

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Saad Dahlab - Blida I -

N° D'ordre :



Faculté des sciences
 Département d'informatique

Mémoire Présenté par :

EL MEDJADJI Djilali

HARRACHIF Aniss

En vue d'obtenir le diplôme de Master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Génie des systèmes informatiques

Sujet : Etude, fonctionnement et réalisation d'un réseau de capteurs sans fil

Soutenu le : / 06/2016

Devant le jury :

M. m^e. *Arroussi*

Président

M. m^e. *Bentoumi*

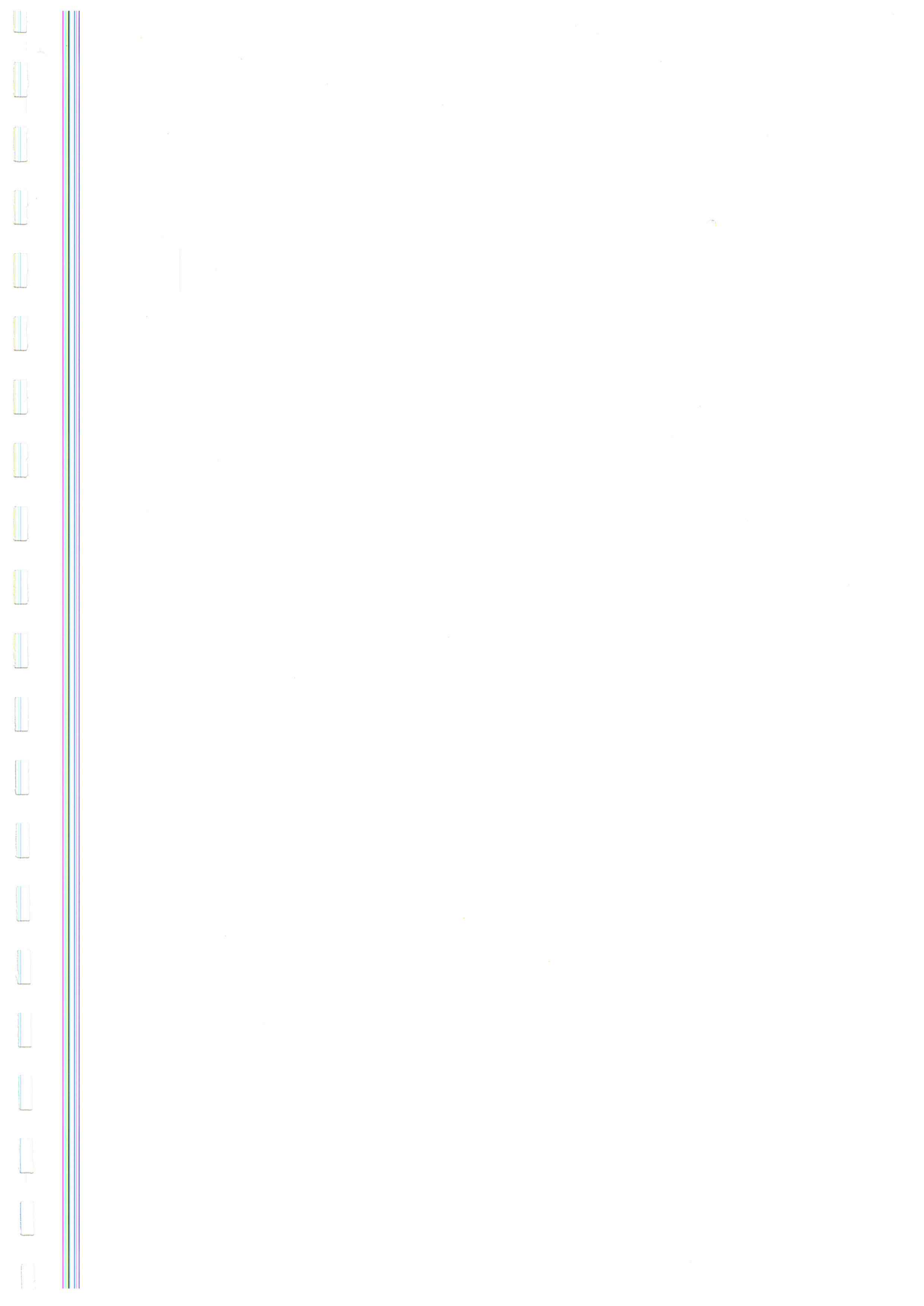
Examineur

M. ALLAM ABDELKRIM

Encadrant

MA-004-295-1

Promotion
 2015/2016



Remerciements

Le travail réalisé tout au long de notre formation n'a été possible qu'avec l'aide et le soutien de nombreuses personnes.

Nous profitons de l'occasion qui nous est donnée ici pour exprimer toute notre gratitude à toutes les personnes qui nous ont suivi et soutenu durant notre cursus universitaire au sein de l'université *SAAD DAHLEB* de Blida.

En premier lieu, nous tenons à remercier notre directeur de projet *Dr.ABDELKRIM ALLAM* pour l'investissement sans faille dont il a fait preuve dans notre encadrement tout au long de cette année. Il a été au cours de ce stage au *Centre de Développement des Technologies Avancées* bien plus qu'un simple directeur de projet et nous ne le remercierons jamais assez. C'est grâce à sa confiance, sa disponibilité, ses conseils et la liberté qu'il a su nous donner que nous avons pu mener à bien ce travail.

Nous tenons à remercier aussi *Dr.CHIKHI NASSIM*, *Mlle ARKAM MALEK* et *Mlle OUKID LAMIA* pour l'intérêt qu'ils ont montré envers notre travail en acceptant de conseiller, guider et corriger nos résultats.

Nos profonds remerciements vont aussi pour les membres de jury qui ont accepté d'évaluer ce travail.

Dédicace

À nos parents et nos famille,

Aucune dédicace ne peut exprimer tout ce que nous ressentons pour vous. Nous vous remercions pour tout le soutien exceptionnel que vous nous portez chaque jour.

À tous nos amis,

Quoi que nous puissions dire, nous ne pouvons exprimer nos sentiments d'amour et d'amitié à votre égard.

Le doute est la clé de toute connaissance...

Résumé

Les améliorations récentes dans les dispositifs électroniques intégrés abordables et efficaces ont considérablement fait avancer la technologie des capteurs sans fil. Les capteurs sans fil disponibles aujourd'hui sont équipés de modules de détection, émetteurs-récepteurs radio, des unités de traitement, de stockage et d'actionneurs. Ces capteurs forment des réseaux de capteurs sans fil (WSN) et collaborent entre eux pour effectuer diverses fonctionnalités dans différentes applications, telles que la surveillance de l'environnement, la poursuite de cibles et de sauvetage en cas de catastrophe. L'utilité d'un réseau de capteurs sans fil dans ces applications découle principalement à partir des données qu'il recueille. Ainsi, l'une des questions clés des défis des WSN est de fournir efficacement des données en fonction des demandes des utilisateurs dans un réseau de capteurs à grande échelle. On parle de : la diffusion des données.

Un réseau de capteurs sans fil est une collection de périphériques minuscules de faible puissance, appelés noeuds capteurs. Ces derniers sont capables de surveiller l'environnement, de collecter et de traiter les données détectées, et de communiquer les uns avec les autres afin d'accomplir certaines tâches courantes. De plus, les réseaux de capteurs sans fil possèdent un point de rassemblement central, appelé station de base, où toutes les données recueillies peuvent être stockées. Le défi majeur dans la conception et le développement des réseaux de capteurs sans fil est principalement due à des contraintes sévères qui sont imposées à l'architecture du réseau.

Dans ce mémoire, nous nous sommes attachés à présenter deux architectures qui permettent de s'affranchir de ces contraintes qui mettent à mal ce concept d'indépendance des dispositifs, et qui portent préjudice aux performances globales des réseaux de capteurs, en particulier dans certaines situations spécifiques de surveillance.

Pour cela, nous avons proposé une architecture basée sur une topologie en étoile qui permet de collecter les données issues du réseau de capteurs.

Une autre architecture en multi-sauts qui va nous permettre d'étendre la zone de couverture du réseau.

Mots-clés : Réseaux de capteurs sans fil, Kit Professionnel Advanticsys, Plateforme Telosb, Bridge SG1000.

Abstract

Recent improvements in affordable and efficient integrated electronic devices have significantly advanced the technology of wireless sensors. Wireless sensors available today are equipped with sensing modules, radio transceivers, processing units, storage and possible actuators. These sensors form a wireless sensor network (WSN) and collaborate with each other to perform various functionalities in different applications, such as environment surveillance, target tracking and disaster rescue. The utility of a sensor network in these applications appears primarily from the data it gathers. Thus, one of the key issues of a WSN application is to efficiently deliver data according to user requests in a large scale sensor network. This is referred to in the literature as data dissemination.

A wireless sensor network is a collection of low-powered, physically tiny devices, called sensor nodes, which are capable of sensing the physical environment, collecting and processing sensed data, and communicating with each other in order to accomplish certain common tasks.

