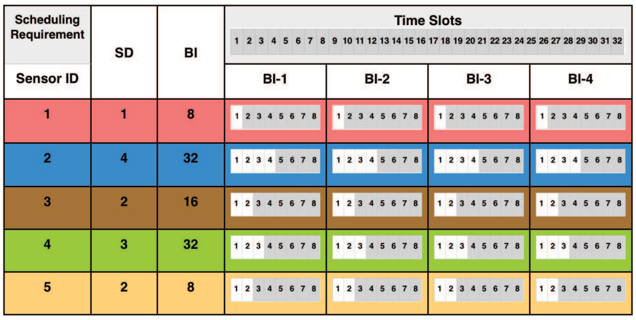


Figure 3.11: Sensors Scheduling Requirements Example

slots are bounded by two beacons, as shown in Figure 3.12. The time slot availability is set to $T_s[4] = \{0, 8, 16, 24\}$. The scheduling requirements for all nodes will be stored in Table $A = \{SensorID, SDi, BIi\}$. The example shows Table $A = \{\{1,1,8\},\{2,4,32\},\{3,2,16\},\{4,3,32\},\{5,2,8\}\}$. The minimum beacon intervals computation are done in $BI_{min} = 8$. Lastly, the requirements sortation in the table $\overline{A} = \{\{1,1,8\},\{5,2,8\},\{3,2,16\},\{2,4,32\},\{4,3,32\}\}$ are done increasingly based on BIs, which facilitate the scheduling process.

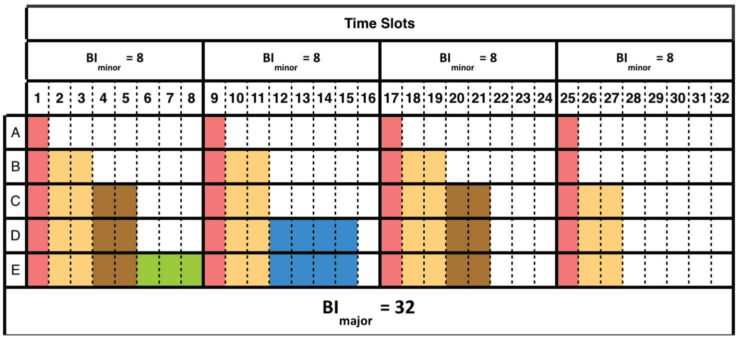


Figure 3.12: Example of the Scheduling Process

The PANC in the dynamic part will synchronize its child nodes based on time slot availability. The allocation of time slots is based on the auto-generation of network addresses. The dynamic addressing mechanism allows every new joining node to dynamically have a unique address. Each Router will be served based on the First Come First Served (FCFS) algorithm, whereas leaf nodes will be served based on the First Ready First Served (FCFS) algorithm. The FFD (routers) will be authorized to synchronize their child nodes according to the PANC timetable (scheduling algorithm). The organization of the timetable is maintained on the PANC according to the information obtained from exchanging the control messages mechanism.

In the following example, we give an overview of the algorithm description that is briefly depicted in Figure 3.12. In the next five steps, we explain the concept of the scheduling process:

Step 1 : The sensor with ID 1 needs one time slot every 8 BI. It allocates them in the first Beacon Interval and as a result : $T_s[4] = \{1, 9, 17, 25\}$.

Step 2 : The sensor with ID 5 needs two time slots every 8 BI. It allocates them in the first Beacon Interval and as a result : $T_s[4] = \{3, 11, 19, 27\}$.

Step 3 : The sensor with ID 3 needs two time slots every 16 BI. It allocates them in the first Beacon Interval and as a result : $T_s[4] = \{5, 11, 21, 27\}$.

Step 4 : The sensor with ID 2 needs four time slots every 32 BI. The coordinator can't allocate them in the first Beacon Interval because there is no enough free time slots. Then, it allocates them in the second Beacon Interval and as a result : $T_s[4] = \{5, 15, 21, 27\}$.

Step 5 : The sensor with ID 4 needs three time slots every 32 BI. It allocates them in the first Beacon Interval and as a result : $T_s[4] = \{5, 11, 21, 27\}$.

To explain this algorithm further, we will detail in the next paragraph the example presented in Figure 3.12. In this example, the scheduling result is obtained after 5 steps:

Step 1 : At every $BI_{minor} = 8$, the Red Sensor needs only one time slot. They will be allocated at the first $BI1_{minor} = 8 = T_s[4] = \{1, 9, 17, 25\}$.

Step 2 : At every $BI_{minor} = 8$, the Blue Sensor needs two time slots. They will be allocated at the first $BI1_{minor} = 8 = T_s[4] = \{3, 11, 19, 27\}$.

Step 3 : At every $BI_{minor} = 8 * 2$, the Brown Sensor needs two time slots. They will be allocated at the next available $BIi_{minor} = 8 = T_s[4] = \{5, 11, 21, 27\}$.

Step 4 : At every $BI_{minor} = 8 * 4$, the Green Sensor needs two time slots.They will be allocated at the next available $BIi_{minor} = 8 = T_s[4] = \{5, 15, 21, 27\}$

Step 5 : At every $BI_{minor} = 8 * 4$, the Yellow Sensor needs three time slots.They will be allocated at the next available $BIi_{minor} = 8 = T_s[4]\{5, 11, 21, 27\}$.

The novel design of our merged algorithm has brought a new concept of scheduling execution over TelosB motes based on the ZigBee/ IEEE 802.15.4. This concept can be applied to the other considered protocols with minor adds-on and/or changes.

In this part, we illustrated our algorithm by providing an example of the scheduling execution process. We outline the advantages of both types:-

Static Scheduling:

- Optimizing power consumption by reducing the usage of the energy on the PANC because the network organization is pre-defined.
- High performance in a fixed environment and small-scale network.

Dynamic Scheduling:

- Optimizing the network scalability by enlarging the network leafs automatically.
- Automating the assignment of the network address as well as the time slots at the network leafs.

At the end of this part, we noticed that the possibility for improving our algorithm to be a fully dynamic scheduling process. For this reason, we have optimized our previous work in the following section.

3.3.2 Implementation and Performances Analysis

We have validated our proposal by using the open-source code of TinyOS 2.1.2 / ZigBee. We modified it and improved it to suit our proposal. The operating system that we used is Ubuntu 12.0.4. In terms of hardware, we used TelosB motes. For evaluating our experiment, we chose Z-Monitor to monitor the network behavior by sniffing the generated packets. Then, we divided our evaluation into six levels based on the total number of packets.

In this experiment, we used sixteen TelosB motes (sniffer, coordinator, 5 routers and 9 end devices).

Type of Scheduling / Summary of Testing	Static1	Static2	Static3	Static4	Static5	Static6
Total No. of Packet	750	1000	1500	2000	2500	3000
Average No. of Packet/Sec	1.902	1.799	1.843	1.844	1.839	1.860
Average Packet Size/bytes	8.101	7.903	7.864	7.854	7.876	7.901
Total No. of Bytes	6032	7878	12022	15997	19864	23789
Average No. of Bytes/Sec	15.322	14.109	14.293	14.342	14.493	14.642

Figure 3.13: Static Scheduling Traffic Summary

Type of Scheduling / Summary of Testing	Semi-D1	Semi-D2	Semi-D3	Semi-D4	Semi-D5	Semi-D6
Total No. of Packet	750	1000	1500	2000	2500	3000
Average No. of Packet/Sec	2.199	2.025	1.992	1.988	1.994	1.976
Average Packet Size/bytes	8.187	8.109	8.112	8.134	8.122	8.145
Total No. of Bytes	6521.0	8234	12321	16455	20899	24432
Average No. of Bytes/Sec	16.990	16.002	15.894	15.783	15.609	15.533

Figure 3.14: Dynamic Scheduling Traffic Summary

Figure 3.13 and 3.14 shows the traffic summary of each type. We found that the performance of the network is improved in our proposal. Also, we have noticed that level-by-level the performance is increased.

In Figure 3.15 and Figure 3.16, we have six levels of indicator values (total number of packets), average number of packets per second, and the average number

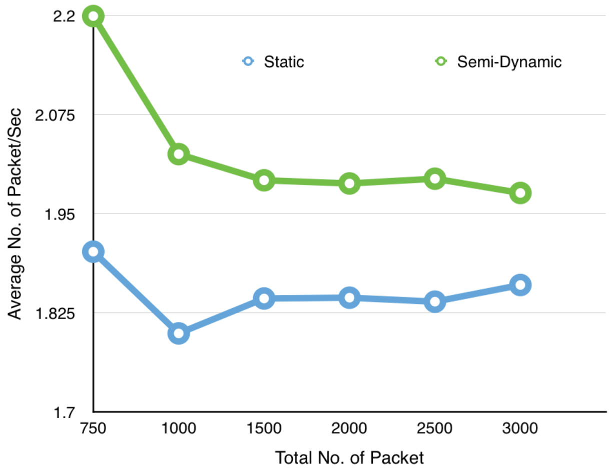


Figure 3.15: Average Packet/Sec

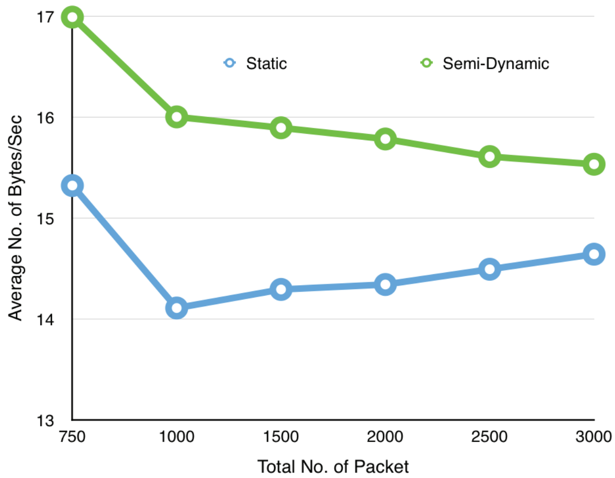


Figure 3.16: Average No. of Byte/Sec

of Byte/Sec that indicate where the level of each line is. Although we could evaluate each level with the green line and the blue one, which represent the scheduling type (static or Semi-D), it is more efficient to use a different level to test the network performance efficiently. Our line graphs represent that Semi-Dynamic Scheduling decreases the latency ratio, increases throughput capacity, energy consumption in every ED and finally, provides efficient duty-cycle management.

3.4 Efficient Dynamic Scheduling Approach (EDS)

3.4.1 Design

The crucial role of this part is to design a fully dynamic scheduling algorithm that will improve the overall performance of the network. Previously, we only applied the dynamic algorithm on the leaf nodes (EDs), but in this section, we apply it throughout the entire network, including the routers and the EDs. The PANC will synchronize its child nodes based on the traffic pattern, meaning that the scheduling competition will be done instantly over the PANC.

	Static				Semi-Dynamic				Dynamic			
Device Type	ID	Address	Time-slots	Mobile	ID	Address	Time-slots	Mobile	ID	Address	Time-slots	Mobile
PANC	0	0x0000	Provider	no	0	0x0000	Provider	no	0	0x0000	Provider	yes
Router	1 - 4	0x0001 to 0x0004	Pre-defined	no	1 - 4	0x0001 to 0x0004	Pre-defined	no	auto-generated	Auto-Assigned	Auto-Compute	yes
End Device	5 - 20	0x0005 to 0x0020	Pre-defined	no	auto-generated	Auto-Assigned	Auto-Compute	yes	auto-generated	Auto-Assigned	Auto-Compute	yes

Figure 3.17: A comparison of the scheduling approaches

The dynamic slot scheduling algorithm is maintained by having different time constraints executed over wireless sensor networks (WSN). These tasks may have different priorities. They are usually repeated periodically. We present an algorithm that is able to provide a dynamic scheduling procedure for all tasks. In the actual model, the scheduling algorithm is executed on the network coordinator that controls time slots distribution over its network devices. This algorithm is implemented over TelosB motes.

We consider a WSN composed of n nodes identified as $s_1, ..., s_n$. Each sensor will repeat a periodic task (with a period BI_i).
Each sensor s_i needs the amount of time SD_i to achieve its task ($SD_i \leq BI_i$). Each sensor has its own clock c_i. We consider that sensor clocks are synchronized. This is a realistic assumption, since we work on the ZigBee/IEEE 802.15.4 beacon-enabled mode, which means that the ZigBee Coordinator will synchronize all its children by sending a periodic beacon: $c_i(0) \neq c_j(0), \forall i \neq j$. We assume that each period SuperFrame duration (SD) $T_i (\forall i \neq n)$ has the value T. That means that all sensors

calibrate their task on the period beacon interval BI_i. Each sensor has an unique identifier, $\forall(i,j) \in V^2, d(i,j) < R$ all sensors are in the same radio range.

The Coordinator will have a dynamic transmission offset that will be distributed dynamically for all its associated nodes that have been accepted to act as a router. Any full function device (FFD) can be a router, but in order to do so, there is a need for a prior negotiation with the PANC that is located in the same transmission range.

In this study, we consider 4 routers.$PC\{Ri = (\mathrm{S}D_i, BI_i)\}, for\ 1 \leq i \leq P$ And 15 EDs, $PC\{EDi = (\mathrm{S}D_i, BI_i)\}, for\ 1 \leq i \leq P$

Algorithm 3 Dynamic Scheduling Algorithm Scheduling Verification

for $(i \in \overline{A})$ **do** ▷ Loop over Sensors
    for $(k = 0, k < BI, k++)$ **do** ▷ Loop over Beacon Intervals
        for $(l = 1, l < BI_{major}/BI_i, l++)$ **do** ▷ Loop over sensor requirements
            if $((T_s[k] + SD_i) * (l+1) < (k+1) * BI_{minor} * (l+1)))$ **then** ▷ Verify if there is enough free slots
                allocate $Slots\ from\ T_s[k] * (l+1)\ to\ T_s[k] * (l+1) + SD_i$;
                $T_s[k] * (l+1) = T_s[k] * (l+1) + SD_i$;
                if $(l == BI_{major} - 1)$ **then** ▷ Sensor i is Scheduable at Beacon Interval k
                    Scheduable = True;
                end if
            else
                Scheduable = False
                break; ▷ Sensor i Not Scheduable at Beacon Interval k
            end if
        end for ▷ End of loop over sensor needs
    end for ▷ End of loop over Beacon Intervals
end for ▷ End of loop over Sensors

In the algorithm 3.18, we set: A $= \{BI_i = 2^{BO_i}\}, \quad \forall\ i,\ 1 \leq i \leq P$. It is the number of BIs of all routers at the same transmission range.

The presented algorithm 3.9, as mentioned previously, has some drawbacks, such as not supporting a large-scale network, which means that it is non-scalable. Also, it shows high latency on every duty-cycle, which is caused by the dimensional waiting time. Every node must wait for its talking time every fixed duty-cycle. Generally, this process lacks flexibility in terms of synchronizing the network. In addition, it does not support mobility, i.e. nodes have to be fixed in one location. Hence, the main contribution of this part is to overcome these drawbacks.

The algorithm in Figure 3.18 shows the dynamic scheduling process. The length of each period is depicted as T_n time units in seconds. Firstly, the duty-cycle management is maintained on-process by the PANC. Secondly, every node's task will be performed based on the traffic pattern, i.e., it is not a pre-defined task.

The execution of each duty-cycle will be balanced based on the network re-

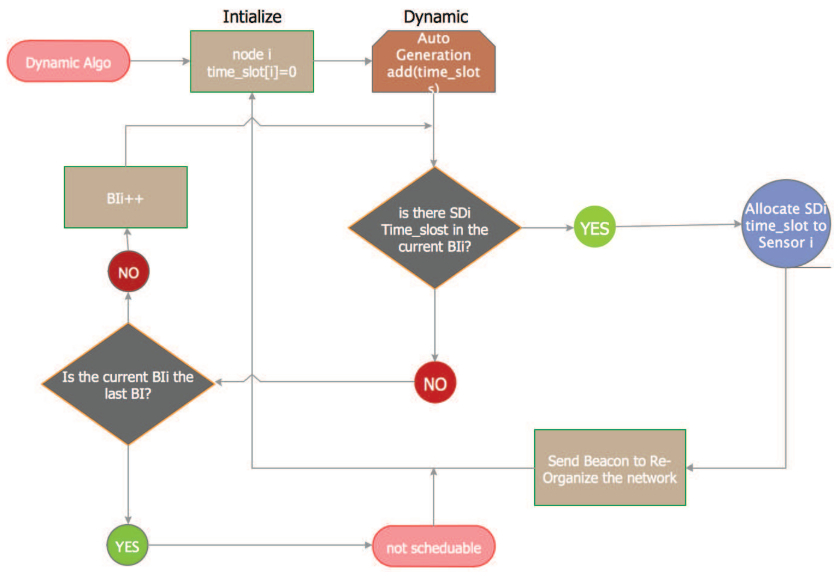


Figure 3.18: Dynamic Algorithm

quirements. Allowing the new joining node to associate will be confirmed based on its functionality. Therefore, the new node will be initialized by: $Node\ i = TimeSlot\,[i] = 0$. After confirming the association by the PANC, the related node will be able to exchange its messages within the network based on its functionality, whether FFD or RFD.

The PANC decides the priority for each event based on the First-Ready-First-Serve algorithm, with the ability to change every duty-cycle according to the nodes' activities, i.e., some nodes do not have data to send. In this case, the PANC will re-schedule its devices to avoid the network delay.