Wireless sensor networks possess a central gathering point, called the sink (or base station), where all the collected data can be stored. The major challenge in the design and development of wireless sensor networks is mainly due to the severe constraints that are imposed on the network architecture.

In this paper we made sure to present two different types of architecture that allow overcoming these constraints that undermine the concept of total independence of devices and more, which can damage the overall performance of sensor networks in specific situations.

The second architecture is based on a multi-hop relay topology which allows us to expand the coverage zone using relay nodes in our wireless sensor network.

Keywords : Wireless sensors Network, Advanticsys Professional Kit, Telosb Platform, Bridge SG1000.

ملخص

ساهمت التحسينات الأخيرة في الأجهزة الإلكترونية المتكاملة بأسعار معقولة وفعالة في تقدم تكنولوجيا الاستشعار بشكل كبير حيث تم تجهيز أجهزة الاستشعار اللاسلكية المتاحة اليوم بوحدة الكشف وأجهزة الإرسال والاستقبال اللاسلكية إضافة إلى وحدات المعالجة والتخزين والمحركات. هذه المكونات تشكل شبكات استشعار لاسلكية تتعاون مع بعضها البعض لأداء وظائف مختلفة في تطبيقات مختلفة، مثل الرصد البيئي، وتتبع الهدف والإنقاذ عند حدوث الكوارث. تكمن فائدة شبكة الاستشعار اللاسلكية في المقام الأول من خلال البيانات التي تجمعها. وهكذا، فإن التحدي الرئيسي الذي يمس هذه الشبكات هو توفير البيانات بشكل فعال بناء على طلبات من المستخدمين في شبكة استشعار على نطاق واسع وهو ما يطلق عليه: فاعلية نشر البيانات .

شبكات الاستشعار اللاسلكية هي عبارة عن مجموعة من الأجهزة منخفضة الطاقة صغيرة، تسمى عقد الاستشعار. وهي قادرة على مراقبة البيئة، وجمع ومعالجة البيانات، والتواصل مع بعضها البعض من أجل إنجاز بعض المهام المشتركة.

وبالإضافة إلى ذلك، شبكات الاستشعار اللاسلكية لديها نقطة تجمع مركزية، ويطلق عليها: المحطة الأساسية، حيث يمكن تخزين جميع البيانات التي تم جمعها. يتمثل العائق الأصعب في شبكات الاستشعار اللاسلكية من خلال تصميم وتحقيق هندسات تسمح بتجاوز الصعوبات والقيود المتعلقة بهذا الصنف من الشبكات .

سعيًا في هذه المذكرة إلى تقديم نمطين من الهندسات التي تسمح بالتغلب على بعض القيود التي تقوض مفهوم استقلالية هذه الأجهزة، والتي تضر الأداء العام لشبكات الاستشعار اللاسلكية، وخاصة في حالات معينة ومحددة .

لذا، اقترحنا في المقام الأول هندسة تركز على طوبولوجيا STAR التي تقوم بجمع البيانات من شبكة أجهزة الاستشعار وإرسالها مباشرة إلى المحطة الرئيسية من أجل الحفظ والتخزين.

يتمثل الاقتراح الثاني في هندسة مستنبطة من طوبولوجيا MULTI-HOP والتي تسمح بتوسيع الشبكة وتغطية مساحة جغرافية أكبر.

الكلمات الرئيسية: شبكات الاستشعار اللاسلكية، المستشعر Telosb، الطقم اللاسلكي Advanticsys، المكثف SG1000.

Table des matières

Introduction générale	12
1 Généralités sur les réseaux de capteurs	14
1.1 Introduction	14
1.2 Présentation des réseaux de capteurs sans fil	14
1.2.1 Définition d'un capteur	14
1.2.2 Architecture des capteurs	15
1.2.3 Les contraintes en ressources des capteurs	16
1.2.4 Les Réseaux de capteurs	17
1.2.5 Les types de trafic dans les réseaux de capteurs	18
1.3 Domaines d'application des réseaux de capteurs	18
1.3.1 Environnement	18
1.3.2 Surveillance et détection	19
1.3.3 Domaine militaire	19
1.3.4 Domaine médicale	19
1.3.5 Grand public	20
1.4 Les catégories des réseaux sans fil	21
1.4.1 Le réseau étendu sans fil (WWAN)	21
1.4.2 Réseaux métropolitains sans fil (WMAN)	22
1.4.3 Réseaux locaux sans fil (WLAN)	22
1.4.4 Réseaux personnels sans fil (WPAN)	22
1.5 Comparaison des technologies sans fil	23
1.6 Topologies de la norme Zigbee - IEEE 802.15.4	23
1.6.1 La Topologie en Étoile	24
1.6.2 La Topologie en Mesh	24
1.6.3 La topologie en Arbre	25
1.7 La pile protocolaire des RSCF	25
1.7.1 La couche physique	26
1.7.2 La couche liaison de données	26
1.7.3 La couche réseau	26
1.7.4 La couche transport	26
1.7.5 La couche application	27
1.7.6 Le niveau de gestion d'énergie	27
1.7.7 Le niveau de gestion de mobilité	27
1.7.8 Le niveau de gestion des tâches	27
1.8 Conclusion	28

2	Configuration et analyse du réseau de capteurs sans fil	29
2.1	Introduction	29
2.2	Outils de développement	29
2.2.1	Le Kit Professionnel Advanticsys	29
2.2.2	Capteur Telosb	30
2.2.3	Capteur XM1000	30
2.2.4	Capteur CM5000	31
2.2.5	Capteur CM4000	32
2.2.6	Le Bridge SG1000	34
2.2.7	Décomposition analytique des trames	35
2.2.8	Décodage des trames	36
2.3	Conception du réseau	37
2.4	Configuration des architectures du réseau	37
2.4.1	Architecture en étoile	37
2.4.2	Architecture en multi-saut	39
2.5	Configuration des noeuds	41
2.5.1	Choix du langage de programmation NesC	41
2.5.2	Les Nœuds capteurs	42
2.5.3	Les Nœuds relais	44
2.6	Conclusion	45
3	Conception et réalisation de l'application de surveillance	46
3.1	Introduction	46
3.2	Analyse et spécification des besoins	46
3.2.1	Spécification des besoins fonctionnels	46
3.2.2	Spécification des besoins non fonctionnels	47
3.3	Conception du système	47
3.3.1	Diagramme de cas d'utilisation (use cases)	47
3.3.2	Diagrammes de séquence	52
3.3.3	Diagrammes de classe	58
3.3.4	Règles de gestion	60
3.4	Le passage vers le modèle relationnel	61
3.5	Implémentation de l'application	62
3.5.1	Présentation des outils de développement	62
3.5.2	Présentation de l'application	63
3.6	Conclusion	74
	Conclusion générale	75
A	Annexes	76

Table des figures

1.1	Capteur intelligent sans fil - CM5000	15
1.2	Architecture d'un nœud capteurs sans fil	16
1.3	Exemple d'un réseau d'un capteurs sans fil	17
1.4	Domaines d'applications des réseaux de capteurs sans fil	21
1.5	Les différentes catégories des réseaux sans fil	21
1.6	Représentation de la topologie en étoile	24
1.7	Représentation de la topologie en mesh	24
1.8	Représentation de la topologie en arbre	25
1.9	Pile protocolaire d'une architecture de réseau de capteurs sans fil [1]	26
2.1	Le Kit Professionnel Advanticsys	29
2.2	La plate-forme Telosb	30
2.3	La Plate-forme XM1000	31
2.4	La plate-forme CM5000	31
2.5	La plate-forme CM4000	32
2.6	La passerelle SG1000 - Advanticsys Technologie	34
2.7	Procédure d'enregistrement des données	34
2.8	Représentation des trames du SG1000	35
2.9	Représentation globale du réseau de capteur sans fil	37
2.10	Architecture en étoile	38
2.11	Architecture en multi-saut à 4 niveaux - v1	40
2.12	Architecture en multi-saut à 4 niveaux - v2	41
2.13	Interaction nœud capteur-bridge SG1000	42
2.14	La fonction Temperature.readDone()	42
2.15	Transtypage des données	43
2.16	Déclenchement de l'interface radio au niveau du capteur	43
2.17	Réception , transtypage et renvoi des données	44
3.1	Diagramme de cas d'utilisation	49
3.2	Diagramme de cas d'utilisation : gestion des utilisateurs	50
3.3	Diagramme de cas d'utilisation : gestion des sites	50
3.4	Diagramme de cas d'utilisation : gestion des nœuds	51
3.5	Diagramme de cas d'utilisation : Réglage des Seuils	51
3.6	Diagramme de séquence : Authentification	53
3.7	Diagramme de séquence : Créer nœud	54
3.8	Diagramme de séquence : Gestion des nœuds	55
3.9	Diagramme de séquence : créer réglage	57
3.10	Diagramme de classe	59
3.11	l'interface d'authentification	63

3.12	Le menu fichier	64
3.13	Le menu affichage	64
3.14	Le menu configuration	65
3.15	Menu Utilisateur	65
3.16	Menu Alerte	65
3.17	L'interface principale : Tableau de bord	66
3.18	L'interface : panneau des configurations	67
3.19	L'interface : États des capteurs	68
3.20	L'interface : l'historique des température	69
3.21	Gestion des Sites	70
3.22	Gestion des Noeuds	71
3.23	Interface Fiche Réglage	72
3.24	Interface Alerte	73

Liste des tableaux

1.1	Comparaison des protocoles Zigbee, Bluetooth et Wi-Fi	23
2.1	Les propriétés du Telosb de MEMSIC	30
2.2	Les propriétés du XM1000	31
2.3	Les propriétés du Telosb du CM5000	32
2.4	Les propriétés du Telosb du CM4000	33
3.1	Identification des acteurs	48
3.2	Description du diagramme de classe	58
3.3	Dictionnaire des données de la classe utilisateur	61

Liste des abréviations

WLAN : Wireless Local Area Network

WMAN : Wireless Metropolitan Area Network

WPAN : Wireless Personal Area Network

WWAN : Wireless Wide Area Network

WSN : Wireless sensors network

WECA : Wireless Ethernet Compa-tibility Alliance

LLC : Logical LinkControl

MAC : Medium Access Control

UML : Unified Modeling Language

Introduction générale

Les avancées technologiques et techniques opérées dans le domaine des réseaux sans fil, de la micro-fabrication et de l'intégration des microprocesseurs ont fait naître une nouvelle génération de réseaux à grande échelle adaptés à une gamme d'applications très variée, ces réseaux sont appelés Réseaux de Capteurs Sans Fil. Ces réseaux sont constitués par de petits appareils électroniques, autonomes, équipés de capteurs et capables de communiquer entre eux par des liens radios. Ces réseaux sont capables de superviser une région ou un phénomène d'intérêt, de fournir des informations utiles par des mesures prises par les différents capteurs et de les communiquer ensuite via le support sans fil à l'utilisateur.

Cette nouvelle technologie promet de révolutionner notre façon de vivre, de travailler et d'interagir avec l'environnement physique qui nous entoure. En effet, des capteurs communicants sans fil et dotés de capacités de calcul facilitent une série d'applications irréalisables ou trop chères il y a quelques années. Aujourd'hui, des capteurs minuscules et de faible coût peuvent être éparpillés sur des zones de contrôle pour détecter une variété de phénomènes physiques. De nombreux domaines d'application sont alors envisagés tels que la détection et la surveillance des désastres, le contrôle de l'environnement, le bâtiment intelligent, l'agriculture de précision, la surveillance et la maintenance préventive des machines, la médecine et la santé, la logistique et les transports intelligents.

C'est dans ce cadre que s'inscrit notre projet de fin d'études. L'objectif majeur de ce travail consiste à réaliser une application de surveillance de l'environnement à base d'un réseau de capteurs sans fil afin de permettre à l'utilisateur de surveiller l'environnement en temps-réel. De plus, nous avons pu développer une application sur les nœuds capteurs afin de permettre la détection des paramètres de l'environnement tel que température, humidité et luminosité.

Pour relater les travaux réalisés dans le cadre de notre projet, notre mémoire s'organisera de la façon suivante :

- Dans le premier chapitre, nous présentons les réseaux de capteurs sans fil, l'architecture et la composition des capteurs sans fil.
Par la suite nous parlerons des différents domaines d'applications des RCSF et également des différentes technologies de communication sans fil.

- Dans le deuxième chapitre, nous détaillons la configuration et la vue globale du réseau à réaliser, ainsi que l'analyse des différentes parties qui composent le réseau.
- Enfin, le troisième chapitre sera réservé à la conception et la réalisation de l'application de surveillance.

Chapitre 1

Généralités sur les réseaux de capteurs

1.1 Introduction

Les avancées technologiques récentes confortent la présence de l'informatique et de l'électronique au cœur du monde réel. De plus en plus d'objets se voient ainsi équipés de processeurs et de moyens de communication mobiles, leur permettant de traiter des informations mais également de les transmettre.

Les réseaux de capteurs sans-fil entrent dans ce cadre. En effet, ceux-ci sont constitués d'un ensemble de petits appareils, ou capteurs, possédant des ressources particulièrement limitées, mais qui leur permettent néanmoins d'acquérir des données sur leur environnement immédiat, de les traiter et de les communiquer.

Dans ce chapitre, nous définissons dans un premier temps les capteurs sans fil, leurs architectures et leurs compositions ainsi que le concept général des réseaux de capteurs sans fil. Ensuite, nous présentons les différents domaines d'applications des réseaux de capteurs sans fil. Enfin, nous nous intéressons aux différentes technologies et normes de communication sans fil.

1.2 Présentation des réseaux de capteurs sans fil

1.2.1 Définition d'un capteur

Un capteur est un petit dispositif électronique qui peut transformer une mesure physique en une mesure électronique, qui sera ensuite traduite en données binaires compréhensibles pour des systèmes informatiques.

Aujourd'hui il existe différents capteurs qui sont conçus pour mesurer des valeurs diverses, par exemple la température, l'humidité, la luminosité, les mouvements, la pression etc [2].

Un autre terme qui est largement utilisé dans la littérature est noeud intelligent, qui désigne particulièrement les noeuds programmables. Le développement des technologies des systèmes embarqués a donné la possibilité d'utiliser le même modèle de noeud pour

des applications différentes. Les noeuds du même modèle peuvent être déployés pour assurer des missions complètement différentes, et il suffit de changer le programme installé. Par ailleurs, l'apparition des systèmes d'exploitation pour les noeuds a encore facilité la création des applications en divisant la procédure en deux parties indépendantes : le développement des programmes et la conception de la plate-forme matérielle.

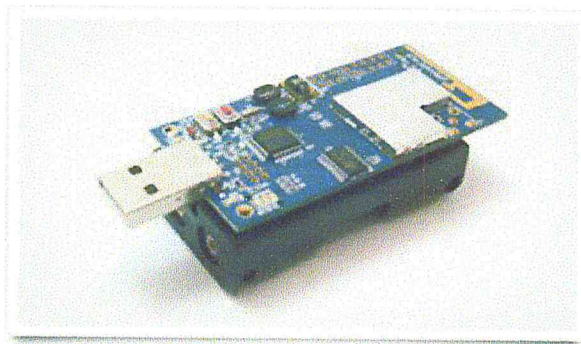


FIGURE 1.1: Capteur intelligent sans fil - CM5000

1.2.2 Architecture des capteurs

L'architecture des capteurs a beaucoup évolué depuis les années 90. Les capteurs de la première génération ont une architecture très simple, qui ne possède qu'une unité de capture et une unité d'alimentation. Comme les détecteurs de fumée qui sont encore beaucoup utilisés dans les immeubles. Ils sont souvent alimentés par des piles, et la seule fonctionnalité est de déclencher une alarme sonore en présence de fumée.

L'architecture des noeuds actuels est devenue beaucoup plus complexe après l'ajout des fonctionnalités complémentaires, comme le traitement et le stockage de données.

Nous pouvons constater qu'elle est constituée de quatre unités principales [1].

- ***l'unité d'acquisition*** : l'unité d'acquisition est composée d'un capteur qui va obtenir des mesures numériques sur les paramètres environnementaux (température, humidité...) et d'un convertisseur Analogique/Numérique dit ADC qui va convertir l'information relevée et la transmettre à l'unité de traitement.
- ***l'unité de traitement*** : C'est l'unité la plus importante d'un noeud intelligent, comme les autres systèmes embarqués, on utilise généralement un micro-contrôleur qui rassemble tous les éléments essentiels d'un ordinateur, le processeur, la mémoire et les interfaces entrées-sorties.
- ***l'unité de communication*** : Elle désigne généralement l'antenne radio du noeud, qui est chargée de l'émission et de la réception des signaux radio. Dans la littérature elle s'appelle aussi *transceiver*, qui est en effet une contraction des mots anglophones *transmitter* et *receiver*. À l'aide de cette unité, on peut construire des réseaux de capteurs en reliant les noeuds qui sont déployés dans la même zone.

Ces trois unités sont alimentées par une batterie comme le montre la figure 1.2.

- **Unité d'alimentation** : Une unité cruciale pour toute l'architecture, car c'est elle qui fournit l'énergie à toutes les autres unités. elle correspond souvent à une pile et une batterie qui s'épuise graduellement au fil du temps. À cause de la limitation des ressources des nœuds, la consommation d'énergie est devenu un facteur critique dans quasiment toutes les applications de capteurs.