3.4.2 Implementation and Performances Analysis

We tested our system using two methods. The first one uses the OPNET Network Simulator and the second one uses the TelosB motes implementing the TinyOS operating system.

Simulation Results

In this section, we present the performance evaluations of our simulation models. The main project contains four scenarios for each scheduling algorithm. The feasibility of our work is done by comparing each simulation model against the static model.

Under the OPNET modeler, the ZigBee model contains four models according to the network layers of the ZigBee definition standards, the application model, the MAC model, the CSMA/CA model under the MAC Model, and finally the network model. By using these models, the simulation process can be analyzed easily. The behavior of each scenario can be configured and analyzed using two functions: the efficient graphical user interface (EGUI) and the discrete event simulation graphs.

We examine our proposals based on the analysis of the network traffic behavior. The obtained results are based on two main parameters: the network latency (delay) and the traffic throughput.

The network model of each scenario contains 20 nodes, one PANC, four routers and fifteen end devices. They are populated over a logical mapping area. Node distribution of the static network is fixed (non-mobile), whereas the dynamic network supports the mobility model. In the dynamic model, the trajectory of each node is defined based on the average human walking speed, which is about 1.0 meter per second. In the dynamic model, each cluster represents one logical patient. Each cluster in the network will be allowed to move within the PANC transmission range, which is about 20 meters in our work. The clutter-tree network topology is defined based on the logical parent-child relationship.

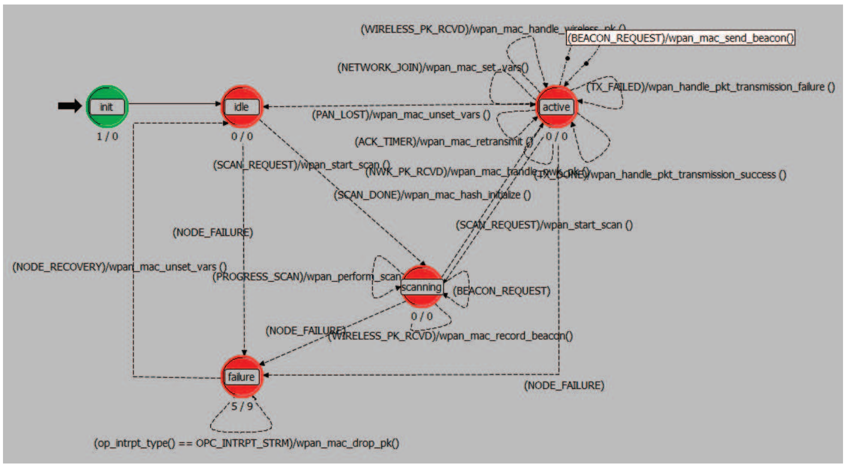


Figure 3.19: The MAC Layer over OPNET Modeler based on the IEEE 802.15.4

Parameter	Values
Size of Packet (bit)	1024
Min backoff exponent	3
Channel sensing duration (second)	dynamic
The power of packet reception threshold (Watt)	0.5
The type of traffic	adaptive
The protocol of routing	tree-based
The time of simulation (second)	3600

Table 3.1: Simulation parameters of the experimentation over OPNET Modeler

Figure 3.23 represents the end-to-end delay of all the packets received by the

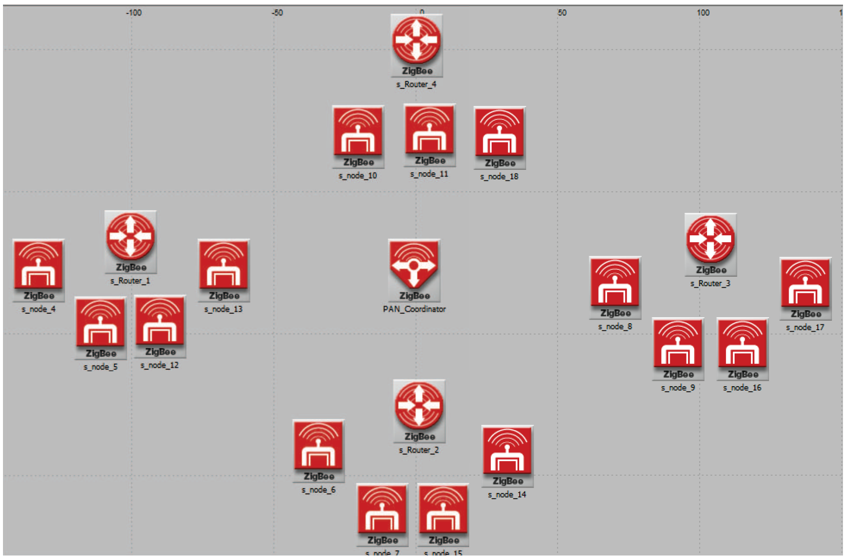


Figure 3.20: Cluster-tree topology of our experimentation over OPNET Modeler based on ZigBee/IEEE 802.15.4

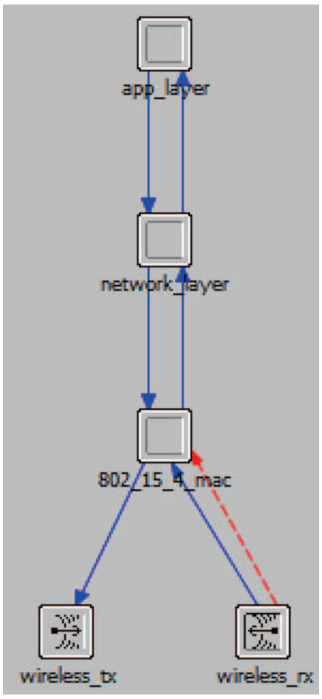
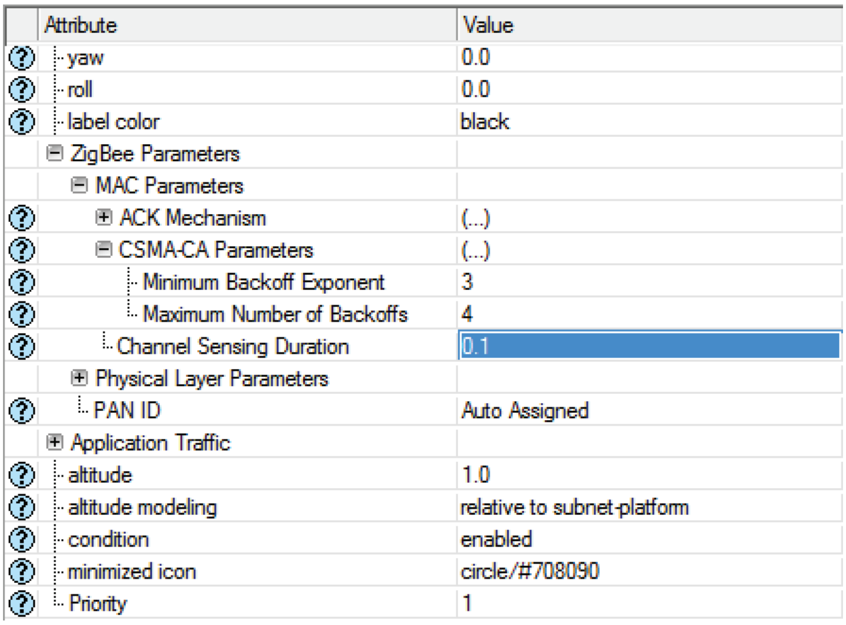


Figure 3.21: The node model over OPNET Modeler based on ZigBee/IEEE 802.15.4



Attribute	Value
yaw	0.0
roll	0.0
label color	black
ZigBee Parameters	
MAC Parameters	
ACK Mechanism	(...)
CSMA-CA Parameters	(...)
Minimum Backoff Exponent	3
Maximum Number of Backoffs	4
Channel Sensing Duration	0.1
Physical Layer Parameters	
PAN ID	Auto Assigned
Application Traffic	
altitude	1.0
altitude modeling	relative to subnet-platform
condition	enabled
minimized icon	circle/#708090
Priority	1

Figure 3.22: An example of the configuration profile of the priority model over OPNET Modeler based on ZigBee/IEEE 802.15.4

802.15.4 MACs of all WPAN nodes in the network and forwarded to the higher layer cite OPNET. It is obvious from the obtained results in Figure 3.23 that the average of the traffic delay in the dynamic scheduling is positively lower than the others by at least 0.03 seconds. The semi-dynamic scheduling also provides a high trend in front of the static scheduling. The line variation of the semi-dynamic model comes in the middle because of its scheduling combination algorithm, which is reflected in the graph behavior.

The results presented in Figure 3.24 shows the total number of bits (in bits/sec) forwarded from 802.15.4 MAC to higher layers in all the WPAN nodes of the network. The throughput in the dynamic scheduling model is improved by 5% over other models. The semi-dynamic scheduling model also shows an acceptable improvement by 2% as compared to the static model. It is noticeable that the trend of the dynamic model will be increasingly improved, whereas in the other models, the trend will be constant after a certain amount of time.

In the same context, Figure 3.25 provides the total traffic successfully received by the MAC from the physical layer in bits/sec, which indicates that the dynamic model generally optimizes the received traffic from the physical layer to the upper layers, i.e., the utilization of the network traffic is better used in the dynamic model than the others. By contrast, inflexibility of the static scheduling process usually leads to maximizing the amount of the overhead in every duty-cycle. In some cases, the network could stopped working when the traffic is heavy.

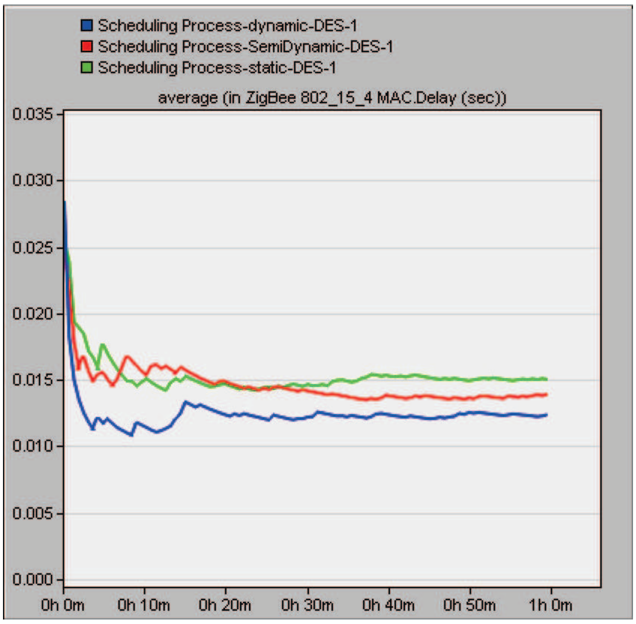


Figure 3.23: Comparison of the three types of scheduling based on the delay of all the packets received by the 802.15.4 MACs of all WPAN nodes in the network and forwarded to the higher layer

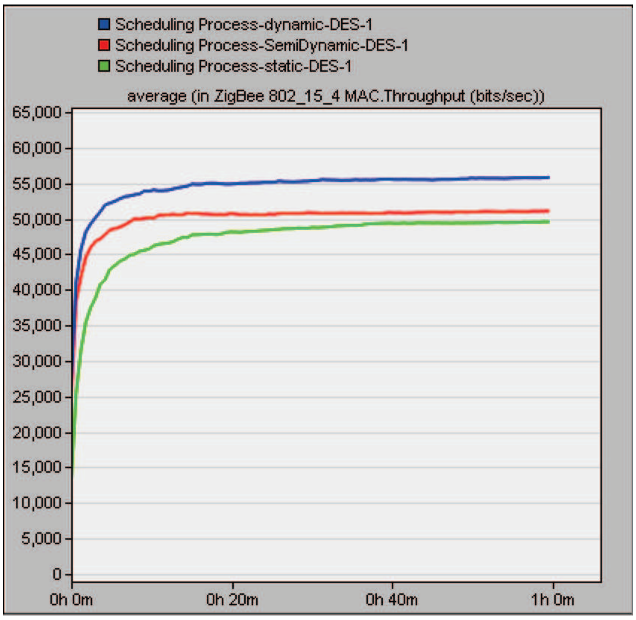


Figure 3.24: Comparison of the three types of scheduling based on the total number of bits (in bits/sec) forwarded from 802.15.4 MAC to higher layers in all WPAN nodes of the network

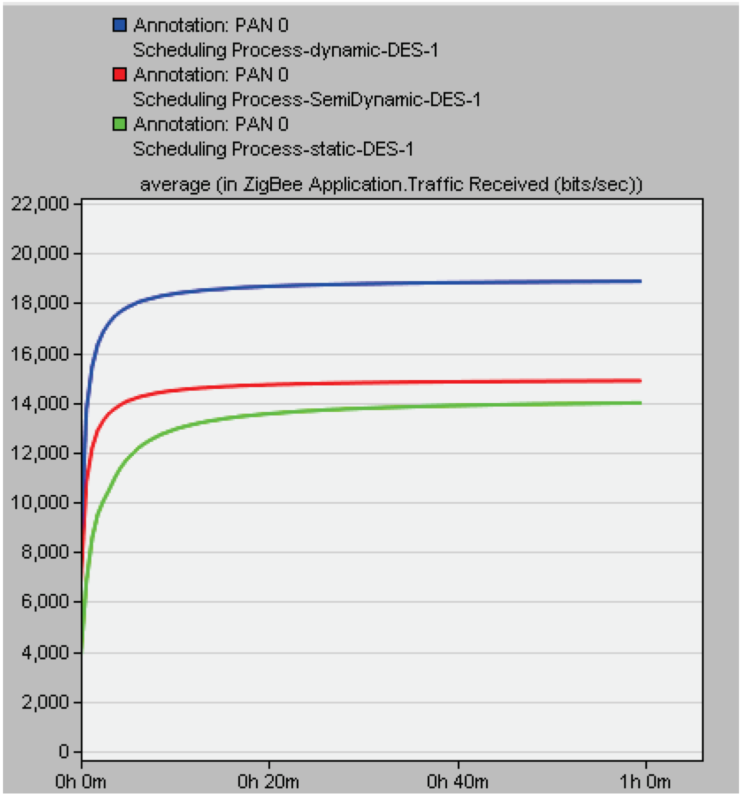


Figure 3.25: Comparison of the three types of scheduling based on the traffic received (packets/sec) in the application layer

Experimental Results

The idea, here, is to evaluate the performance of the network for each type of scheduling. Then, we will compare the three types to check the feasibility of our proposals. First, the overall performance of our network is evaluated and compared with the static scheduling approach. Secondly, the data packets (payload) for each type are calculated, as well as the overhead. The payloads are only the data packets, and the overhead packets are beacons, acknowledgments and commands. We compared the number of selective motes for furthering evaluation. Conducting detailed experimental tests are important to check the efficiency of the different types of scheduling.

In Figure 3.26, it can be seen from the graph that the total number of data packets in the dynamic scheduling is higher than in static scheduling. The next results show that the dynamic scheduling throughput usually can be higher than in static scheduling. The network behavior reflects the positive nature of our proposal, as it can be seen that at 1000 seconds, the dynamic scheduling process could send 12% more than in the static scheduling process. The high percentage of the dynamic scheduling process comes from the flexibility of assigning the network address, as well as the time offsets for each SD in the same network. Regarding the semi-dynamic scheduling process, it is clear that its performance usually comes in the middle between static and dynamic. The main reason for that is because of the "crossbreeding" between the static and dynamic approach.

In the same context, in Figure 3.27, we can see that the performance under dynamic scheduling always provides higher than static scheduling. It is noticeable that by comparing Figure 3.26 with Figure 3.27, they are always similar, due to the relationship between the number of packets and the total number of bytes. In fact, the data packet size is always the same.

By contrast, we have calculated other packets (the overhead) to measure the overhead in the network. Figure 3.28 shows the number of packets to measure the overhead of the network. The approach with the largest number of packets is static scheduling, with over 5,014 packets at 1000 seconds, whereas dynamic scheduling has the smallest number of the overhead packets, about 434 packets less than the static approach, which is a decrease of approximately 13.4%.

Figure 3.29 shows a comparison based on the total number of bytes (overhead). There was a significant increase in the percentage of the overhead in the static scheduling process. It reached a peak of 46% at 1000 seconds. The approach with the largest number of bytes is static scheduling, with over 454,795 bytes at 1000 seconds, whereas dynamic scheduling has the smallest number of unuseful bytes, about 107,059 bytes less than the static approach. There are some reasons this, mainly the commutation life cycle when any SD attempts to associate and disassociate or join and disjoin. Also, the network re-synchronization process wastes time and maximizes the overhead. The inflexibility of the static scheduling process usually leads to maximization of the amount of the overhead in every duty cycle. In some cases, the network could stop working when the traffic is heavy.