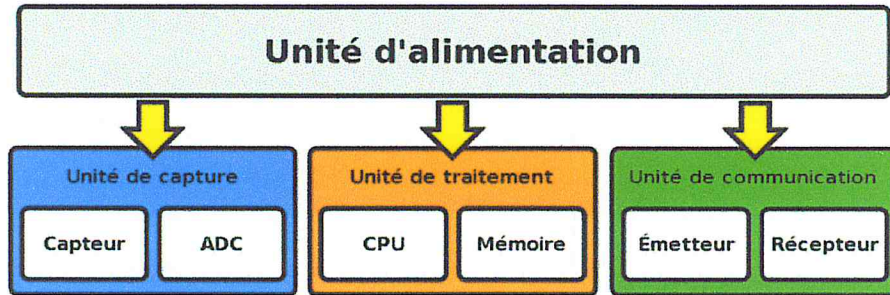


FIGURE 1.2: Architecture d'un nœud capteurs sans fil

1.2.3 Les contraintes en ressources des capteurs

Les capteurs sans fil déployés dans des zones souvent difficiles d'accès, n'embarquent qu'une quantité limitée de matériel, qui ne peut pas toujours être remplacé. Les capteurs se retrouvent donc avec des capacités limitées [3], notamment en ce qui concerne :

- **les capacités de calcul** : les processeurs embarqués sont relativement peu puissants, principalement pour des raisons de coût. Les algorithmes exécutés par les capteurs doivent donc être de complexité relativement basse ;
- **les capacités de mémoire** : les capteurs disposent de mémoire vive (RAM, Random Access Memory, « mémoire à accès non séquentiel » en anglais) et d'un peu d'espace de stockage, mais ils ne sont pas du tout conçus pour sauvegarder de grandes bases de données. Les informations récoltées doivent être acheminées à la station de base, et non stockées sur le long terme par les capteurs eux-mêmes ;
- **l'énergie disponible** : les capteurs disposent d'une batterie qui leur fournit une quantité d'énergie finie, et (la plupart du temps) non rechargeable. Il est donc essentiel de conserver à l'esprit une gestion parcimonieuse de l'énergie pour tout programme implémenté sur les capteurs. Des calculs importants, ainsi que des émissions/réceptions d'ondes électromagnétiques nombreuses ou mal gérées, sont les principaux facteurs d'un épuisement prématuré de la batterie.

Il est à noter qu'au regard de ces contraintes qui affectent les capteurs, la station de base est considérée comme disposant de capacités « illimitées »

1.2.4 Les Réseaux de capteurs

Les réseaux de capteurs sans fil *WSN*¹ sont un type particulier de réseau, dans lesquels les noeuds sont des « capteurs intelligents ». Ils se composent généralement d'un grand nombre de capteurs communicants entre eux via des liens radio pour le partage d'information et le traitement coopératif. Dans ce type de réseau, les capteurs échangent des informations par exemple sur l'environnement pour construire une vue globale de la région contrôlée, qui est rendue accessible à l'utilisateur externe par un ou plusieurs noeud(s) [4]. Les données collectées par ces capteurs sont acheminées directement ou via les autres capteurs de proche en proche à un « point de collecte », appelé station de base (ou SINK s'il s'agit d'un noeud). Cette dernière peut être connectée à une machine puissante via internet ou par satellite. En outre, l'utilisateur peut adresser ses requêtes aux capteurs en précisant l'information d'intérêt.

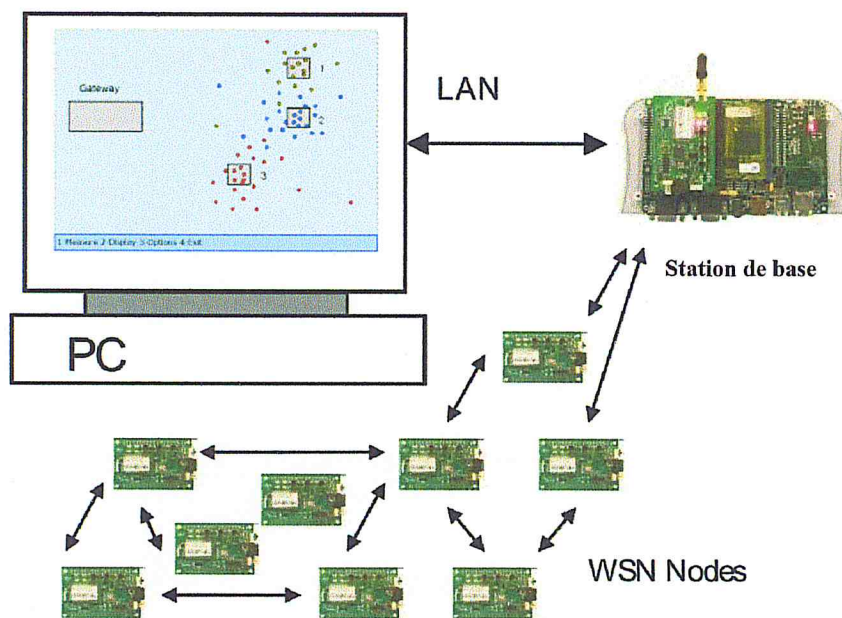


FIGURE 1.3: Exemple d'un réseau d'un capteurs sans fil

Un exemple de réseaux de capteurs est fourni dans la Figure 1.3 : les capteurs sont déployés d'une manière aléatoire dans une zone d'intérêt, et une passerelle, située à l'extrémité de cette zone, est chargée de récupérer les données collectées par les capteurs. Lorsqu'un capteur détecte un événement pertinent, un message d'alerte est envoyé à la passerelle par le biais d'une communication Radio. Les données collectées sont traitées et analysées par un ordinateur par la suite.

Les réseaux de capteurs viennent en soutien de l'environnement et de l'industrie grâce aux récents développements réalisés dans le domaine des techniques sans fils. Depuis quelques décennies, le besoin d'observer et de contrôler des phénomènes physiques tels que la température, l'humidité ou encore la luminosité est essentiel pour de nombreuses applications industrielles et scientifiques.

1. *WSN* : Wireless Sensor Network

1.2.5 Les types de trafic dans les réseaux de capteurs

Dans un réseau de capteurs, différents types de trafic peuvent être rencontrés et sont divisés en trois catégories :

- La première catégorie, qui peut être considérée comme le trafic le plus répandu dans les réseaux de capteurs est le trafic dit convergecast (Multi-point to Point) : les données sont générées par les nœuds dans le réseau et envoyées vers un nœud central du réseau appelé puits (Sink) afin d'y être traitées.
- La deuxième catégorie est celle où le trafic est généré par le puits et envoyé aux nœuds du réseau (Point to Multi-point).
- Et pour finir, la dernière catégorie est la catégorie Point à Point qu'on peut résumer comme l'échange de données entre deux nœuds quelconques du réseau.

Dans ces différentes catégories le trafic peut être de type périodique ou non périodique.

Dans les réseaux de collecte d'informations par exemple, le type de trafic le plus répandu sera le trafic périodique : chaque nœud après mesure périodique, mesure de la température dans une zone par exemple, doit envoyer un paquet.

Un autre type de trafic assez représentatif des réseaux de capteurs est le trafic dit événementiel. Nous le retrouvons par exemple dans des applications liées à la surveillance d'une zone donnée. Les capteurs ne transmettent les données qu'après détection d'un événement, par exemple la détection de fuites dans une centrale nucléaire ou pétrolière. Ce trafic sera sous forme de burst généré à la détection d'un événement.

Le troisième type est le trafic à la demande, où un nœud n'envoie les informations qu'à la demande (réception de requête) : nous pouvons citer par exemple les applications de domotique ou de télé surveillance.

1.3 Domaines d'application des réseaux de capteurs

1.3.1 Environnement

La gestion de l'environnement par l'Homme a de plus en plus recours à des mesures distribuées reposant sur l'usage de capteurs. Les prévisions météorologiques (reposant sur des mesures d'hygrométrie, de pression de l'air) ont été l'un des premiers champs d'application des capteurs. Les mesures de qualité de l'air et de taux de pollution, en ville comme en campagne, se généralisent peu à peu. Les réseaux de capteurs permettent même de pousser la mesure de ce taux vers de nouveaux milieux, comme les glaciers [5] ou bien les océans.

L'agriculture est également susceptible d'avoir recours aux capteurs : des essais sont menés sur la réalisation de mesures faites par des micro-capteurs semés en même temps que les cultures, permettant de surveiller au mieux les conditions de développement de ces dernières. Placés judicieusement dans l'habitat naturel de certaines espèces, les capteurs peuvent être utilisés pour suivre et analyser le comportement de la faune d'un milieu .

En forêt, les réseaux de capteurs peuvent permettre de détecter les départs de feu et de lutter plus efficacement contre les incendies. Et en zones à risques, ils peuvent être utilisés pour mesurer l'activité sismique ou volcanique du sol, et permettre une meilleure anticipation des phénomènes naturels. L'industrie a également recours à ces réseaux pour détecter d'éventuelles fuites de produits toxiques (pétrole, gaz, éléments radioactifs).

Il est à noter que certaines de ces applications touchent à des domaines critiques (en matière de sécurité des installations, voire de vies humaines) : il est indispensable d'assurer le bon fonctionnement du réseau déployé dans un tel contexte.

1.3.2 Surveillance et détection

Les réseaux de capteurs sans fil sont aussi employés pour des mesures de sûreté ou de sécurité, par exemple pour surveiller l'intégrité structurelle de certaines architectures (ferroviaires, aérospatiales, maritimes, ou plus simplement dans le bâtiment : gros œuvre, ouvrages d'art) et peuvent permettre une prévention efficace des pannes matérielles.

Mis à part la détection de pannes, les capteurs peuvent être utilisés dans des systèmes de vidéo-surveillance en milieu urbain, ou sur des sites sensibles afin de prévenir les intrusions humaines, ou bien pour détecter des incidents ou encore établir le suivi de certaines entités ; la gestion du trafic routier, ou même seulement des autobus, en sont des exemples. Le CEREMA (Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement) a mis en ligne un site Internet particulièrement riche en exemples d'application des capteurs, et détaillant notamment les différentes technologies utilisées pour réaliser les mesures [6]. Pour exemple la seule détection de véhicules sur les voies de circulation peut reposer sur :

- la détection d'une modification du champ magnétique environnant ;
- la pression exercée par le véhicule sur la voie ;
- la variation de la pression dans un tube pneumatique ;
- les déformations de la structure d'un pont sous le poids des véhicules ;
- Les variations d'intensité lumineuse ou sonore ;
- le renvoi d'ondes radar, ou lumineuses (laser), ou infrarouges, ou ultrasons ;
- l'analyse d'image vidéo visibles ou infrarouges, par reconnaissance de forme, ou bien par détection du mouvement des groupes de pixels. . .

1.3.3 Domaine militaire

La surveillance et la détection sont aussi très utilisés dans le domaine militaire. Qu'il s'agisse de mener des opérations de renseignement ou bien de suivre l'évolution d'un combat sur le champ de bataille, pour analyser les cibles, relier entre eux fantassins et véhicules de tous types, pour détecter les agents radioactifs, chimiques ou biologiques répandus par un ennemi, toutes les informations sont bonnes à prendre [1]. Un maximum d'entre elles doivent être remontées au centre de commandement, afin de permettre une supervision optimale des forces en mouvement. Plus encore que les exemples précédents, l'usage des capteurs en contexte militaire implique de fortes contraintes quant à la sécurité du réseau déployé.

1.3.4 Domaine médicale

Dans le domaine de la médecine, les réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro-

capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers, etc.). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battements du coeur, etc, à l'aide des capteurs ayant chacun une tâche bien particulière. Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient . D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri, etc.) chez les personnes dépendantes (handicapées ou âgées).

1.3.5 Grand public

Le public a lui aussi de plus en plus accès à des applications reposant sur les capteurs. En domotique par exemple, la température des différentes pièces d'un lieu d'habitation peut être rendue accessible à tout instant, tandis que des fonctions comme l'activation de volets électriques, l'ouverture de portes, l'activation de sources lumineuses ou de dispositifs de chauffage peuvent être effectuées à distance par l'habitant (par exemple par le biais d'une application logicielle sur téléphone portable) ou bien de façon automatique en fonction des besoins identifiés à partir des données récoltées [1].

Un autre domaine d'application en voie de développement est ce que l'on appelle l'Internet des objets (the Internet of things en anglais), et qui consiste en quelque sorte à étendre Internet au monde réel, par le biais d'une interconnexion réseau entre les objets de la vie courante : de plus en plus d'objets du quotidien tendent à devenir « connectés ». Ils se voient alors équipés de capteurs, d'un module de communication sans fil, et offrent de nouvelles possibilités en termes d'usages. Quelques exemples :

- les montres (interfacées avec les téléphones « intelligents » pour afficher des notifications diverses) ;
- les ampoules d'éclairage (gestion de la luminosité, de la couleur de l'éclairage, parfois dynamiques) ;
- les ceintures (accessoires vestimentaires : réajustement automatique du réglage, collecte de données sur le tour de taille pour le contrôle du poids) ;
- les réfrigérateurs (gestion des réserves alimentaires, création de listes d'achat) ;
- les véhicules autonomes, tels que les voitures sans conducteurs ou les drones (pilotage de l'engin, relevé de mesures tactiques pour les drones militaires).

Des puces RFID (de l'anglais Radio Frequency Identification, « identification sur fréquences radio ») se retrouvent par ailleurs de plus en plus utilisées, entre autres raisons pour faciliter de telles interactions dans le monde « connecté » [7]. Au fur et à mesure que ces objets intègrent la vie quotidienne des utilisateurs, ils auront probablement tendance à communiquer de plus en plus entre eux, entre objets, en ne centralisant les données que sur une seule interface présentée à l'utilisateur final.

Les exemples présentés ici ne représentent bien sûr que quelques applications des capteurs parmi les nombreuses qui existent, et dont la quantité ne cesse par ailleurs de croître au fil du temps.

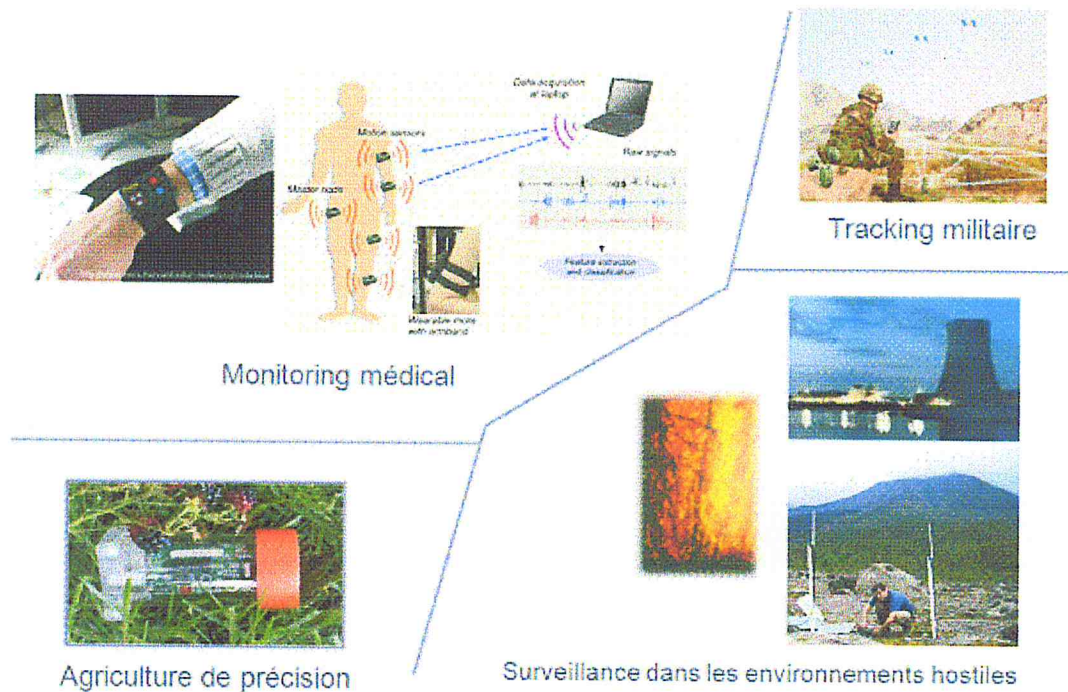


FIGURE 1.4: Domaines d'applications des réseaux de capteurs sans fil

1.4 Les catégories des réseaux sans fil

Il existe plusieurs catégories de réseaux sans fil qui diffèrent par le périmètre géographique qu'ils couvrent ainsi que par les types d'applications supportées. Le schéma suivant illustre les catégories des réseaux sans fil.

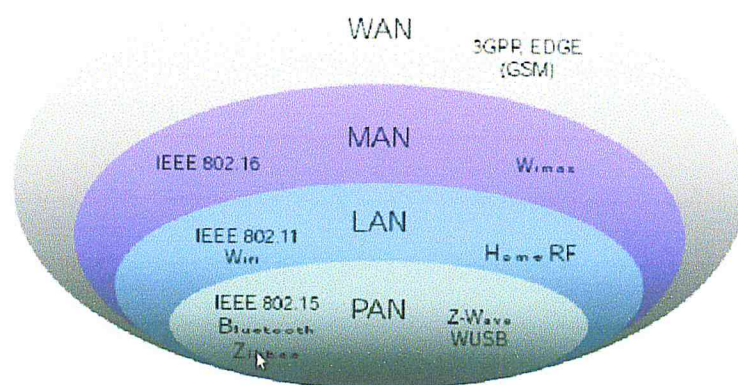


FIGURE 1.5: Les différentes catégories des réseaux sans fil

1.4.1 Le réseau étendu sans fil (WWAN)

Le réseau étendu sans fil (WWAN pour Wireless Wide Area Network) est également connu sous le nom de réseau cellulaire mobile. Il s'agit des réseaux sans fil les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fil tel que GSM, GPRS, UMTS [8].

1.4.2 Réseaux métropolitains sans fil (WMAN)

Le réseau métropolitain sans fil (WMAN pour Wireless Metropolitan Area Network) est connu sous le nom de Boucle Locale Radio (BLR). Les WMAN sont basés sur la norme IEEE 802.16. La boucle locale radio offre un débit utile de 1 à 10 Mbit/s pour une portée de 4 à 10 kilomètres, ce qui destine principalement cette technologie aux opérateurs de télécommunication.