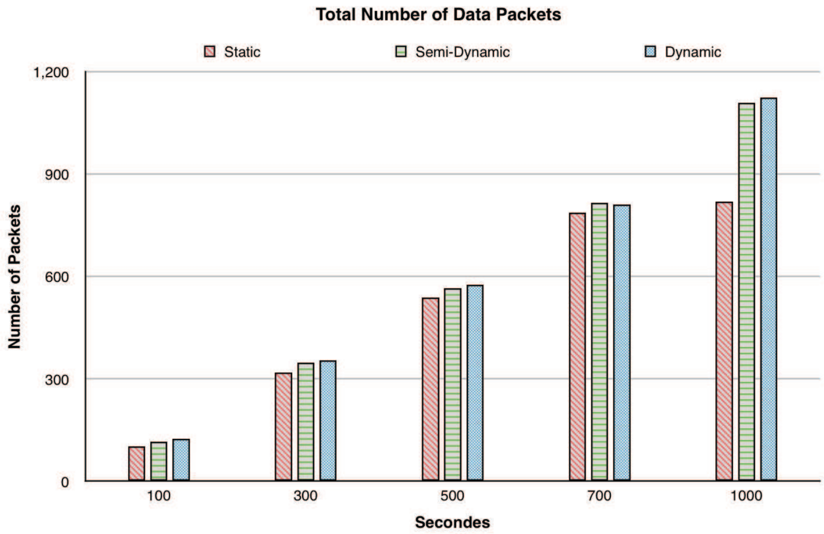


Figure 3.26: Comparison of the three types of scheduling based on Total Number of Data Packets

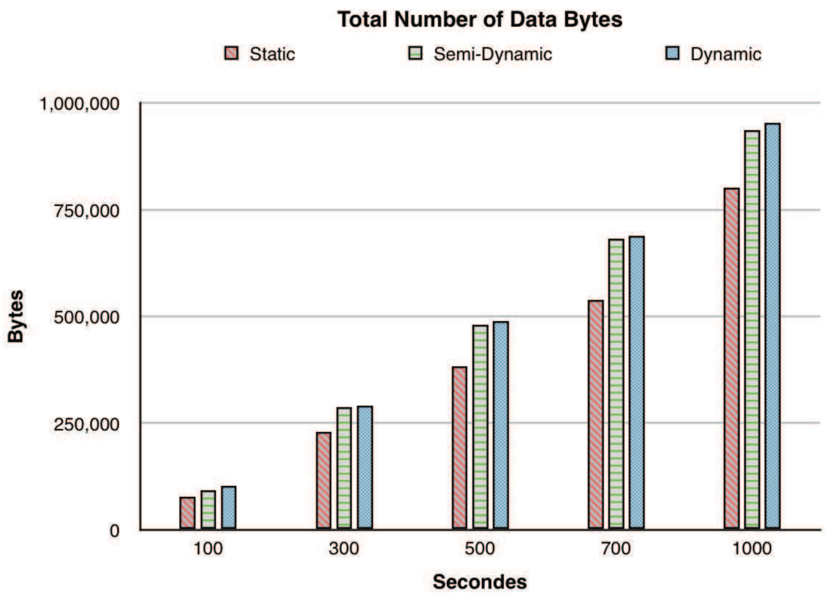


Figure 3.27: Comparison of the three types of scheduling based on Total Number of Bytes

The graph 3.30 depicts a comparison based on the average number of bytes per second for the data packets and the overhead. It is illustrated that there is a large gap between the static and our proposals. The static's average number of bytes/S at 300 seconds is lower by 13% than the other approaches, which shows the large number of unuseful bytes (bandwidth) that were transmitted. This illustrates the energy dissipation in the static scheduling process as compared to the semi-dynamic and dynamic scheduling approaches.

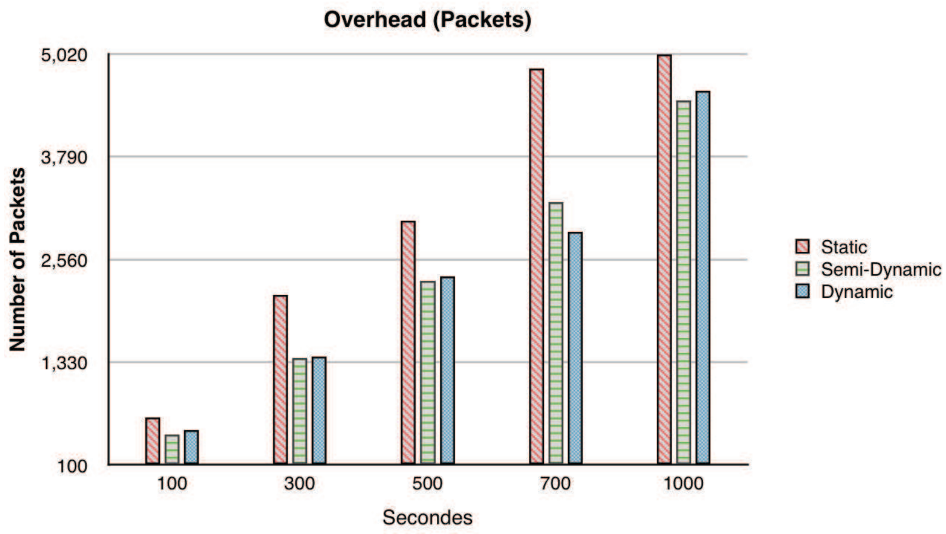


Figure 3.28: Comparison of the three types of scheduling based on the Packets Overhead

3.5 Conclusion

We have proposed a Semi-Dynamic Scheduling (SDS) process for Wireless Sensor Networks (WSN). We combined two types of scheduling in this study: static scheduling for coordinator nodes and dynamic scheduling for leaf nodes. This approach performs static generation of the source address assignment by the PAN coordinator to its children, which are accepted to be working as routers, and dynamic generation of the source address assignment to the end devices, each of which can have a time slot to send its data based on the traffic pattern. Then, we illustrate the feasibility of our proposal by showing the results obtained and summarized in the stated figures.The feasibility of our proposal is demonstrated using real implementation, and we have noticed that our proposal is able to decrease the latency ratio, increase throughput capacity, energy consumption in every ED and finally, provide efficient duty-cycle management.

We have also proposed an Efficient Dynamic Scheduling approach named EDS that is designed for Wireless Sensor Networks. This approach performs self genera-

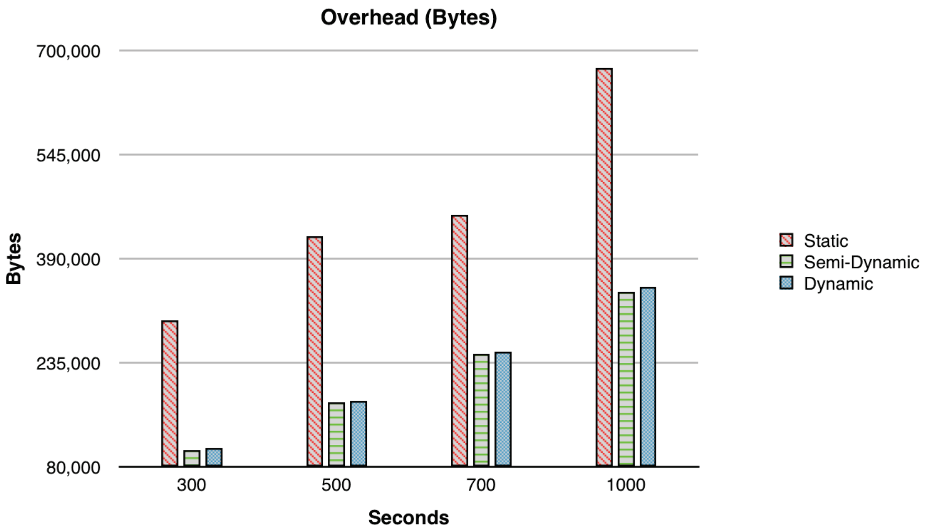


Figure 3.29: Comparison of the three types of scheduling based on the Bytes Overhead

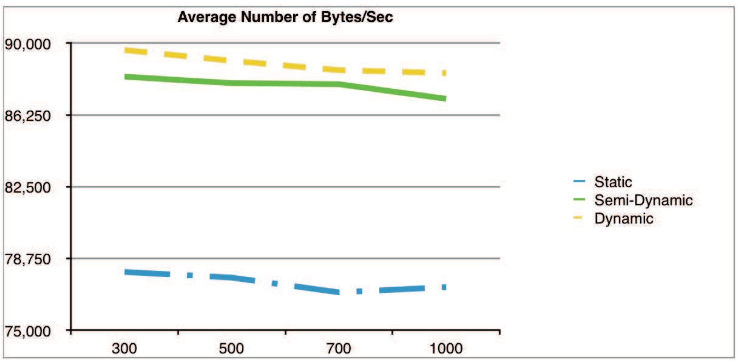


Figure 3.30: Comparison of the three types of scheduling based on the Average Number of Bytes/Second

tion of the source address assignment by the PAN coordinator to its children which are accepted to be working as routers, and each one of them can have a time slot to send its beacons based on the traffic pattern. For this purpose, we proposed an algorithm that will solve the shortcoming of the Static approaches. Then, we have illustrated the feasibility of our proposals by showing the obtained results that are summarized in in the stated figures. Moreover, we improve the current EDS proposal to handle different priority events to tackle the emergency cases in the WSNs as well as to overcome the main drawback in our current implementation: when the coordinator assigns many offsets to many nodes and the SuperFrame length becomes very high (n*15360 where n is the node number). However, we will overcome this drawback in the next improvement version by allowing assigning smaller transmission offsets to specific nodes as we have already started of the implementation code.

3.6 Résumé du Chapitre 3

Ce chapitre présente nos algorithmes d'ordonnancement dans les réseaux de capteur sans fils (Wireless Sensor Networks WSNs). Tout d'abord, nous avons décrit les hypothèses, les métriques ainsi que les objectifs de l'ordonnancement dans les WSNs. Par la suite, nous avons détaillé le principe de l'ordonnancement dans ces types de réseau qui utilise la topologie en arbre (Cluster-Tree) de Zigbee / IEEE 802.15.4. Ainsi, cette section expose aussi les fondements de la synchronisation des capteurs entre eux. Elle décrit notamment la structure de la "Superframe" utilisée dans Zigbee et envoyée par le PANC pour synchroniser les noeuds fils grâce à des beacons. Par la suite, nous allons présenter les deux algorithmes conçus pour améliorer les performance des réseaux WSNs.

3.6.1 Approche Semi-Dynamique d'Ordonnancement (SDS)

Cette approche consiste à séparer la topologie en arbre en deux niveaux :

- Niveau supérieur (PANC + routeurs) : ce niveau est considéré comme statique. En effet, les times slots de communication sont assignés statiquement suivant les adresses prédéfinis des différents noeuds capteurs.

- Niveau inférieur (End Devices EDs) : les time slots sont distribués au besoin et selon le trafic des données dans le réseau. Les EDs peuvent avoir des données à envoyer, ils seront considérés sinon il n y aura pas de temps perdu en leur associant des times slots qu'ils n'utiliseront pas.

Afin de valider la faisabilité de notre approche, nous avons implémenté cette approche sur des capteurs réels TelosB sous l'environnement TinyOS qui est un système d'exploitation adapté pour ce type de plateforme à faible énergie, mémoire et puissance de calcul.

Les résultats de cette implémentation sont présentés dans la figure 3.31.

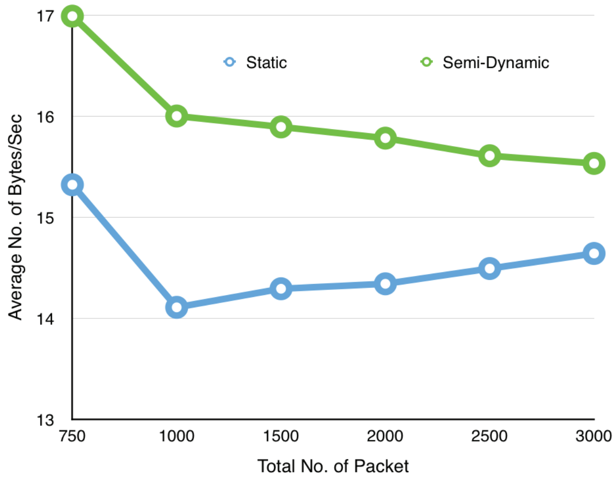


Figure 3.31: Moyenne du nombre d'Octets/Sec échangés

Dans la figure 3.31, nous avons six niveaux de valeurs pour l'indicateur de performance qui est le nombre moyen d'Octets / s par rapport au nombre total de paquets. Sur cette figure sont représentés les deux types d'ordonnancement (statique et semi-dynamique), il est clair que le semi-dynamique est plus efficace que le statique en terme de performances réseau. SDS permet d'augmenter la capacité et le débit et assure une gestion cyclique plus efficace.

3.6.2 Approche Dynamique Efficace d'Ordonnancement (EDS)

Les résultats prometteurs de SDS nous on poussé à rendre tout le réseau dynamique et adaptatif. Il n y a plus de séparation effective comme dans SDS. Dans EDS, tout les capteurs du réseau (PANC, routeurs et EDs) se voient assigner des adresses dynamiquement ainsi que les time slots.

Cette fois-ci nous avons eu recours à deus méthodes de validation de notre approche : la simulation et l'implémentation réelle.

Simulations

La simulation a été réalisé en utilisant le simulateur réseau OPNET. Nous avons pu ainsi simuler une topologie arbre contenant 20 noeuds. Un des résultats les plus significatif est la quantité de trafic de données reçue qui est représenté par la figure 3.32.

Cette figure démontre bien que EDS dépasse largement SDS et l'approche purement statique en terme de quantité de donnée échangée. Ceci s'explique par le fait que la gestion du temps est bien réalisée et que le médium est utilisé efficacement.

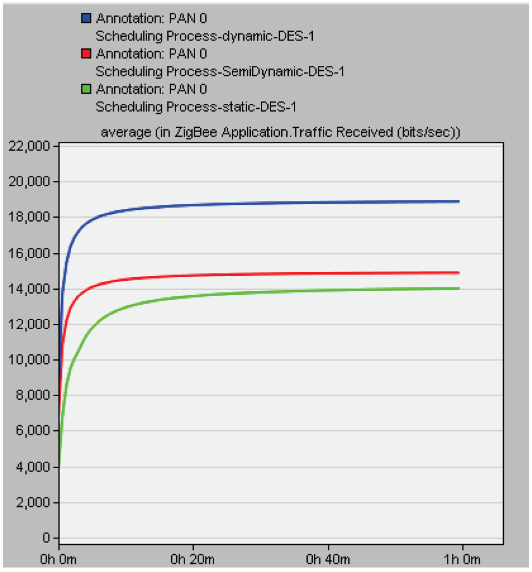


Figure 3.32: Comparaison des trois types d'ordonnancement basée sur le trafic reçu (bit/sec) dans la couche application

Expérimentations

L'expérimentation réelle a consisté à installer sur des capteurs réels, de marque TelosB qui tourne avec l'environnement TinyOS, les différentes approches d'ordonnancement. Nous avons ainsi pu comparer le comportement réel et en fonctionnement de ces algorithmes. Les résultats ont été visualisés à l'aide de notre outil développé appelé B-monitor (extension de l'outil open source Z-monitor).

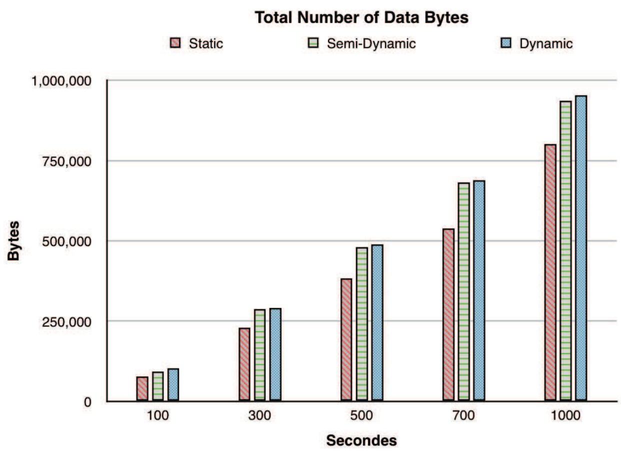


Figure 3.33: Comparaison des trois types d'ordonnancement basée sur le nombre total d'Octets reçus

La figure 3.33 représente le nombre total d'Octets reçus au fil du temps. EDS a de meilleures performances que SDS et TDBS dans l'ordre. Ceci est du à la réactivité de l'approche statique et de son adaptation au trafic réseau et au noeuds qui souhaitent transmettre dès qu'ils sont prêts pour éviter toute attente qui pénaliserait le débit et la latence.

Chapter 4

Task Scheduling Technique for Wireless Body Area Network

Contents

This study presents a priority based scheduling approach for tasks executed over Wireless Body Area Networks (WBANs). We consider tasks which have timing constraints for their execution and different priorities. They are usually repeated periodically. In this study we propose an algorithm which is able to provide a priority scheduling procedure for all tasks. In the actual version, the scheduling algorithm is executed over the network coordinator. This algorithm is implemented over TelosB nodes composing a Wireless Body Area Network.

The wireless sensor networks (WSN) has been considered as a reliable communication mechanism that can be used in many fields like healthcare, agriculture and military etc. In healthcare, a WSN can be used as a health-monitoring system by using special kind of sensor that called body sensor device (BSDs). The IEEE community agreed to create a healthcare WSN and they named it Wireless Body Area Network (WBAN). It is derived from the original WSN, but with more advance specifications that can help to monitor patients, animals etc. However, the WBAN-based platforms have been developed by many people around the world under academic or industrial projects. Most of those platforms are not yet being manufactured, because of the difficulties of customizing the WBAN-based software and hardware.