La norme de réseau métropolitain sans fil la plus connue est le WiMAX, permettant d'obtenir des débits de l'ordre de 70 Mbit/s sur un rayon de plusieurs kilomètres.

1.4.3 Réseaux locaux sans fil (WLAN)

Le réseau local sans fil (noté WLAN pour Wireless Local Area Network) est un réseau permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres. Il permet de relier entre-eux les terminaux présents dans la zone de couverture. Il existe plusieurs technologies concurrentes :

- Le Wifi (ou IEEE 802.11) : soutenu par l'alliance WECA (Wireless Ethernet Compatibility Alliance) offre des débits allant jusqu'à 54Mbps sur une distance de plusieurs centaines de mètres.
- La norme européenne élaborée par l'ETSI (European Telecommunications Standards Institute). HiperLAN 2 permet d'obtenir un débit théorique de 54 Mbps sur une zone d'une centaine de mètres dans la gamme de fréquence comprise entre 5 150 et 5 300 MHz [8].

1.4.4 Réseaux personnels sans fil (WPAN)

Le réseau personnel sans fil (appelé également réseau individuel sans fil ou réseau domestique sans fil et noté WPAN pour Wireless Personal Area Network) concerne les réseaux sans fil d'une faible portée : de l'ordre de quelques dizaines mètres. Ce type de réseau sert généralement à relier des périphériques (imprimante, téléphone portable, appareils domestiques, ...) ou un assistant personnel (PDA) à un ordinateur sans liaison filaire ou bien à permettre la liaison sans fil entre deux machines très peu distantes. Il existe plusieurs technologies utilisées pour les WPAN :

- **Le Bluetooth** : ancée par Ericsson en 1994, proposant un débit théorique de 1 Mbps pour une portée maximale d'une trentaine de mètres [8]. Bluetooth, connue aussi sous le nom IEEE 802.15.1, possède l'avantage d'être très peu gourmande en énergie, ce qui la rend particulièrement adaptée à une utilisation au sein de petits périphériques [9].
- **L'infra-rouges** Elles permettent de créer des liaisons sans fil de quelques mètres avec des débits pouvant monter à quelques mégabits par seconde [8]. Cette technologie est largement utilisée dans la domotique (télécommandes), elle souffre toutefois des perturbations dues aux interférences lumineuses.
- **Le Zigbee** La technologie ZigBee (aussi connue sous le nom IEEE 802.15.4) permet d'obtenir des liaisons sans fil à très bas prix et avec une très faible consommation

d'énergie, ce qui la rend particulièrement adaptée pour être directement intégrée dans de petits appareils électroniques (appareils électroménagers, hifi, jouets, ...). La technologie Zigbee, opérant sur la bande de fréquences des 2,4 GHz et sur 16 canaux, permet d'obtenir des débits pouvant atteindre 250 Kb/s avec une portée maximale de 100 mètres environ [10].

1.5 Comparaison des technologies sans fil

Le tableau suivant présente une comparaison entre trois de ces technologies sans fil les plus répandues [11].

Cette comparaison s'appuie sur cinq critères : le nombre de nœuds, l'importance de consommation d'énergie, Le débit, la portée du signal et les besoins de la mémoire disponible.

Protocole	Zigbee	Bluetooth	Wifi
IEEE	802.15.4	802.15.1	802.11 a/b/g/n
Besoins mémoire	4-32kb	250 kb +	1 Mb +
Autonomie avec pile	Années	Jours	Heures
Nombre de noeuds	65 000+	7	32
Vitesse de transfert	250 Kb/s	1 Mb/s	11-54 et + Mb/s
Portée	100 m	10-100 m	300 m

TABLE 1.1: Comparaison des protocoles Zigbee, Bluetooth et Wi-Fi

Le Zigbee est la technologie qui semble offrir les caractéristiques idéales pour un réseau de longue durée. Ce dernier permet entre autres une zone de couverture plus importante que le Bluetooth. Cette technologie offre également la possibilité de gérer un grand nombre de nœuds avec un débit moins important par rapport aux autres technologies sans fil, cela permet une consommation d'énergie plus optimale.

1.6 Topologies de la norme Zigbee - IEEE 802.15.4

La technologie ZigBee prévoit deux types d'entités :

- les entités complètes, ou FFD (Full Function Device) ;
- les entités réduites, ou RFD (Reduce Function Device).

Les FFD implémentent la totalité de la spécification ZigBee alors que les RFD sont des entités allégées dans un objectif de moindre consommation énergétique et de moindre utilisation mémoire pour le microcontrôleur associé.

Les entités RFD sont nécessairement des nœuds terminaux du réseau ; un tel nœud ne saura pas router un paquet sur le réseau. Typiquement, un capteur embarqué sera RFD et alimenté sur batteries, alors qu'une unité centrale de traitement, alimentée par une source non contrainte énergétiquement, sera FFD avec une fonction de coordination du réseau.

La norme IEEE 802.15.4 prévoit trois types de topologies :

1.6.1 La Topologie en Étoile

La topologie en étoile se compose d'un coordinateur et de plusieurs dispositifs d'extrémité (nœuds), comme le montre la figure ci-dessous .

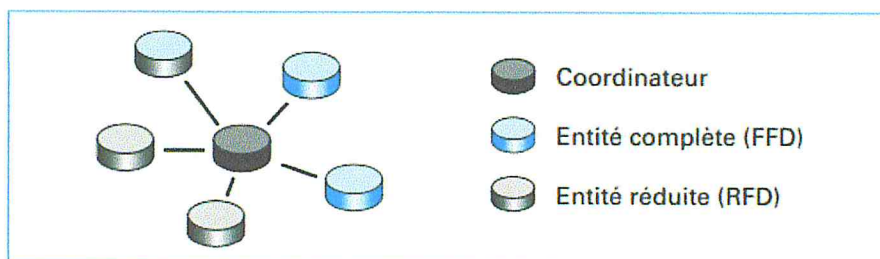


FIGURE 1.6: Représentation de la topologie en étoile

Dans cette topologie, le dispositif d'extrémité communique uniquement avec le coordinateur. Tout échange de paquets entre les dispositifs d'extrémité doit passer par le coordinateur du réseau, et parce que tous les paquets entre les appareils doivent passer par coordinateur, le coordinateur peut devenir un goulot d'étranglement. En outre, il n'y a aucun chemin alternatif de la source vers la destination. L'avantage de la topologie en étoile est qu'il est simple et les paquets passent par au plus deux sauts pour atteindre leur destination.

1.6.2 La Topologie en Mesh

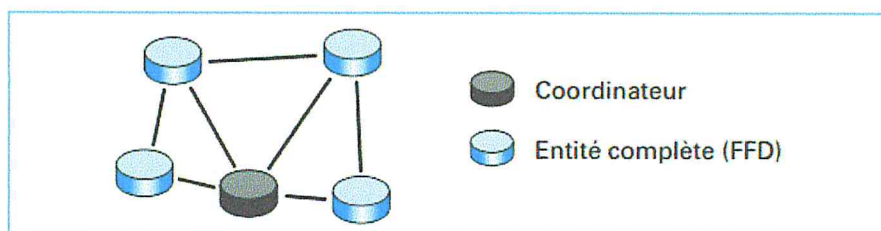


FIGURE 1.7: Représentation de la topologie en mesh

appelée aussi un réseau peer-to-peer, se compose d'un coordinateur, plusieurs routeurs et terminaux, comme le montre la figure. Voici les caractéristiques d'une topologie maillée :

- Une topologie de maillage est un réseau multi-sauts ; les paquets passent par plusieurs sauts pour atteindre leur destination.
- La portée d'un réseau peut être augmentée par l'ajout de plusieurs dispositifs sur le réseau.
- Il peut éliminer les zones mortes.
- Une topologie maillée est auto-guérison, lors de la transmission, si un chemin échoue, le nœud est chargé de trouver un autre chemin vers la destination.
- Les appareils peuvent être proches les uns des autres afin qu'ils consomment moins d'énergie.

1.6.3 La topologie en Arbre

Dans cette topologie, le réseau se compose d'un noeud central (racine de l'arborescence), qui est un coordonnateur, plusieurs routeurs, et des dispositifs d'extrémité .

La fonction du routeur est d'étendre la couverture du réseau. Les noeuds d'extrémité qui sont connectés au coordonnateur ou des routeurs sont appelés enfants. Seuls les routeurs et le coordonnateur peut avoir des enfants. Chaque dispositif d'extrémité est seulement capable de communiquer avec son parent (routeur ou coordonnateur).

Le coordonnateur et les routeurs peuvent avoir des enfants et, par conséquent, sont les seuls appareils qui peuvent être parents.

Un dispositif de fin ne peut pas avoir des enfants et, par conséquent, peuvent ne pas être un parent.