WBANs have a number of challenges such as power consumption, security, reliability, qualities of service (QoS). One of the most important challenges in WBANs is power consumption. However, time constraints play very important role in power consumption in WBANs. To avoid energy wasting (power), SNs need to be scheduled in a time division manner, in order to avoid collisions with each other and to be synchronized properly under the same network topology. Some static scheduling mechanisms pre-define the jobs of all SNs : each SN can do its job in a different time slot than others in the same network by defining a static time slots in order to avoid collisions with them. Then, it will repeat its scheduled job every duty-cycle continuously. But those mechanism might be not suitable for some applications typically for healthcare applications. For this reason, it is required to find more advance scheduling mechanism that can handle the WBAN-based platforms requirements.

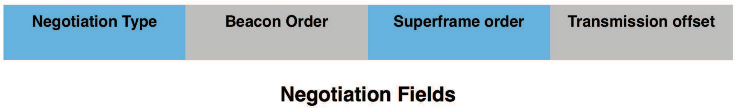


Figure 4.1: Negotiation Fields

1. **Negotiation type:** routers will ask the coordinator to have the eligibility to start sending beacons. This field will have the following values:
   - Number one, is for a negotiation of the priority request that handles the WBAN requirements.
   - Number two, is used for a negotiation of the normal request.
   - Number three, is used for accepting a node to be working as a router to start sending beacons to its children.
   - Number four, is used in case of refusing the node request to work as a router.
2. **Beacon Order:** the router's beacon order.
3. **Superframe Order:** the router's superframe order.
4. **Transmission Offset:** the transmission offset will be scheduled dynamically by the coordinator if the node is accepted to be working as a router.

The priority diagram shows the negotiation between the node and the coordinator, starting from the association step until the final step, which represent a successful negotiation. i.e. the node is eligible to work as router to start sending beacons. see Figure 4.2

This chapter is organized as follows, Section 2 presents related works about real-time systems based on WBANs. In Section 3, it shows preliminary definitions and assumptions required to detail our scheduling technique. Section 4 it shows our contribution, our algorithm in details. We also show the algorithm's implementation by using TelosB sensors. Section 5 presents concluding remarks and gives some ideas about our future works. The aim of this paper is to suggest a priority based

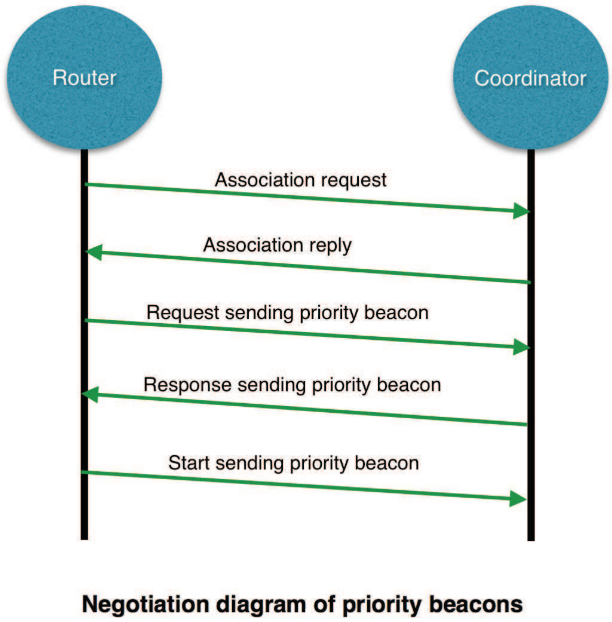


Figure 4.2: Negotiation Priority

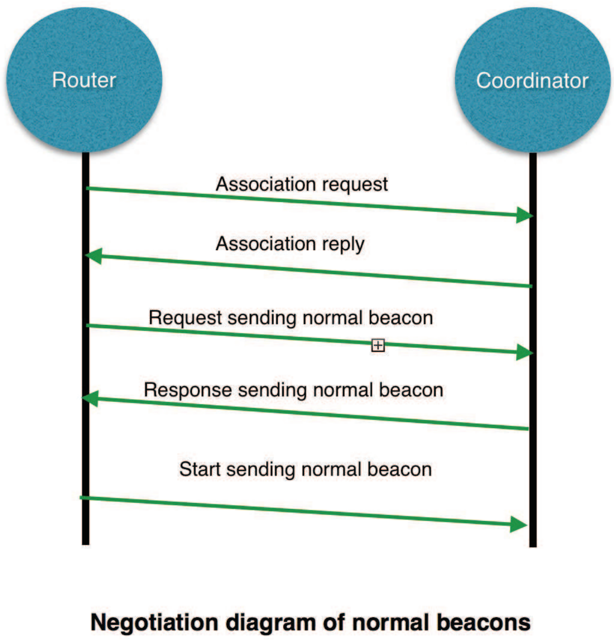


Figure 4.3: Negotiation Non-Priority

scheduling approach for WBANS. In our proposed solution, we have considered the timing and the priority constraints.

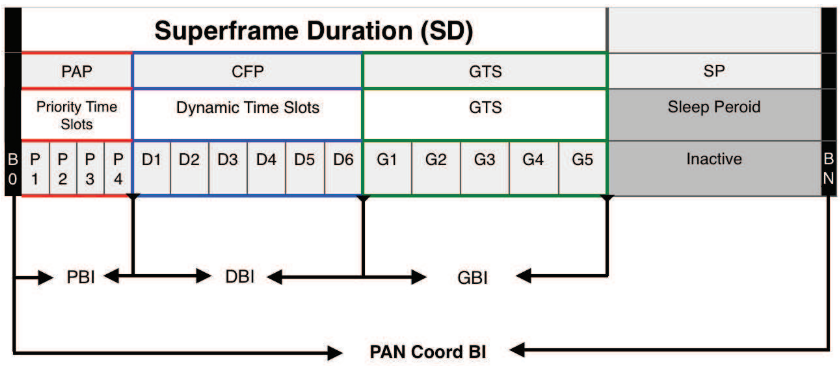
4.1 High Priority Scheduling Approach (HPS)

We propose a High Priority Scheduling Approach (HPS) for WBANs, which performs two kind of scheduling ; first one is to handle the priority traffics (abnormal events) e.g to report the heart beat cases, the second one for normal traffic, e.g. to monitor the usual vital signs. We consider the Time Division Beacon Scheduling by [2] as a key point of our proposal. We enhanced their work to meet the WBANs requirements. And, to adapt our proposal, there is a need to adapt the TDBS algorithm and the experimental implementation.

The coordinator will have a priority transmission offsets that will be distributed dynamically for all its associated nodes that are accepted to be working as routers.

4.1.1 Design

In this section, we add priority events to improve the dynamic algorithm. This extension will serve the corresponding nodes based on their priorities, i.e. The Priority-First-Serve (PFS) which is based on Shortest-Job-First (SJF), and the rest nodes will be served based on our dynamic algorithm which is the First-Ready-First-Serve (FRFS) In this case, every duty-cycle the PANC will have to synchronize its child nodes based on the prioritized control packets information, then the first ready nodes will be served just after. This feature need a minor modification in the superframe based on ZigBee/ IEEE 802.15.4 as presented in Figure 4.5.



$$BI = aBaseSuperframeDuration \times 2^{BO},$$
$$SD = aBaseSuperframeDuration \times 2^{SO},$$
$$Duty\ Cycle = \frac{SD}{BI} = 2^{SO-BO},$$

Figure 4.4: Superframe Duration structure in the Priority-based mode

The superframe that we have designed shows three active periods followed by one inactive period. First active duration for priority time slot allocation, which is for the received priority events. Second active duration for normal time slot allocation, which is for the received dynamic events. The last active duration is for the confirmed time slots, based on the ZigBee/IEEE802.15.4 standards this period can be used for up to seven devices simultaneously to support more priority events or even for receiving some priority devices to be guaranteed by the PANC.

We consider a number of priority coordinators, $PC\{ZRi = (SD_i, BI_i)\}, for\ 1 \leq i \leq P$ with superframes in different durations. And, also a number of normal coordinators, $NC\{ZRj = (SD_j, BI_j)\}, for\ 1 \leq j \leq N$ that have superframes in different durations.

Algorithm 4 Priority Scheduling Algorithm

$Scheduable = False;$ ▷ Initialization of scheduling result
$BI_{minor} = 8;$ ▷ Initialization of minor Beacon Interval
$BI = 4;$ ▷ Initialization of number of Beacon Intervals
$BI_{major} = BI * BI_{minor};$ ▷ Initialization of major Beacon Interval
$T_s[BI] = \{0, BI_{minor}, 2 * BI_{minor}, ..., (BI - 1) * BI_{minor}\};$ ▷ Initialization of First Available Consecutive Time Slots
$A \leftarrow \{BI_i = 2^{BO_i}\}, \ \forall\ i,\ 1 \leq i \leq P;$ ▷ The set of priority beacon intervals
$B \leftarrow \{BI_j = 2^{BO_j}\}, \ \forall\ j,\ 1 \leq j \leq N;$ ▷ The set of normal beacon intervals
$BI_{min} = 2^{BO_{min}};$ ▷ The minimum beacon interval
$\overline{A} \leftarrow SortBeaconInterval(A);$ ▷ Organizing in the increasing priority beacon order
$\overline{B} \leftarrow SortBeaconInterval(B);$ ▷ Organizing in the increasing normal beacon order

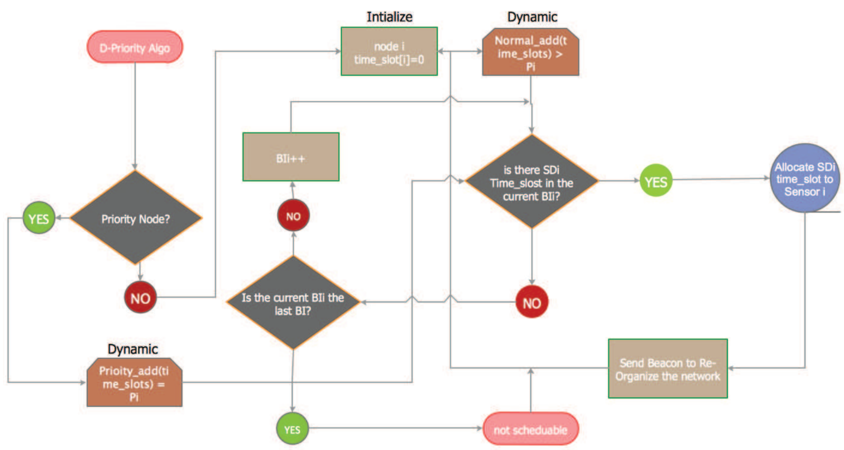


Figure 4.5: Priority Dynamic Algorithm

In the presented algorithm 4.5, first, we set: A $= \{BI_i = 2^{BO_i}\}, \ \forall\ i,\ 1 \leq i \leq P$. It represents the priority beacon intervals number for routers that are in the same transmission range.

B=$\{BI_j = 2^{BO_j}\}$, $\forall\ j,\ 1 \leq j \leq N$ It represents the priority beacon intervals number for routers that are in the same transmission range.

We initialize the algorithm by using two kind of intervals (minor and major). Let $BI_{minor} = 8$; it is the minimum beacon interval. And $BI_{major} = 4 * BI_{minor} = 32$; it is the maximum beacon interval.

As shown in the previous section, we apply the current scheduling approach on the exact example of the scheduling requirements previously that are presented in the two scheduling techniques. Thereafter, the arrangement of the beacon interval $\overline{A} = SortBeaconInterval(A)$ and $\overline{B} = SortBeaconInterval(B)$ will be maintained respectively starting by the priority events followed by the normal events.

The definition of the time slots set to $T_s[4] = \{0, 8, 16, 24\}$. Usually, the priority events will be allocated at the first time slots based on the modified superframe. The priority superframe duration (SD) is managed by the PANC. The network is maintained by using the hand-shaking process of the control messages between the PANC and its child routers. Hence, the PANC can learn and maintain its network behavior by exchanging the control messages with its child routers. In turn, each router will have to forward to/from its end devices and the PANC.

The non-prioritized time slots will be allocated just after serving the priority nodes based on the FRFS algorithm.

Example of scheduling:

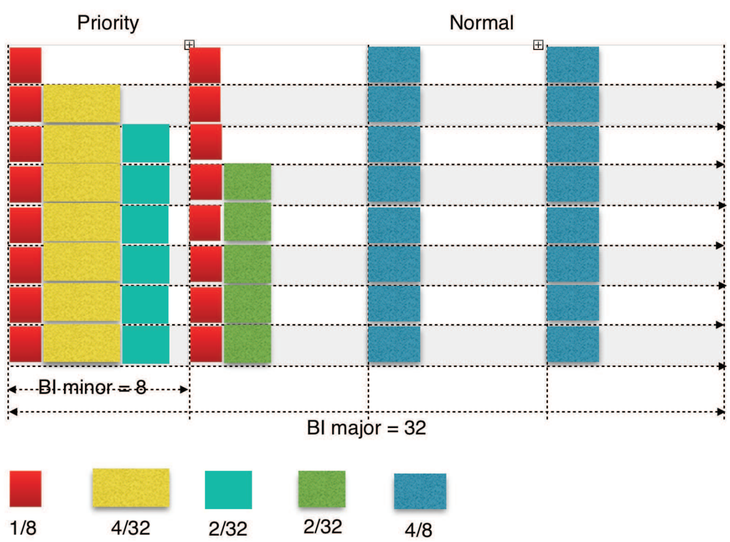


Figure 4.6: Example of the Priority Scheduling Process

4.1.2 Implementation and Performances Analysis

Our implementation is done by modifying and improving the open source code of TDBS NesC code that is available online [http://www.open-zb.net] under TinyOS and Ubuntu OS. The application used for writing the code is Eclipse with TinyOS adds-on. We used TelosB sensors. We installed the "tkn154 Packet Sniffer" on one of our TelosB sensors to measure the network activity, we used Z-Monitor "A Monitoring Tool for IEEE 802.15.4 Wireless Personal Area Networks" available online [http://www.z-monitor.org] to monitor the network performance in order to evaluate and compare our proposal with the TDBS one.

Simulation Results

We used the same simulation parameters that are defined 3.4.2.

The following Figures 4.7 and 4.8 represent the traffic evaluation for prioritized traffic. The result presented In Figure 4.9 represent the performance evaluation of the priority objects (nodes) compared with non-prioritized objects. As seen from the presented graph that the throughput trend in the priority node number five will be increased rapidly. Also, the delay in node five is very low compared with the non-priority node in the dynamic mode only, which indicate that the fast delivery of every priority message that will be sent from priority node.

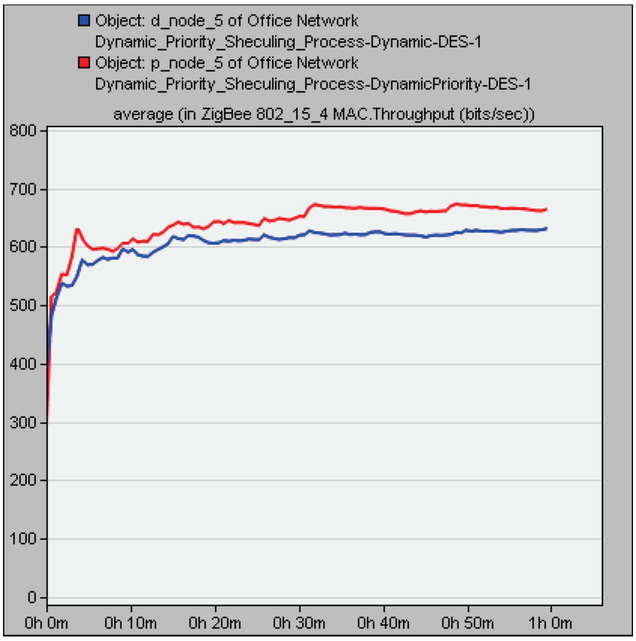


Figure 4.7: The throughput of the priority node number 5 compared with the dynamic node number 5

In the same context, the obtained results presented in Figure 4.9 shows a general performance of the prioritized traffic. These results translate the accumulative traffic by all priority nodes in the priority scheduling model. The priority scheduling model improves the network performance in terms of the total traffic successfully received/sent thought the network. As seen, the enhancement of the throughput

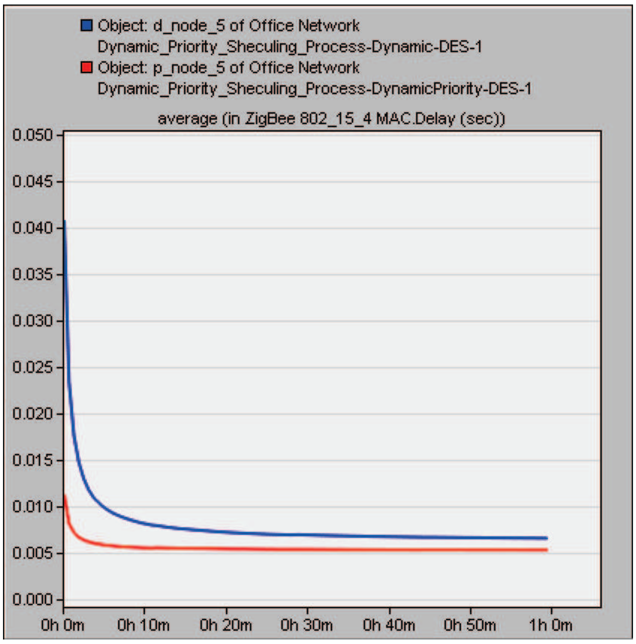


Figure 4.8: The delay of the priority node number 5 compared with the dynamic node number 5

in the prioritized traffic is effectively increased from 55,000 bit/sec dynamically to be over 60,000 bit/sec. In latency measurement in Figure 4.10 also shows big gape between the priority model and the pure dynamic model i.e. the priority is lower delay than the dynamic model.