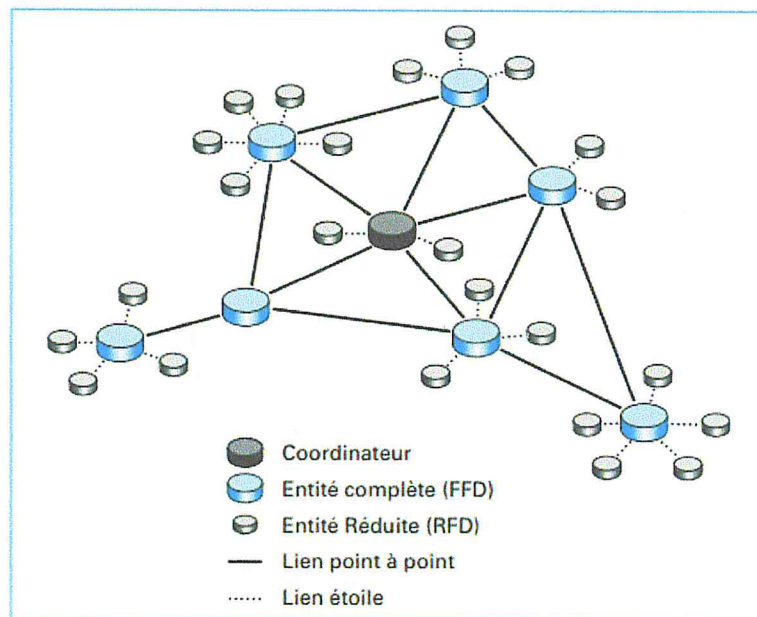


FIGURE 1.8: Représentation de la topologie en arbre

1.7 La pile protocolaire des RCSF

Le rôle de cette pile consiste à standardiser la communication entre les participants afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles.

Ce modèle comprend 5 couches qui ont les mêmes fonctions que celles du modèle OSI ainsi que 3 couches pour la gestion de la puissance, la gestion de la mobilité et la gestion des tâches.

Le but d'un système en couches est de séparer le problème en différentes parties (les couches) selon leur niveau d'abstraction.

Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous). Chaque couche utilise ainsi les services des couches inférieures et en fournit à celle de niveau supérieur.

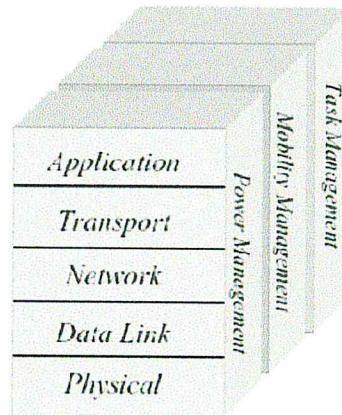


FIGURE 1.9: Pile protocolaire d'une architecture de réseau de capteurs sans fil [1]

1.7.1 La couche physique

S'occupe de la spécification du câblage, des fréquences porteuses, etc.

Cette couche doit assurer des techniques d'émission, de réception et de modulation de données d'une manière robuste.

1.7.2 La couche liaison de données

Elle spécifie comment les données sont expédiées entre deux nœuds/routeurs dans une distance d'un saut. Elle est responsable du multiplexage des données, du contrôle d'erreurs, de l'accès sur le media.

Elle assure la liaison point à point et point à multipoint dans un réseau de communication. Elle est composée de la couche de contrôle de liaison logique (LLC pour Logical Link Control) qui fournit une interface entre la couche liaison et la couche réseau en encapsulant les segments de messages de la couche réseau avec des informations d'entête additionnelles, et la couche de contrôle d'accès au médium (MAC pour Medium Access Control) qui contrôle la radio.

Comme l'environnement des réseaux de capteurs est bruyant et les nœuds peuvent être mobiles, la couche de liaison de données doit garantir une faible consommation d'énergie et minimiser les collisions entre les données diffusées par les nœuds voisins.

1.7.3 La couche réseau

Cette couche permet de gérer l'adressage et le routage des données, c'est-à-dire leur acheminement via le réseau.

1.7.4 La couche transport

Cette couche est chargée du transport des données, de leur découpage en paquets, du contrôle de flux, de la conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission.

1.7.5 La couche application

Cette couche assure l'interface avec les applications. Il s'agit donc du niveau le plus proche des utilisateurs, géré directement par les logiciels.

1.7.6 Le niveau de gestion d'énergie

Les fonctions intégrées à ce niveau consistent à gérer l'énergie consommée par les capteurs. Dès lors, un capteur peut par exemple éteindre son interface de réception dès qu'il reçoit un message d'un nœud voisin afin d'éviter la réception des messages dupliqués. De plus, quand un nœud possède un niveau d'énergie faible, il peut diffuser un message aux autres capteurs pour ne pas participer aux tâches de routage, et conserver l'énergie restante aux fonctionnalités de capture [1].

1.7.7 Le niveau de gestion de mobilité

Ce niveau détecte et enregistre tous les mouvements des nœuds capteurs, de manière à leur permettre de garder continuellement une route vers l'utilisateur final, et maintenir une image récente sur les nœuds voisins. Cette image est nécessaire pour pouvoir équilibrer l'exécution des tâches et la consommation d'énergie [1].

1.7.8 Le niveau de gestion des tâches

Lors d'une opération de capture dans une région donnée, les nœuds composant le réseau ne doivent pas obligatoirement travailler avec le même rythme. Cela dépend essentiellement de la nature du capteur, son niveau d'énergie et la région dans laquelle il a été déployé. Pour cela, le niveau de gestion des tâches assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau. [1]

1.8 Conclusion

A travers ce chapitre nous avons commencé par présenter les réseaux de capteurs sans fil, l'architecture des capteurs ainsi que les domaines d'applications des RCSF.

Ensuite nous avons mené une étude comparative sur les différentes technologies sans fil. Enfin, nous avons mentionné quelques unes des attaques les plus connues sur les RCSF.

Dans le chapitre suivant, nous entamons la partie de configuration et analyse du réseau .

Chapitre 2

Configuration et analyse du réseau de capteurs sans fil

2.1 Introduction

Dans ce chapitre, nous détaillons le fonctionnement du réseau que nous avons développé. Nous commençons dans la première partie par présenter les capteurs et les plate-formes utilisées dans le développement. Par la suite, nous exprimons une vue globale du réseau. Enfin, nous présenterons les différentes architectures et solutions proposées et conçus ainsi que la configuration des capteurs du réseau.

2.2 Outils de développement

2.2.1 Le Kit Professionnel Advanticsys

Le Kit Professionnel *Advanticsys* est un puissant ensemble de nœuds et de capteurs IEEE 802.15.4 pour des déploiements de taille moyenne (Telosb, XM1000, CM4000 et CM5000). Le Kit Professionnel permet à l'utilisateur de mettre en oeuvre des configurations de réseaux complexes et offre la possibilité de l'acquisition de données à partir de l'environnement [12]. Il comprend également la passerelle SG1000, capable de faire face à un grand nombre de nœuds et de configurer facilement les nœuds et le réseau lui-même.



FIGURE 2.1: Le Kit Professionnel Advanticsys

Le kit professionnel ADVANTICSYS fournit une large gamme de modules sans fil afin de répondre aux différents besoins des réseaux sans fil IEEE 802.15.4 [12].

2.2.2 Capteur Telosb



FIGURE 2.2: La plate-forme Telosb

MEMSIC TelosB TPR 2420 est une plate-forme open-source conçue pour permettre l'expérimentation de pointe pour la communauté de recherche. le TPR 2420 regroupe tous les éléments essentiels pour des études de laboratoire en une seule plate-forme notamment : la capacité de programmation USB, une radio IEEE 802.15.4 avec antenne intégrée, une faible consommation d'énergie MCU (microcontrôleur) avec mémoire étendue et une extension de capteur optionnel. TPR 2420 offre de nombreuses fonctionnalités, y compris [13] :

Processeur	Sensors	Portée
MSP430 microcontroller 8 Mhz	Lumière Température	Datasheet : 30m(indoor) / 100m(outdoor)
48k Flash (ROM) 10K (RAM)	Humidité Voltage	
Transmission	Os	
Norme de transmission : IEEE 802.15.4 (Zigbee) Débit : 250 Kbps (Bande 2.4-2.4835 Ghz) Antenne Intégrée	TinyOs	

TABLE 2.1: Les propriétés du Telosb de MEMSIC

2.2.3 Capteur XM1000

Le XM1000 est une nouvelle génération de capteurs basée sur les spécifications techniques de l'architecture "TelosB", avec mise à niveau de 116kb-EEPROM et 8Kb-RAM avec intégration des sensors : température, humidité , lumière et radiations IR. Le capteur