Experimental Results

We conducted various experimental levels to compare and evaluate the performance of both HPS and TDBS. We used 16 TelosB sensors as follows; one sniffer, one coordinator, 5 routers and 9 end devices.We assumed that the requests for transmission address is received by coordinator in level orders and in each level from left to right nodes in the cluster tree. The idea is the coordinator assign two kind of time slots; priority and normal based on their requirements.

The details of the procedure used for the experimentation are described deeply in the Appendix A.

Figure 4.11 shows a comparison between dynamic approach and dynamic priority-based approach, each approach is demonstrated by comparing four EDs. In the static the four EDs has the same priority, addresses and time offsets. In the dynamic priority-based approach, four end-devices with different priorities i.e. two of EDs (ED2, ED4) have been chosen to be higher priority than the others (ED1,ED3). The priority device can send its packets whenever has sensory data to be sent without needing to wait until the next duty cycle. On the same context, the end-devices

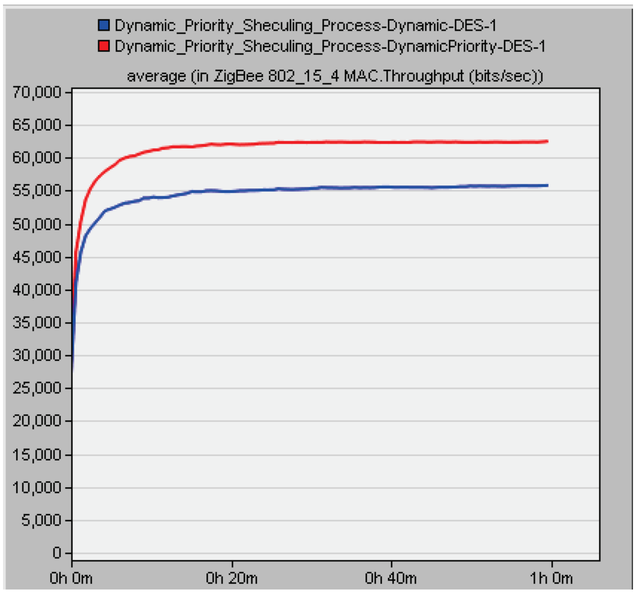


Figure 4.9: The average throughput in the priority network compared with dynamic network

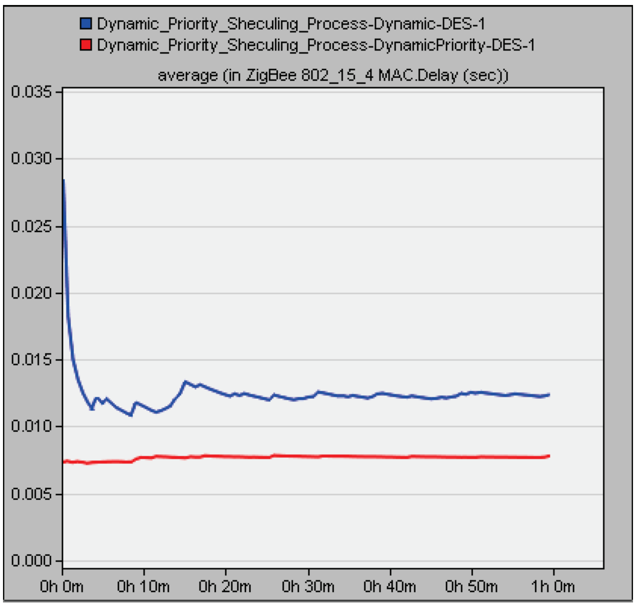
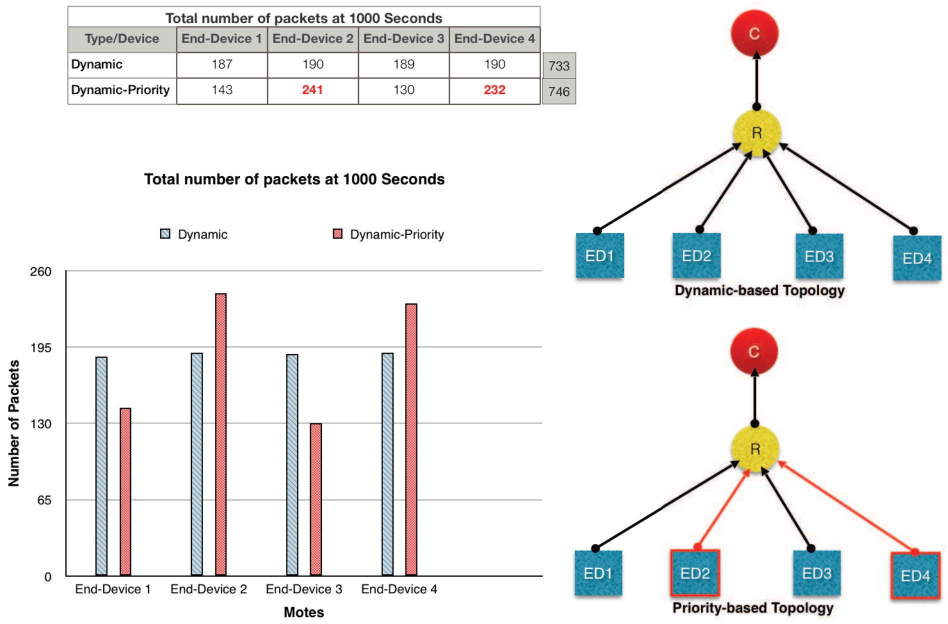


Figure 4.10: The average delay in the priority network compared with dynamic network

in the dynamic approach has nearly the same number of packets, whereas in the dynamic approach, each device can send its packets based on its priority which leads to be different in their total number of packets, i.e. the highest number of packets for the hight priority devices. The priority process is illustrated in the graph 4.11 , the performance of the priority devices are increased by 10.9% when comparing them by the low priority devices.



Total number of packets at 1000 Seconds

Type/Device	End-Device 1	End-Device 2	End-Device 3	End-Device 4	
Dynamic	187	190	189	190	733
Dynamic-Priority	143	241	130	232	746

Figure 4.11: Comparison between dynamic priority-based approach and Dynamic-based approach based on total number of packets

4.2 Conclusion

We proposed a High Priority Scheduling approach for Wireless Body Area Networks named HPS that designed for the Wireless Body Area Networks. This approach can handle both normal and prioritized traffics. For this purpose, we created an algorithm that will solve the shortcoming of the TDBS to meet the WBAN requirements. Then, we illustrate the feasibility of our proposal by showing the results obtained and summarized in the stated figures. Moreover, this approach is an improvement of our previous version to handle high priority events to tackle the emergency cases in the WBAN platform. Besides, it allows to overcome the main drawbacks of the first implementation that is when the coordinator assigns many offsets to many nodes and the super frame length become very high (n*15360 where n is the node number).

4.3 Résumé du Chapitre 4

Ce chapitre présente l'extension de l'approche dynamique EDS et une adaptation pour les réseaux WBANs. En effet, High Priority Scheduling (HPS) enrichi EDS par la prise en compte de la priorité de l'envoi des sonnées qui est une notion très importante dans les WBANs. En effet, les envois se font à travers une demande de transmission au PANC. Ce dernier dispose de deux files d'attente, une urgente et l'autre normale. Aucun paquet ne sera transmis de la file d'attente normale tant qu'il reste des paquets dans la file d'attente d'urgence.

De même que pour EDS, HPS a été testé et validé en simulation et sur des vrais capteurs.

4.3.1 Simulations

La simulation est réalisée à l'aide d'OPNET. La figure 4.12 présente le débit moyen dans un noeud prioritaire dans HPS et le compare par rapport à ce même noeud dans EDS. Puisque le trafic dans ces noeuds est prioritaire, il sera pris en charge rapidement. Ceci explique cette différence entre les performances dans HPS et EDS. Cette différence est inversée si le noeud n'est pas prioritaire.

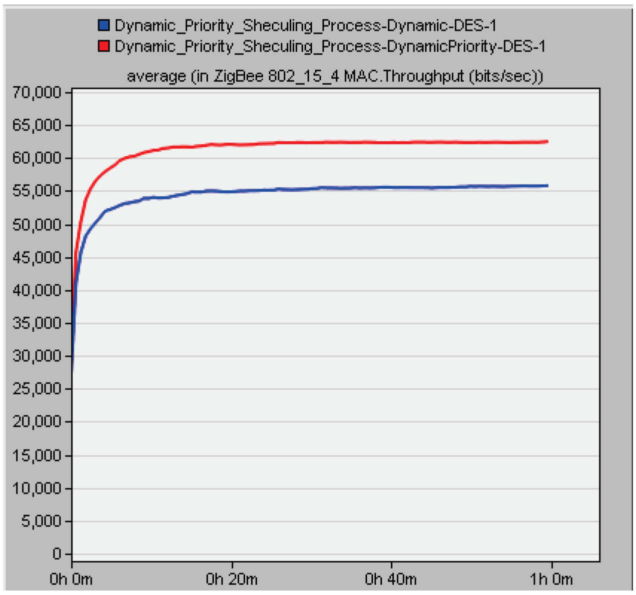
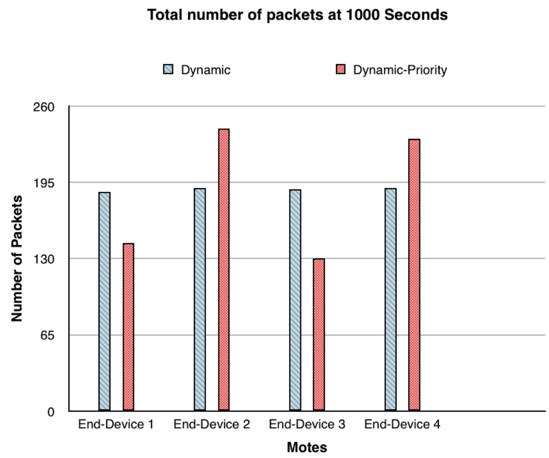


Figure 4.12: Le débit moyen dans HPS comparé avec EDS

4.3.2 Expérimentations

L'expérimentation a été réalisée sur des capteurs TelosB. Le réseau est composé d'un coordinateur PANC, d'un capteur espion relié à un PC pour surveiller le réseau, 5 routeurs et 9 EDs. Les résultats obtenus sont résumés dans la figure 4.13.

Total number of packets at 1000 Seconds					
Type/Device	End-Device 1	End-Device 2	End-Device 3	End-Device 4	
Dynamic	187	190	189	190	733
Dynamic-Priority	143	241	130	232	746



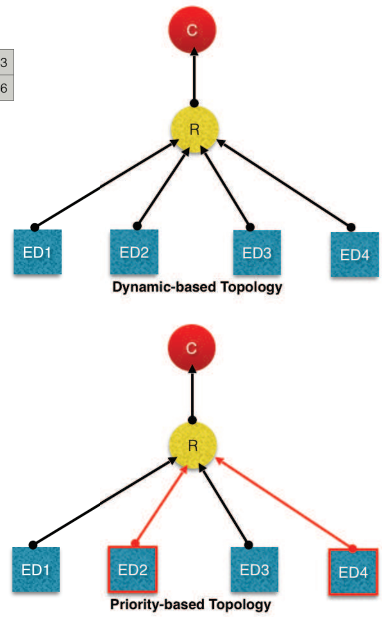


Figure 4.13: Comparaison entre les approches dynamique et celle basée sur la priorité suivant le nombre total de paquets

La figure 4.13 montre une comparaison entre l'approche dynamique (EDS) et l'approche dynamique basée sur la priorité (HPS). Dans l'approche dynamique EDS, les quatre EDs possèdent la même priorité. Dans l'approche dynamique basée sur la priorité HPS, quatre EDs possèdent des priorités différentes, soit deux des EDS (ED2, ED4) ont été choisis pour avoir une priorité plus élevée que les autres (ED1, ED3). Les capteurs prioritaires peuvent envoyer leurs paquets chaque fois que les données sont disponibles sans avoir besoin d'attendre le prochain cycle de travail. Dans le même contexte, les EDs dans l'approche dynamique ont presque tous le même nombre de paquets envoyés, alors que dans l'approche dynamique, chaque appareil peut envoyer ses paquets en fonction de sa priorité, ce qui conduit à des différences notables dans leur nombre total de paquets envoyés, c-à-d un nombre plus grand pour les dispositifs prioritaires. Les performances des dispositifs prioritaires sont augmentés de 10,9 % quand on les compare par rapport aux dispositifs de faible priorité.

Conclusion et Travaux Futurs (Version Française)

L'ordonnancement des tâches dans les réseaux de capteurs sont des questions difficiles et critiques. En fait, les algorithmes d'ordonnancement des tâches sont très importants dans toutes les applications. Précisément, c'est le cas dans les applications sensibles au latence telles que les applications médicale de santé. Tous les dispositifs (noeuds) de capteurs sans fil qui sont interconnectés et ont besoin de communiquer les uns avec les autres afin d'échanger des données entre eux. Ces dispositifs ont besoin d'un algorithme intelligent pour formaliser et organiser leurs communications. La nécessité de l'organisation des tâches des nœuds provient des exigences en matière de communication temps réel.

En outre, la planification des tâches des réseaux de capteurs sont en pleine émergence. La raison de cela est dû aux normes des applications sensibles au temps de haute qualité. Ces exigences sont principalement issues de ces applications qui ont pour objectif de remplacer les réseaux de confiance en cas d'urgence ou en cas de graves problèmes de santé, par exemple. De plus, ces réseaux doivent supporter efficacement le modèle de trafic réseau. La mobilité dans ces réseaux doit être considérée dans ces types d'applications. La prise en charge de la mobilité du réseau est une tâche complexe, parce que le corps humain se déplace aléatoirement et chaque être humain peut se déplacer différemment des autres. Cela peut modifier la topologie du réseau en continu. Ces questions doivent être prises en compte lors de l'échange des données entre le corps humain ou dans de l'échange des données au sein même corps humain.

Travaux réalisés

Cette thèse s'est intéressée à la conception de algorithmes d'ordonnancement pour les réseaux de capteurs. Nous avons proposé trois différents algorithmes basés sur ZigBee / IEEE 802.15.4. Les protocoles et les normes de communication, parfois, ne possèdent pas une capacité suffisante pour coopérer avec le modèle de trafic. Pour cette raison, nous les avons conçu selon des fonctionnalités spécifiques. En fait, la communication entre les nœuds de capteurs ne sont pas une tâche facile. Cette communication doit être bien organisée pour échanger les données entre les dispositifs de réseau en mode sans collision. La communication de bout en bout doit également veiller à l'interopérabilité des nœuds de capteurs sur différentes parties du corps humain par exemple. Le traitement des algorithmes d'ordonnancement décrivent la façon d'échanger les données entre les dispositifs de réseau. De plus, ces algorithmes doivent supporter la fonctionnalité d'auto-organisation.

Nous avons développé nos propres architectures logicielles qui sont définis dans les couches ZigBee / de IEEE802.15.4 sous environnement TinyOS. Ils ont été élaborés sur la base des phases du cycle de vie de développement d'un système pour produire un travail de haute qualité. Les logiciels sont modélisés dans un certain nombre de processus. Ces processus sont intégrés au sein de la pile complète du système d'exploitation minimal TinyOS. Ces logiciels ont été validés à l'aide de deux principaux outils de vérification. La première est notre version développée pour l'outil analyseur de réseau. Cet outil espionne le trafic sur le réseau et le transpose dans l'interface utilisateur graphique sur un ordinateur personnel. Le deuxième outil est le simulateur réseau OPNET, en développant trois scénarios différents selon les algorithmes d'ordonnancement conçus. Les résultats ont validé la faisabilité de nos approches.

La mise en œuvre des différents modèles ont été présentés. Ces applications ont été développées en utilisant l'environnement TinyOS sous le système d'exploitation Ubuntu Linux. En outre, ils ont été modélisés sur OPNET dans trois scénarios différents. Ils ont été comparés les uns avec les autres et également avec un algorithme d'ordonnancement bien connu appelé TDBS (Time Division Beacon Scheduling). Les deux méthodes de validation de notre travail nous ont permis de démontrer la faisabilité de nos propositions. Ainsi, les résultats obtenus ont prouvé que les propositions sont efficace et évolutive.

Les WSNs en générale, en incluant WBANs, sont considérés comme des technologies très intéressantes indépendamment de leurs inconvénients et leurs lacunes. Néanmoins, ils ont quelques défis qui doivent être résolus tels que l'organisation efficace du trafic réseau, ainsi que la gestion du modèle de mobilité dans les applications critiques telles que les applications de santé. L'intégration de service de positionnement géographique (GPS) avec les applications WSN pourrait aider à détecter la position d'un patient particulier, par exemple. En effet, plusieurs protocoles peuvent être intégrés avec les applications WSN afin de parvenir à une des plateformes entièrement automatisées. Ces protocoles peuvent être facilement intégrés avec notre travail parce qu'ils sont élaborés sur la base d'un système open source.

Dans ce travail, nous avons présenté une étude approfondie des applications WSN. Nous avons également détaillé les exigences des algorithmes d'ordonnancement de tâches sur la couche ZigBee / IEEE 802.15.4. En plus, nous avons présenté les plateformes WBAN bien connus. Par la suite, ces plateformes ont été classées et évaluées en fonction de certains paramètres importants. Ce travail a examiné également le modèle de mobilité dans les applications de soins de santé en proposant une méthode de suivi d'une épidémie. Trois algorithmes différents d'ordonnancement de tâches ont été conçus, développés et évalués. La première proposition, appelée Semi-Dynamic Scheduling (SDS), est le fruit du mixe de deux types d'ordonnancement, statique et dynamique. La deuxième est l'approche Efficient Dynamic Scheduling (EDS). Elle est considéré comme le travail de base de cette étude. Cette approche peut faire face au changement du trafic réseau. La troisième approche est l'approche High Priority Scheduling (HPS). Elle a été conçue spécifiquement pour les WBANs qui permet prend en charge les événements critiques. Les résultats obtenus montrent que nos propositions sont plus efficaces que les approches entièrement statiques. En outre, l'approche dynamique est plus évolutive que la semi-dynamique. De plus, la version étendue de l'approche dynamique qui prend en charge les événements de haute priorité est plus efficace que l'approche entièrement dynamique en terme de prise en charge des situations critiques en cas d'urgence.

Formellement, la technique d'ordonnancement statique des tâches dans les réseaux de capteurs a été conçue séparément de la technique d'ordonnancement des tâches dynamique. Dans ce travail, nous avons tout d'abord mélangé les deux pour vérifier la faisabilité de notre étude. En effet, cette proposition hybride nous a permis d'aller un pas en avant en concevant une approche totalement dynamique et efficace. Les résultats expérimentaux de l'approche hybride (approche de d'ordonnancement semi-dynamique) démontrent les avantages de la transformer en une approche entièrement dynamique. Les attributions d'adresses doivent être reliées avec les tranches de temps disponibles. Les paquets de données seront transmis dans une relation parent-enfant. L'initiateur des données (noeud de capteur) doit être prévenu par le coordinateur. Chaque noeud de capteur peut avoir une priorité différente de celle des autres. Ce qui signifie que, chaque noeud de capteur qui se trouve dans un réseau ne dispose pas toujours de données à envoyer. Les intervalles de temps (time slots) seront affectés aux nœuds prêts à envoyer uniquement. Cette approche réduit le temps dû à la prise en compte de noeuds qui ne sont pas prêts à envoyer ou en attente. Ainsi, il n'y a pas besoin d'attendre une durée de temps supplémentaire pour rien. Elle permet également de maximiser le taux de livraison des paquets de données (PDR), car elle évite les surcoûts en minimisant les paquets inutiles tels que les paquets de contrôle. Le débit sera également amélioré en raison du chargement de la gestion de chaque cycle dans le réseau. En plus, la performance du réseau est bien gérée et est améliorée en général.

En résumé, la première approche hybride (approche d'ordonnancement semi-dynamique SDS) divise le même réseau en deux niveaux. Le niveau supérieur de l'arbre, qui sont le PANC et ses routeurs des noeuds fils qui sont considérés comme statiques. Le niveau inférieur, qui est constitué de nœuds à l'extrémité du réseau (appareils finaux ou End Devices EDs) sont considérés comme le niveau dynamique. La communication entre les deux niveaux est géré par le PANC. Dans le niveau supérieur, les routeurs sont fixés et ils sont pré-définies par des intervalles de temps et d'adresses. Les EDs du niveau inférieur peuvent se déplacer et changer leurs routeurs (pères) et ils se verront attribués des intervalles de temps et d'adresses sur la base du Premier-Prêt-Premier-Envoyé. Le processus de transmission se fait selon le modèle de trafic. Dans la seconde approche qui est l'approche Efficient Dynamic Scheduling (EDS), l'ensemble des dispositifs dans le même réseau sera géré de la même manière que le niveau inférieur dans l'approche hybride. Le processus d'échange de paquets sera géré par le coordinateur (PANC) selon l'état en instantané de chaque dispositif dans son réseau. L'approche à haute priorité (HPS) est considéré comme une approche dynamique qui tient compte des cas d'urgence. Les événements prioritaires seront envoyés en premier (Premier-Noeud-Prioritaire-Prêt–Premier-Envoyé). Tant qu'il y a des noeuds qui sont prêts à émettre, il regarde d'abord s'il y a un noeud prioritaire qui est prêt à émettre dans la file d'attente avant de les servir.

Travaux futurs

Dans ce travail, nous avons évalué nos propositions par l'obtention de certains résultats qui reflètent l'importance de l'amélioration de cette recherche. Les trois approches pourraient être améliorées. Elles pourront ainsi être appliqués sur une zone large (environnement à large échelle) avec beaucoup plus de dispositifs. Chaque approche pourrait être utilisée à des fins différentes. Ils ont la capacité à être intégré avec certaines autres technologies telles que (4G, GPS, VANET, etc.). De plus, ils pourraient être améliorés pour être plus efficaces en ajoutant plus de fonctionnalités. Ainsi, ils seront plus faciles à configurer et à utiliser.

Dans la première proposition d'approche d'ordonnancement semi-dynamique (SDS), il y a quelques améliorations que nous comptons appliquer. Tout d'abord, la distance entre les deux niveaux peut être étendue en intégrant un des protocoles ou normes telles que 4G ou VANET. Cela permettra à chaque groupe de travailler à distance. Ensuite, l'intervalle de temps entre deux dispositifs dans différents niveaux pourra être optimisé en contrôlant l'overhead. En plus, des messages de contrôle (acquittements) peuvent être transmis seulement pour les prêt à envoyer.

L'approche Efficient Dynamic Scheduling (EDS) pourra également être améliorée en termes de processus de ré-ordonnancement. Ce dernier induit une perte temps et consomme de la mémoire au sein du coordinateur du réseau car les affectations d'adresses demandent un calcul complexe. Toujours en termes de modèle de mobilité, cette approche peut être amélioré pour être utilisé partout indépendamment de son emplacement. Par exemple dans la demande de soins de santé, ce modèle peut surveiller un patient particulier à distance et n'importe ou. Ce patient peut faire son activité quotidienne sans aucune barrière. Des extensions à ce service peuvent être intégrés afin d'étendre le modèle de surveillance comme avec l'utilisation de réseau 4G et du GPS pour détecter la position de chaque patient dans le cas de situations d'urgence. En outre, dans l'ambulance les réseaux VANET pourraient être utilisés pour interagir avec le réseau WBAN ainsi qu'avec les systèmes de communication à bord de véhicules.

De la même manière avec l'approche High Priority Scheduling (HPS), le modèle de la mobilité peut être intégré facilement. En terme de support de plusieurs dispositifs prioritaires, un autre protocole peut être utilisé. Cela sera étudié dans le futur afin de trouver les protocoles appropriés. Ajouté à cela, il faudrait prévoir un mécanisme qui permettrait de détecter à l'avance les problèmes de santé critiques chez un patient. Pour cela, un modèle de prédiction peut être intégré aussi bien. Ainsi, une étude des modèles de prédiction sera réalisée. Basé sur cette étude, l'un d'eux pourra être intégré dans le processus d'ordonnancement à haute priorité.

Finalement, tout le travail prévu sera évalué par la mise en place d'expérimentations réelle à large échelle. La mise en œuvre sera axée sur le domaine de la santé. Un certain nombre de scénarios sont prévus pour les trois propositions avec les futures améliorations. Le premier scénario sera mis en œuvre dans un hôpital pour certain nombre de patients en mettant l'accent sur le niveau inférieur de l'approche d'ordonnancement semi-dynamique (SDS). Le niveau supérieur est toujours fixe. Les membres du corps humain seront équipés de dispositifs de niveau inférieur (EDs). Les routeurs transmettent les données au PANC. Les EDs échantillonnent les données médicales sur la base du premier noeud prêt est le premier à envoyer. Chaque patient dispose d'un module de localisation, il ou elle peut seulement se déplacer dans l'environnement hospitalier. Le but de ce scénario est de détecter tout problème de santé lié à la localisation d'un patient à l'hôpital. Le scénario suivant sera réalisé sur la même base, mais avec plus de liberté. La mise en œuvre se fera au domicile des patients. Tout patient particulier peut effectuer son activité quotidienne à la maison sans aucune barrière. Pour étendre cette distance, chaque patient sera équipé d'un assistant numérique personnel (PDA) avec un module GPS et une connexion 4G. Ainsi, il ou elle peut aussi se déplacer autour de son domicile. Enfin, le même scénario sera aussi évalué pour la prise en compte de la priorité, ainsi des problèmes de santé plus précises seront vérifiées afin de valider le travail sur un environnement à large échelle.

Conclusion and Future Work (English Version)

Task scheduling of WSNs are challenging and critical issues. In fact, task scheduling algorithms are very important in all applications. Precisely, it is the case in the time sensitive applications such as healthcare applications. All sensor nodes (SNs) that are interconnected wirelessly need to communicate with each other in order to exchange data between them. These devices need a smart algorithm to formalize and organize their communications. The necessity of the nodes' tasks organization comes from the real-time communication requirements.

In addition, task scheduling of WSNs are emerging. The reason for that is due to the high quality standards of the time sensitive applications. These requirements are mainly coming from these applications to replace trustful networks in emergency cases of critical health problems for instance. Furthermore, these networks have to support the traffic pattern. Also, the mobility in such networks have to be considered in these applications. It is very challenging issue to support the mobility of the network, because human body moves randomly and each human limb can move differently than the others for example. This can change the network topology continuously. These issues must be managed when exchanging data between human bodies or within the same human body.

Outlook of the Work

This thesis is interested in designing scheduling algorithms for WSNs. We have designed three different algorithms based on ZigBee/IEEE802.15.4. The communication protocols and standards sometimes do not have a sufficient ability to cooperate with the traffic pattern. For this reason, we have designed them according to specific functions. In fact, the communication between SNs are not an easy job. This communication must be organized properly to exchange data between the network devices in collision-free mode. The end-to-end communication also needs to ensure the interoperability of the SNs on different part of the human body for example. The processing of the organizational algorithms describe the way of exchanging the data between the network devices. Additionally, the organizational algorithms (scheduling algorithms) have to support self-organizing feature.

Furthermore, we have developed our own software architectures which are defined within the ZigBee/IEEE802.15.4 layers under TinyOS environment. They have been developed based on the System Development Life Cycle (SDLC) phases to produce a high quality work. The softwares are modeled in a number of processes. These processes are integrated within the full stack of the TinyOS framework. These softwares have been validated by using two main verification tools. First one, is our tool's version of the network sniffer. This tool sniffs the network traffic and reflects it into the graphical user interface (GUI) on a personal computer. Second tool is the OPNET modeler by developing three different scenarios according to the designed algorithms. The results have validated the feasibility of our approaches.

The implementation of the different models have been presented. These applications were developed over TinyOS environment under Ubuntu Linux operating system. Also, they are modeled over OPNET in three different scenarios. They have been compared with each others and also with a well-known scheduling algorithm called TDBS. The two different ways of validating our work helped us to prove the feasibility of our proposals. Thus, the obtained results have proved that the proposals are efficient and scalable.

WSNs in general including WBANs are still the most interesting technology regardless of their drawbacks and shortcomings. They have some challenges that should be solved such as organizing the network traffic efficiently. Furthermore, the management of the mobility model in the critical applications such as healthcare applications. Integrating graphical positioning service (GPS) with the WSN applications could help to detect the position of a particular patient for example. Additionally, more protocols can be integrated with WSN applications to achieve a fully automation platforms. These protocols can be easily integrated with our work, because they are developed based on an open source system.

In this work study, we present a comprehensive study of the WSNs applications. We also explained and detailed the requirements of the task scheduling algorithms baed on the ZigBee/IEEE 802.15.4. Moreover, we present a well-known WBAN platforms. Thereafter, these platforms are classified and evaluated based on some important parameters. This work also investigates the mobility model in healthcare applications by proposing epidemic tracking method. Three different task scheduling algorithms have been designed, developed and evaluated. First proposal, Semi-Dynamic Scheduling (SDS) approach, which is mixed of two kinds of scheduling, static and dynamic. Second one is the Efficient Dynamic Scheduling (EDS) approach. It is considered as the core work of this study. This approach can cope with the traffic pattern adaptively. The third approach is the High Priority Scheduling (HPS) approach. It is designed specifically for the WBANs which enables the mobility model and supports the critical events. The obtained results show that our proposals are more efficient than the fully static approaches. Furthermore, the dynamic approach is more scalable than the the semi-dynamic one. Also, the extended version of the dynamic approach which supports the high priority events will be more efficient than the fully dynamic approach in terms of supporting the critical situations in emergency cases.

Formally, static task scheduling technique of WSNs is designed separately from the dynamical task scheduling technique. In this work, we firstly mixed them to check out the feasibility of our study direction. In addition, this hybrid proposal allows us to go step forward by designing an efficient dynamic approach. The experimental results of the hybrid approach (Semi-Dynamic scheduling approach) show that the benefits of transforming it into a fully dynamic approach. The address assignments must be linked with the time slots. Data packets will be transmitted (forwarded) in parent-child relationship. The data initiator (sensor node) has to be scheduled by the coordinator. Each SN could have different priority than the others. Meaning that, every SN that is located under one network not always have data to send. Time slots will be assigned to the ready nodes to send only. This approach reduces waiting time that were caused by not ready node to send. Hence, no need to wait extra time duration for nothing. It also maximizes the data packet delivery rate (PDR), because it avoids overheads by minimizing the useless packets such as control packets. Throughput also will be increased due to load management of every duty-cycle in the network. In addition, the performance of the network is well-managed and is improved generally.

In summary, the first hybrid approach (Semi-Dynamic scheduling approach) divides the same network into two levels. The upper level of the cluster-tree, which are the PANC and its child routers are considered as a static level. The lower level, which are the leaf nodes (end-devices) (EDs) are considered as dynamic level. The communication between the two levels is maintained by the PANC. In upper level the position of the routers are fixed and they are pre-defined by time slots and addresses. When, the EDs of the lower level could move around their Routers and will be assigned by time slots and addresses based on the First-Ready-First-Send algorithm. The transmission process is done based on the traffic pattern. The second approach which is the Efficient Dynamic Scheduling (EDS) approach, the whole devices in the same network will be managed on the same way than the lower level in the hybrid approach. The packet exchanging process will be managed by the decision maker (PANC) according to the instant situation of each device in its network. In the high-priority approach, it is considered as a dynamic approach with taking into account the emergency cases. Priority events will be sent first (First-Ready-Priority-Node-First-Send). Eventhough, there are some Ready-Nodes-to-Send, it looks first if there are a First-Ready-Priority-Nodes-to-Send in the queue to serve them before.

Future Work

In this work, we have evaluated our proposals by obtaining some results, which reflect the importance of improving this research. The three approaches should be improved. They have to be applied on a wide area (large-scale environment) with much more devices. Each approach could be used for different purposes. They have the ability to be integrated with some others technologies such as (4G, GPS, VANET, etc.). Also, they could be enhanced in terms of their functionality to be more efficient by adding more features. Thus, they will be easier to configure and to use.

In the first proposal Semi-Dynamic Scheduling (SDS) approach, there are some improvements we intend to apply on it. Firstly, the distance between the two levels can be expanded by integrating a trustful protocols and standards such as 4G or VANET. This will allow each cluster to work remotely. Then, the time interval between two devices in different levels should be enhanced by controlling overhead. In addition, the control messages (the acknowledgments) could be transmitted only for the Ready-to-Send-Nodes.

The Efficient Dynamic Scheduling (EDS) approach also needs some enhancements in terms of re-scheduling process. This waists time and consumes the On-Chip memory within the coordinator, because of the high-dimensional computation of the address assignments. Also in terms of mobility model, this approach can be enhanced to be used anywhere regardless of its location. For example in healthcare application, this model can monitor a particular patient remotely anywhere. This patient can do his daily activity without any barriers. The adds-on service can be integrated to expand the monitoring model such as using 4G network, GPS to detect the position of each patient in emergency situations. In addition, in the ambulance the VANET could be used to interact with the WBAN side as well as with the vehicular communication systems.

On the same way with the High Priority Scheduling (HPS) approach, the mobility model can be integrated easily. In terms of supporting more priority devices, another protocols can be used. This will be investigated further to find out suitable protocols. Moreover, to predict a critical health problem with a particular, patient a prediction model can be integrated as well. Thus, a study of prediction models will be done. Based on this study, one of them will be integrated into the high priority scheduling process.

Finally, all the intended work will be evaluated by performing real experimentations. The implementation will be focused on the healthcare domain. A number of scenarios are planned for the three proposals with the future improvements. First scenario will be implemented in a hospital for some patients with focusing on the lower level of the Semi-Dynamic Scheduling (SDS) approach. The upper level is always fixed, which is the non-moving part of the human body. The limbs of the human body will be equipped by the lower level devices (EDs). The routers forward the data to the PANC. The EDs sampled the medical data based on the Ready-to-Send-Nodes. Each patient has a localization module, he or she can only move within the hospital environment. The purpose of this scenario is to detect any health problem linked with the patient location within the hospital. Next scenario, will be on the same way but with more freedom. The implementation will be done at patients' home. Any particular patient can do his daily activity at home without any barriers. To expand the distance the patient will be equipped by a Personal Digital Assistant (PDA) with a GPS module and 4G connection. Thus, he or she can move also around his or her home. The same also for the priority model, more accurate healthcare problems will be checked to validate the work on large-scale environment as well.

Appendix A

TinyOS Procedure Descriptions

This appendix describes the required steps to achieve real implementation over TinyOS under Ubuntu environment.

We have validated our proposals by using 15 TelosB motes. The motes are programmed using NesC language under Tinyos version 2.1.2. They are arranged based on a cluster-tree topology. Our network is composed of 15 motes. A unique coordinator is used to establish the network, synchronize its children motes, assign address and time slots. Five routers are used for synchronizing their children motes and forward sensory data to the coordinator. Nine end devises are used to sample the environment parameters. See Figure A.1 for more details of our experimentations.

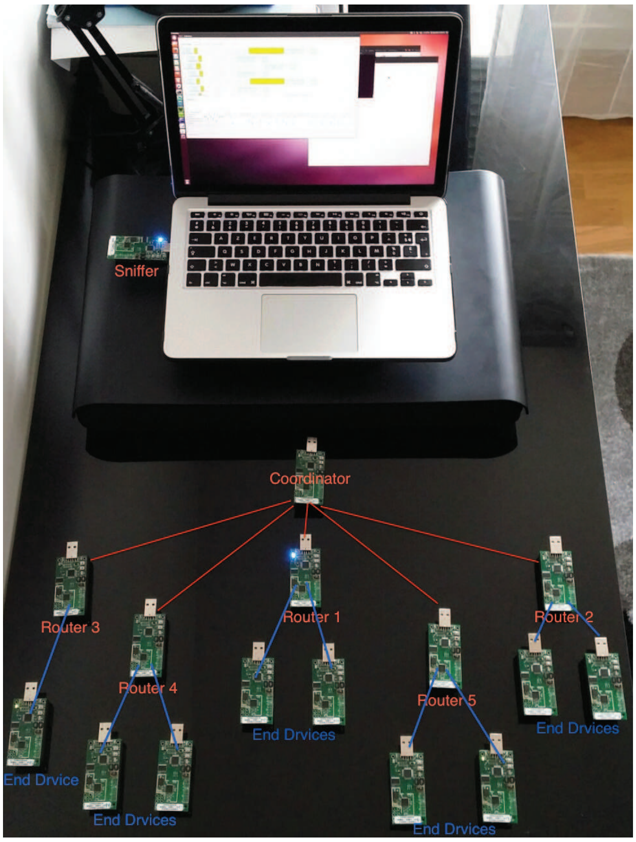
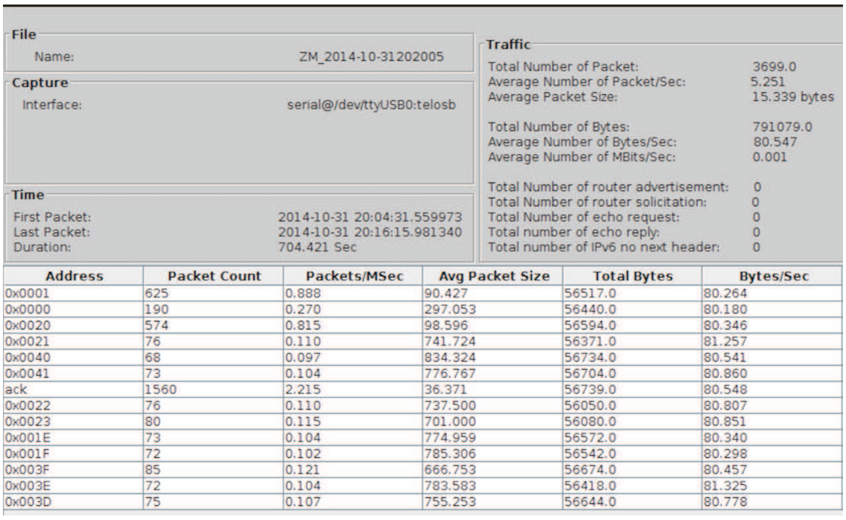


Figure A.1: The experimental network of our proposal

To evaluate the network activity we have modified the open source z-monitor which is based on "tkn154 packt sniffer". Our modifications of the original Z-Monitor was made because of some drawbacks: 1) It does not show the results for each sensor device, 2) the GUI shows only general results without knowing exactly the source of each packet. 3) Furthermore, the traffic summary window provides only overall results. 4) Finally, it dose not have a timer to start and stop the sniffer, it can be done manually.



Address	Packet Count	Packets/MSec	Avg Packet Size	Total Bytes	Bytes/Sec
0x0001	625	0.888	90.427	56517.0	80.264
0x0000	190	0.270	297.053	56440.0	80.180
0x0020	574	0.815	98.596	56594.0	80.346
0x0021	76	0.110	741.724	56371.0	81.257
0x0040	68	0.097	834.324	56734.0	80.541
0x0041	73	0.104	776.767	56704.0	80.860
ack	1560	2.215	36.371	56739.0	80.548
0x0022	76	0.110	737.500	56050.0	80.807
0x0023	80	0.115	701.000	56080.0	80.851
0x001E	73	0.104	774.959	56572.0	80.340
0x001F	72	0.102	785.306	56542.0	80.298
0x003F	85	0.121	666.753	56674.0	80.457
0x003E	72	0.104	783.583	56418.0	81.325
0x003D	75	0.107	755.253	56644.0	80.778

Figure A.2: The summary traffic window of B-Monitor

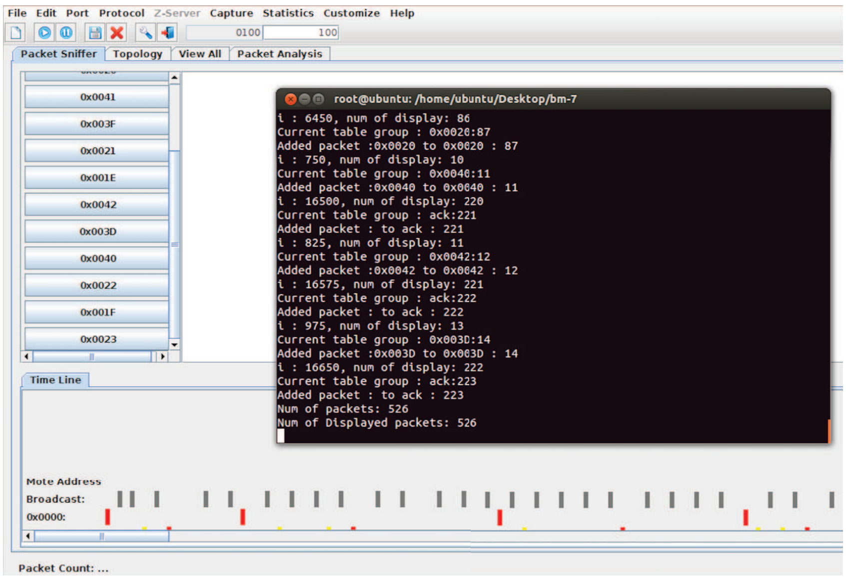


Figure A.3: The main GUI of B-Monitor

Our new version of Z-Monitor called B-Monitor (Beacon-Enabled-Monitor) sniffs the generated packets for each sensor. It provides traffic statistics for each mote in a separate log file, see Figure A.2. Recently, we have added two more features into our sniffer version, first one is for defining the time duration of sniffing the packets automatically, the sniffer will stop automatically at a pre-defined time limit. Second feature is for creating an instant log file for each evaluation based on the pre-defined time limit. These features give us the ability to measure the feasibility of our proposals efficiently, see Figure A.3, A.4, A.5 and A.6.

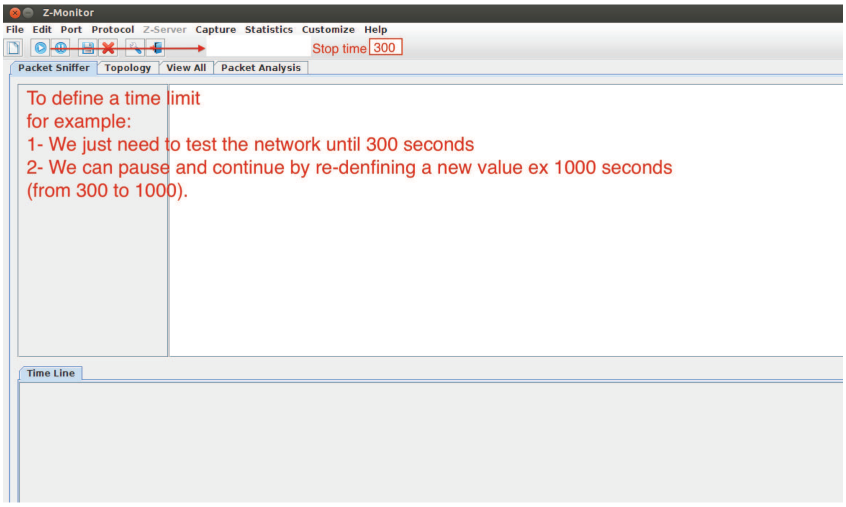


Figure A.4: The modification procedure of start-stop the B-Monitor

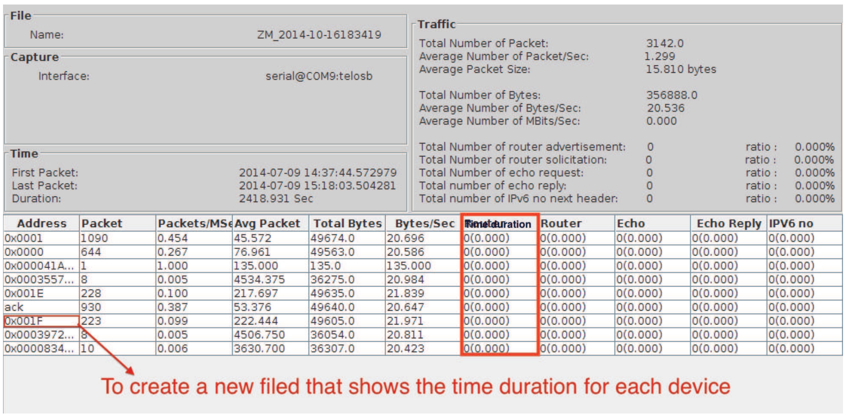


Figure A.5: The modification procedure of summary time duration at each device in the B-Monitor

Codes Examples:

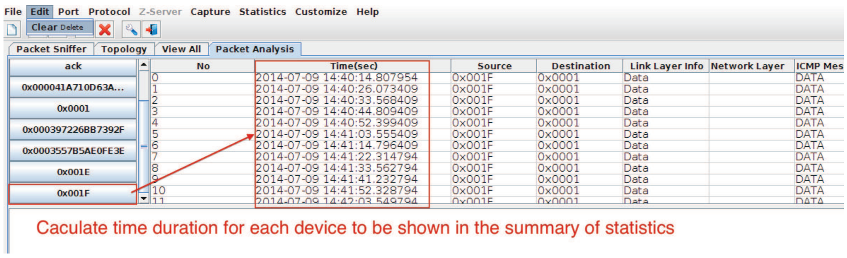


Figure A.6: The modification procedure of time duration at each device in the B-Monitor

```
if (TYPE_DEVICE == ROUTER)// && networkStarted == 0)
{
    //LastTimeSlot+=0x00003C00;

    switch(source_address)
    {

    case 0x0001:
        beacon_scheduling_ptr->request_type = SCHEDULING_ACCEPT;
        beacon_scheduling_ptr->transmission_offset[0] = 0x00;
        beacon_scheduling_ptr->transmission_offset[1] = 0x3C;
        beacon_scheduling_ptr->transmission_offset[2] = 0x00;
        break;
    case 0x0002:
        beacon_scheduling_ptr->request_type = SCHEDULING_ACCEPT;
        beacon_scheduling_ptr->transmission_offset[0] = 0x00;
        beacon_scheduling_ptr->transmission_offset[1] = 0x78;
        beacon_scheduling_ptr->transmission_offset[2] = 0x00;
        break;
    case 0x0003:
        beacon_scheduling_ptr->request_type = SCHEDULING_ACCEPT;
        beacon_scheduling_ptr->transmission_offset[0] = 0x00;
        beacon_scheduling_ptr->transmission_offset[1] = 0xF0;
        beacon_scheduling_ptr->transmission_offset[2] = 0x00;
        break;
    case 0x0004:
        beacon_scheduling_ptr->request_type = SCHEDULING_ACCEPT;
        beacon_scheduling_ptr->transmission_offset[0] = 0x00;
        beacon_scheduling_ptr->transmission_offset[1] = 0x78;
        beacon_scheduling_ptr->transmission_offset[2] = 0x00;
        break;
    default:
        beacon_scheduling_ptr->request_type = SCHEDULING_ACCEPT;
        beacon_scheduling_ptr->transmission_offset[0] = 0x01;
        beacon_scheduling_ptr->transmission_offset[1] = 0x2C;
        beacon_scheduling_ptr->transmission_offset[2] = 0x00;
        break;
    }
}

else// if (source_address > 0x0004)// && networkStarted == 0)
    {

    beacon_scheduling_ptr->request_type = SCHEDULING_ACCEPT;
    beacon_scheduling_ptr->transmission_offset[0] = LastTimeSlot & 0x000000ff;
    beacon_scheduling_ptr->transmission_offset[1] = (LastTimeSlot & 0x0000ff00)>>8;
    beacon_scheduling_ptr->transmission_offset[2] = (LastTimeSlot & 0x00ff0000)>>16;
    LastTimeSlot+=0x00012C00;
```

Figure A.7: NesC Code example of Semi Dynamic Scheduling over the PANC

```
/* PAN COORDINATOR NODE
 * @author Bandar Alghamdi <bandar.alghamdi@etudiant.univ-reims.fr>
 *
 */
//function used to schedule the beacon requests (PAN coordinator)
void process_beacon_scheduling(uint16_t source_address,uint8_t beacon_order, uint8_t superframe_order)
{
    uint8_t nsdu_pay[6];
    beacon_scheduling *beacon_scheduling_ptr = (beacon_scheduling *)(&nsdu_pay[0]);

    beacon_scheduling_ptr->beacon_order = beacon_order;
    beacon_scheduling_ptr->superframe_order = superframe_order;
    beacon_scheduling_ptr->request_type = SCHEDULING_ACCEPT;
    beacon_scheduling_ptr->transmission_offset[0] = LastTimeSlot & 0x000000ff;
    beacon_scheduling_ptr->transmission_offset[1] = (LastTimeSlot & 0x0000ff00)>>8;
    beacon_scheduling_ptr->transmission_offset[2] = (LastTimeSlot & 0x00ff0000)>>16;
    LastTimeSlot+=0x00003C00;
    lclPrintf("Coordinator: Sending negotiation reply\n", "");
    call NLDE_DATA.request(source_address, 0x06, nsdu_pay, 0, 1, 0x00, 0);
}
```

Figure A.8: NesC Code example of Dynamic Scheduling over the PANC

```
/* PAN COORDINATOR NODE
 * @author Bandar Alghamdi <bandar.alghamdi@etudiant.univ-reims.fr>
 *
 */
//function used to schedule the priority beacon requests (PAN coordinator)
void process_beacon_scheduling(uint16_t source_address,uint8_t beacon_order, uint8_t superframe_order)
{
    uint8_t nsdu_pay[6];
    int i=0;
    uint8_t flag=0;
    beacon_scheduling *beacon_scheduling_ptr = (beacon_scheduling *)(&nsdu_pay[0]);
    beacon_scheduling_ptr->beacon_order = beacon_order;
    beacon_scheduling_ptr->superframe_order = superframe_order;
    beacon_scheduling_ptr->request_type = SCHEDULING_ACCEPT;

    for (i=0;i<5;i++)
        if(HighPriorityNodes[i]==source_address)
        source_address = 0x0001, 0x0001F; // Normal 0x0001E;
            flag=1;
    if(flag==0){
        beacon_scheduling_ptr->transmission_offset[0] = LastTimeSlot & 0x000000ff;
        beacon_scheduling_ptr->transmission_offset[1] = (LastTimeSlot & 0x0000ff00)>>8;
        beacon_scheduling_ptr->transmission_offset[2] = (LastTimeSlot & 0x00ff0000)>>16;
        LastTimeSlot+=0x00003C00;
    }
    else{
        beacon_scheduling_ptr->transmission_offset[0] = LastTimeSlotHighPrio & 0x000000ff;
        beacon_scheduling_ptr->transmission_offset[1] = (LastTimeSlotHighPrio & 0x0000ff00)>>8;
        beacon_scheduling_ptr->transmission_offset[2] = (LastTimeSlotHighPrio & 0x00ff0000)>>16;
        LastTimeSlotHighPrio+=0x00003C00;
    }


    lclPrintf("Coordinator: Sending negotiation reply\n", "");
    call NLDE_DATA.request(source_address, 0x06, nsdu_pay, 0, 1, 0x00, 0);
 }
```

Figure A.9: NesC Code example of Priority-Dynamic Scheduling over the PANC

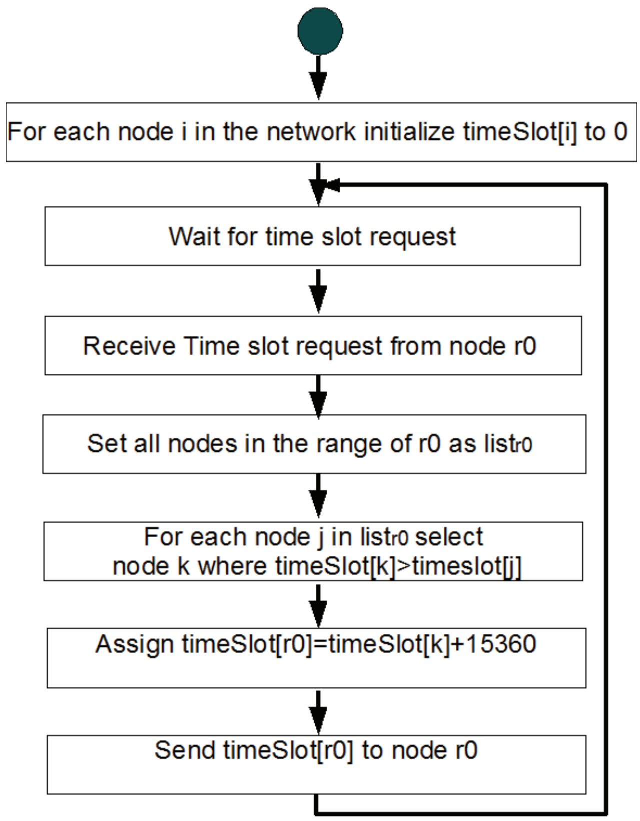


Figure A.10: The time slot assignment mechanism of our proposals

Bibliography

[1] I. for Health in University of California of San Francisco & Aging and R. W. J. Foundation, *Chronic Care in America: A 21st Century Challenge*. Robert Wood Johnson Foundation, 1996. [Online]. Available: https://books.google.fr/books?id=Ek1rAAAAMAAJ (Cited on pages 1 and 7.)

[2] A. Koubaa, A. Cunha, and M. Alves, "A time division beacon scheduling mechanism for ieee 802.15.4/zigbee cluster-tree wireless sensor networks," in *Real-Time Systems, 2007. ECRTS '07. 19th Euromicro Conference on*, July 2007, pp. 125–135. (Cited on pages 3, 9 and 96.)

[3] (2013) Z-monitor: A monitoring tool for ieee 802.15.4 wireless personal area networks. [Online]. Available: http://www.z-monitor.org (Cited on pages 4 and 10.)

[4] G. Jolly and Y. Mohamed, "An energy-efficient, scalable and collision-free mac layer protocole for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 5, pp. 285–304, 2005. (Cited on page 15.)

[5] N. A. Pantazis, D. J. Vergados, D. D. Vergados, and C. Douligeris, "Energy efficiency in wireless sensor networks using sleep mode tdma scheduling," *Ad Hoc Networks*, vol. 7, no. 5, pp. 322–343, 2005. (Cited on page 15.)

[6] C. Lin, N. Xiong, J. H. Park, and T. hoon Kim, "Dynamic power management in new architecture of wireless sensor networks," *Internationl Journal of Communication Systems*, vol. 22, no. 5, pp. 671–693, 2005. (Cited on page 15.)

[7] H. Fouchal, Y. Francillette, P. Hunel, and N. Vidot, "A distributed power management optimization in wireless sensor networks," in *Proceedings of 34th annual IEEE conference on Local Computer Networks (LCN-09)*, Zurich, Switzerland,, October 2009 2009, pp. 763–769. (Cited on page 15.)

[8] B. Yahya and J. Ben-Othman, "Towards a classification of energy-aware mac protocols for wireless sensor networks," *Wireless Communications and Mobile Computing (WCMC)*, vol. 12, no. 3, pp. 1572–1607, Feb 2009. (Cited on page 15.)

[9] B. Yahya and J. B. Othman, "Energy efficient and qos aware medium access control for wireless sensor networks," *Concurrency and Computation : Practice and Experience*, vol. 22, no. 10, pp. 1252–1266, July 2010. (Cited on page 15.)

[10] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002. (Cited on page 23.)

[11] V. Shnayder, B.-R. Chen, K. Lorincz, Thaddeus, and M. Welsh, "Sensor networks for medical care," Division of Engineering and Applied Sciences, New York, NY, USA, Tech. Rep., 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1098918.1098979 (Cited on page 23.)

[12] S. Al-Omari and W. Shi, "Toward highly-available wsns for assisted living," in *HealthNet '07: Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*. New York, NY, USA: ACM Press, 2007, pp. 31–36. (Cited on page 23.)

[13] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic, "Alarm-net: Wireless sensor networks for assisted-living and residential monitoring," 2006. (Cited on pages 23, 27 and 38.)

[14] M. Jelena and B. Vojislav, "Implementation of security policy for clinical information systems over wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 134–144, January 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B7576-4K7X875-2/2/ad52d7bb64a3626589594f3573ba62aa (Cited on page 23.)

[15] K. Malasri and L. Wang, "Addressing security in medical sensor networks," in *HealthNet '07: Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*. New York, NY, USA: ACM Press, 2007, pp. 7–12. (Cited on page 23.)

[16] D. Malan, T. Fulford-jones, M. Welsh, and S. Moulton, "Codeblue: An ad hoc sensor network infrastructure for emergency medical care," in *In International Workshop on Wearable and Implantable Body Sensor Networks*, 2004. (Cited on pages 25 and 38.)

[17] U. Anliker, J. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E. Schenker, F. Catarsi, L. Coluccini, A. Belardinelli, D. Shklarski, M. Alon, E. Hirt, R. Schmid, and M. Vuskovic, "Amon: a wearable multiparameter medical monitoring and alert system," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 8, no. 4, pp. 415–427, Dec 2004. (Cited on pages 25 and 38.)

[18] M. Blount, V. M. Batra, A. N. Capella, M. R. Ebling, W. F. Jerome, S. M. Martin, M. Nidd, M. R. Niemi, and S. P. Wright, "Remote health-care monitoring using personal care connect," *IBM Syst. J.*, vol. 46, no. 1, pp. 95–113, Jan. 2007. [Online]. Available: http://dx.doi.org/10.1147/sj.461.0095 (Cited on pages 26 and 38.)

[19] K. Navarro, E. Lawrence, and B. Lim, "Medical motecare: A distributed personal healthcare monitoring system," in *eHealth, Telemedicine, and Social*

Medicine, 2009. eTELEMED '09. International Conference on, Feb 2009, pp. 25–30. (Cited on pages 27 and 38.)

[20] J. Ko, T. Gao, R. Rothman, and A. Terzis, "Wireless sensing systems in clinical environments: Improving the efficiency of the patient monitoring process," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 29, no. 2, pp. 103–109, March 2010. (Cited on pages 28 and 38.)

[21] W. H. Wu, A. A. T. Bui, M. A. Batalin, L. K. Au, J. D. Binney, and W. J. Kaiser, "Medic: Medical embedded device for individualized care," *Artif. Intell. Med.*, vol. 42, no. 2, pp. 137–152, Feb. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.artmed.2007.11.006 (Cited on pages 28 and 38.)

[22] F. Tay, M. Nyan, D. Guo, C. Ng, L. Xu, and C. Tan, "Memswear - biomonitoring system in a body area network (ban)," in *International Semiconductor Conference, 2006*, vol. 1, Sept 2006, pp. 207–214. (Cited on pages 29 and 38.)

[23] Z. Lv, F. Xia, G. Wu, L. Yao, and Z. Chen, "icare: A mobile health monitoring system for the elderly," in *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, Dec 2010, pp. 699–705. (Cited on pages 29 and 38.)

[24] P. Kuryloski, A. Giani, R. Giannantonio, K. Gilani, R. Gravina, V.-P. Seppa, E. Seto, V. Shia, C. Wang, P. Yan, A. Yang, J. Hyttinen, S. Sastry, S. Wicker, and R. Bajcsy, "Dexternet: An open platform for heterogeneous body sensor networks and its applications," in *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, June 2009, pp. 92–97. (Cited on pages 29 and 38.)

[25] N. Oliver and O. Flores-mangas, "Healthgear: A real-time wearable system for monitoring and analyzing physiological signals," in *In Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*, 2006. (Cited on pages 30 and 38.)

[26] D. Konstantas, A. V. Halteren, R. Bults, K. Wac, I. Widya, N. Dokovsky, G. Koprinkov, V. Jones, R. Herzog, and E. Gmbh, "Mobile patient monitoring: the mobihealth system," 2009. (Cited on pages 31 and 38.)

[27] R. Chakravorty, "A programmable service architecture for mobile medical care," in *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, March 2006, pp. 5 pp.–536. (Cited on pages 31 and 38.)

[28] A. Wong, D. McDonagh, O. Omeni, C. Nunn, M. Hernandez-Silveira, and A. Burdett, "Sensium: an ultra-low-power wireless body sensor network platform: Design and application challenges," in *Engineering in Medicine and Biol-*

ogy Society, 2009. EMBC 2009. Annual International Conference of the IEEE, Sept 2009, pp. 6576–6579. (Cited on pages 31 and 38.)

[29] B. Wang, L. Wang, S. Lin, D. Wu, B. Huang, Y. Zhang, Q. Yin, and W. Chen, "A body sensor networks development platform for pervasive healthcare," in *Bioinformatics and Biomedical Engineering , 2009. ICBBE 2009. 3rd International Conference on*, June 2009, pp. 1–4. (Cited on pages 31 and 38.)

[30] G. López, V. Custodio, and J. Moreno, "Lobin: E-textile and wireless-sensor-network-based platform for healthcare monitoring in future hospital environments," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 6, pp. 1446–1458, Nov 2010. (Cited on pages 32 and 38.)

[31] R. K. Megalingam, D. Kaimal, and M. Ramesh, "Efficient patient monitoring for multiple patients using wsn," in *Advances in Mobile Network, Communication and its Applications (MNCAPPS), 2012 International Conference on*, Aug 2012, pp. 87–90. (Cited on pages 32 and 38.)

[32] L. Xuemei, J. Liangzhong, and L. Jincheng, "Home healthcare platform based on wireless sensor networks," in *Information Technology and Applications in Biomedicine, 2008. ITAB 2008. International Conference on*, May 2008, pp. 263–266. (Cited on pages 33 and 38.)

[33] R. Rashid, M. Rahim, M. Sarijari, and N. Mahalin, "Home-based healthcare monitoring over wireless biomedical sensor network." 2010. (Cited on pages 33 and 38.)

[34] B. Chen, J. Varkey, D. Pompili, J.-J. Li, and I. Marsic, "Patient vital signs monitoring using wireless body area networks," in *Bioengineering Conference, Proceedings of the 2010 IEEE 36th Annual Northeast*, March 2010, pp. 1–2. (Cited on pages 33 and 38.)

[35] Shimmer. (2013). [Online]. Available: shimmer-research.com (Cited on page 45.)

[36] Mica2dot. (2013). [Online]. Available: willow.co.uk/html/mica2do (Cited on page 46.)

[37] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: An acquisitional query processing system for sensor networks," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, Mar. 2005. [Online]. Available: http://doi.acm.org/10.1145/1061318.1061322 (Cited on page 47.)

[38] J. Ben-Othman, K. Bessaoud, A. Bui, and L. Pilard, "Self-stabilizing algorithm for efficient topology control in wireless sensor networks," *Journal of Computational Science*, vol. 4, no. 4, pp. 199 – 208, 2013, {PEDISWESA} 2011 and Sc. computing for Cog. Sciences. [Online]. Available:

http://www.sciencedirect.com/science/article/pii/S1877750312000051 (Cited on page 47.)

[39] UK-Home-Health. (2013). [Online]. Available: homehealth-uk.com (Cited on page 53.)

[40] N. Dessart, P. Hunel, H. Fouchal, and N. Vidot, "Population protocol over wireless sensor networks," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, Oct 2010, pp. 799–805. (Cited on page 55.)

[41] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: ACM, 1992, pp. 144–152. [Online]. Available: http://doi.acm.org/10.1145/130385.130401 (Cited on page 56.)

[42] J. Hilbe, *Logistic Regression Models*, ser. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2009. [Online]. Available: https://books.google.es/books?id=eJcMIAAACAAJ (Cited on page 57.)

[43] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* Norwell, MA, USA: Kluwer Academic Publishers, 1981. (Cited on page 57.)

List of Publications

1. ***International Peer-Reviewed Journals***

    - Al-Ghamdi, B.; Fouchal, H., "A mobile wireless body area network platform", in J. Comput. Science 5(4), pp 664-674 (2014)

2. ***International Peer-Reviewed Conferences***

    - Al-Ghamdi, B.; Ayaida, M.; Fouchal, H., "Scheduling Approaches for Wireless Sensor Networks," in Innovations for Community Services (I4CS), 2015 15th International Conference on, 08-10 July 2015
    - Al-Ghamdi, B.; Ayaida, M.; Fouchal, H., "A dynamic slot scheduling for wireless sensors networks," in Global Communications Conference (GLOBECOM), 2014 IEEE, pp. 357-361, 8-12 Dec. 2014 doi: 10.1109/GLOCOM.2014.7036834
    - Al-Ghamdi, B.; Ayaida, M.; Fouchal, H.; Baala, H., "A slot assignment for Wireless Body Area Networks," in Computers and Communication (ISCC), 2014 IEEE Symposium on, pp. 1-8, 23-26 June 2014, doi: 10.1109/ISCC.2014.6912630
    - Al-Ghamdi, B.; Ayaida, M.; Fouchal, H., "Semi-Dynamic Tree Scheduling Process for Wireless Sensor Networks," in Innovations for Community Services (I4CS), 2014 14th International Conference on, pp. 76-79, 4-6 June 2014, doi: 10.1109/I4CS.2014.6860556
    - Al-Ghamdi, B.; Fouchal, H., "Vehicular Communication projects classification and evaluation," in Computers and Communications (ISCC), 2013 IEEE Symposium on, pp. 43-48, 7-10 July 2013 doi: 10.1109/ISCC.2013.6754921
    - Al-Ghamdi, B.; Fouchal, H., "Wireless body area network platforms evaluation," in Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, pp. 1348-1352, 1-5 July 2013 doi: 10.1109/IWCMC.2013.6583752
    - Al-Ghamdi, B.; Ayaida, M.; Fouchal, H., "Wireless Body Area Network Platform Architecture," in Innovative Internet Community Systems 2013 (I4CS), 2013 13th International Conference on, 19 - 21 June 2013

Titre de la thèse– Etude des méthodes d'ordonnancement sur les réseaux de capteurs sans fil
Résumé– Les Wireless Body Area (WBAN) sont une technologie de réseau sans fil basée sur les radio-fréquences qui consiste à interconnecter sur, autour ou dans le corps humain de minuscules dispositifs pouvant effectuer des mesures (capteurs). Ces réseaux sont considérés comme les plus critiques dans les réseaux de capteurs sans fil. Ils sont basés sur des architectures de réseaux auto-organisés. Chacun des capteurs corporels reçoit ou envoie des paquets du ou au coordinateur du réseau. Ce dernier est responsable de l'ordonnancement des tâches pour l'ensemble des noeuds fils. L'ordonnancement dans les WBAN nécessite un mécanisme dynamique et adaptatif pour gérer les cas d'urgence qui peuvent se produire et permet ainsi d'améliorer les paramètres les plus importants comme la qualité de la transmission, le temps de réponse, le débit, le taux de paquets délivres, etc. Dans ces travaux de thèse, nous avons proposé trois techniques d'ordonnancement qui sont : la méthode semi-dynamique; la méthode dynamique et la méthode basée sur la priorité. De plus, une étude sur les plateformes WBAN est présentée. Dans cette étude, nous avons proposé une classification et une évaluation qualitative des plateformes déjà existantes. Nous avons aussi étudier les modèles de mobilité en proposant une architecture permettant de les décrire. Nous avons aussi mis en place une procédure de diagnostique afin de détecter rapidement des maladies épidémiques dangereuses. Par la suite, ces différentes propositions ont été validées en utilisant deux méthodes afin de vérifier leur faisabilité. Ces méthodes sont la simulation avec OPNET et l'implémentation réelle sur des capteurs TelosB et TinyOS.
Mots Clés – WSN; WBAN; Algorithmes d'ordonnancement; Optimisation; Couche MAC; Protocoles de communication.

Title of PhD Thesis– Study on Scheduling over Wireless Sensor Networks
Abstract – The Wireless Body Area Network (WBAN) is the most critical field when considering Wireless Sensor Networks (WSN). It must be a self-organizing network architecture, meaning that it should be able to efficiently manage all network architecture requirements. The WBAN usually contains at least two or more body sensors. Each body sensor sends packets to or receives packets from the Personal Area Network Coordinator (PANC). The PANC is responsible for scheduling its child nodes' tasks. Scheduling tasks in the WBAN requires a dynamic and an adaptive process in order to handle cases of emergency that can occur with a given patient. To improve the most important parameters of a WBAN, such as quality link, response time, throughput, the duty-cycle, and packet delivery, we propose three scheduling processes: the semi-dynamic, dynamic, and priority-based dynamic scheduling approaches. In this thesis, we propose three task scheduling techniques, Semi-Dynamic Scheduling (SDS), Efficient Dynamic Scheduling (EDS) and High Priority Scheduling (HPS) approaches. Moreover, a comprehensive study has been performed for the WBAN platforms by classifying and evaluating them. We also investigate the mobility model for the WBANs by designing an architecture that describe this model. In addition, we detail a diagnosis procedure by using classification methods in order to solve very sensitive epidemic diseases. Then, our proposals have been validated using two techniques to check out the feasibility of our proposals. These techniques are simulation scenarios using the well-known network simulator OPNET and real implementations over TelosB motes under the TinyOS system.
Keywords – WSN; WBAN; Scheduling Algorithms; Optimization; MAC layer; Communication protocols.

Affiliation – Département Mathématiques, Mécanique et Informatique (MMI)
UFR Sciences Exactes et Naturelles
Campus du Moulin de la Housse - BP 1039 - 51687 Reims Cedex 2