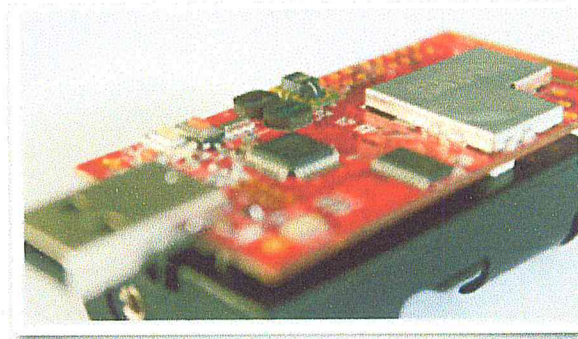


FIGURE 2.3: La Plate-forme XM1000

présente les caractéristiques générales suivantes [14] :

Processeur	Sensors	Portée
MSP430F1611 microcontroller 8 Mhz	Lumière	Datasheet : 30m(indoor) / 100m(outdoor)
116k Flash (ROM) 8K (RAM)	Température	
	Humidité Voltage	
Transmission		Os
Norme de transmission : IEEE 802.15.4 (Zigbee) Débit : 250 Kbps (Bande 2.4-2.4835 Ghz) Antenne Intégrée		TinyOs

TABLE 2.2: Les propriétés du XM1000

2.2.4 Capteur CM5000

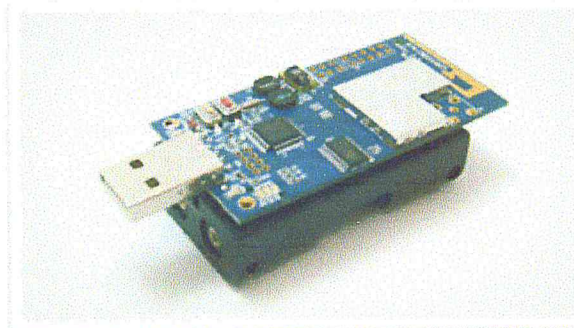


FIGURE 2.4: La plate-forme CM5000

La plate-forme CM5000 est un nœud capteurs sans fil compatible avec la norme IEEE 802.15.4 basée sur l'open-source "TelosB" développé et publié par l'Université de

Californie, Berkeley («UC Berkeley»). Le capteur présente les caractéristiques générales suivantes [15] :

Processeur	Sensors	Portée
MSP430F1611 microcontroller 8 Mhz 48k Flash (ROM) 10K (RAM)	Lumière Température Humidité Voltage	Datasheet : 40m(indoor) / 125m(outdoor)
Transmission	Os	
Norme de transmission : IEEE 802.15.4 (Zigbee) Débit : 250 Kbps (Bande 2.4-2.4835 Ghz) Antenne Intégrée	TinyOs	

TABLE 2.3: Les propriétés du Telosb du CM5000

2.2.5 Capteur CM4000

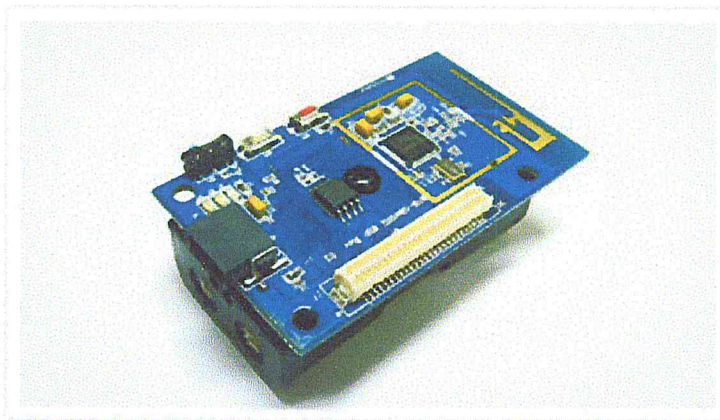


FIGURE 2.5: La plate-forme CM4000

La plate-forme CM4000 est un nœud capteurs sans fil compatible avec la norme IEEE 802.15.4 basée sur l'open-source "TelosB" développé et publié par l'Université de Californie, Berkeley («UC Berkeley»). Le capteur présente les caractéristiques générales suivantes [16] :

Processeur	Sensors	Portée
MSP430F1611 microcontroller 8 Mhz 48k Flash (ROM) 10K (RAM)	Lumière Température Humidité Voltage	Datasheet : 40m(indoor) / 125m(outdoor)

Transmission	Os
Norme de transmission : IEEE 802.15.4 (Zigbee) Débit : 250 Kbps (Bande 2.4-2.4835 Ghz) Antenne Intégrée	TinyOs

TABLE 2.4: Les propriétés du Telosb du CM4000

Le CM4000 présente également un design compact dans lequel le connecteur USB traditionnel est remplacé par un connecteur ERNI compatible avec les modules d'interface. Cette architecture permet l'utilisation d'une vaste variété de sensors amovibles.

2.2.6 Le Bridge SG1000

Le SG1000 est un dispositif de passerelle "Bridge" 802.15.4 -Ethernet qui agit comme concentrateur de données pour notre réseau de capteurs sans fil. Son architecture robuste est capable de traiter des données provenant d'un grand nombre de noeuds de capteurs sans fil, la seule limite étant la mémoire physique du système. Ce bridge présente les caractéristiques suivantes :

- 2.4GHz IEEE 802.15.4 -Ethernet Pont
- 4 ports pour la re-programmation des noeuds capteurs USB.
- Base de données MySQL interne pour stocker les données brutes des capteurs.
- Interface web Compréhensible.



FIGURE 2.6: La passerelle SG1000 - Advanticsys Technologie

Le SG1000 comprend une interface IEEE 802.15.4 , agissant en tant que puits pour les noeuds capteurs sans fil déployés et permettant une communication bidirectionnelle avec le réseau sans fil. Ce dernier permet d'accéder aux données récupérées par le biais d'une connexion IP.

En outre, la base de données MySQL intégrée permet le stockage local de ces données, ce qui permet l'accès à ces données par des applications externes grâce à l'interface Ethernet configurable [12].

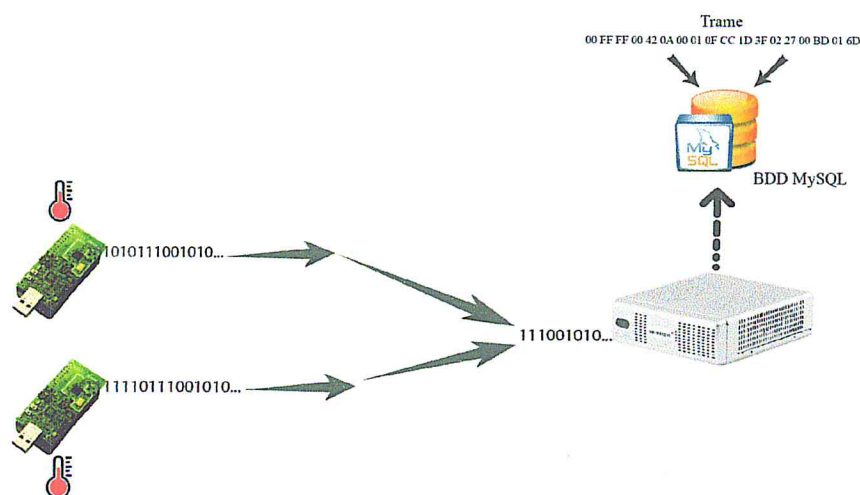


FIGURE 2.7: Procédure d'enregistrement des données

Ces données accessibles par le biais d'une connexion IP a travers le bridge SG1000 sont représentées par des trames [12] comme le montre la figure suivante :

```

00 FF FF 00 42 0A 00 01 0F CC 1D 3F 02 27 00 BD 01 6D
00 FF FF 00 42 0A 00 01 0F CB 1D 3F 02 16 01 40 02 0E
00 FF FF 00 42 0A 00 01 0F CD 1D 3C 02 19 01 55 01 C3
00 FF FF 00 42 0A 00 01 0F CC 1D 3F 02 1A 00 AB 01 3F
00 FF FF 00 42 0A 00 01 0F CB 1D 3F 02 1C 01 2B 01 F5
00 FF FF 00 42 0A 00 01 0F CC 1D 3E 02 1C 01 51 01 AF
00 FF FF 00 42 0A 00 01 0F CC 1D 3D 02 27 00 C4 01 78
00 FF FF 00 42 0A 00 01 0F CC 1D 3D 02 27 01 17 01 E4
00 FF FF 00 42 0A 00 01 0F CC 1D 3F 02 28 01 5D 01 EE
00 FF FF 00 42 0A 00 01 0F CD 1D 3E 02 22 00 FA 01 18

```

FIGURE 2.8: Représentation des trames du SG1000

A présent, nous présentons le modèle relationnel de la table Trame se trouvant dans la base de données MySQL au niveau du bridge SG1000 [12].

Trame (**ID**, Temps, Taille, donnée)

2.2.7 Décomposition analytique des trames

Prenons l'exemple suivant :

00 FF FF 00 42 0A 00 01 0F CC 1D 3F 02 27 00 BD 01 6D

Tout d'abord, nous analysons l'en-tête :

- 00 : Cet octet indique que c'est un message AM.
- FF FF : Ces deux octets sont l'adresse de destination (FF FF indique la diffusion - Broadcast).
- 00 42 : L'adresse source, ce qui indique l'adresse du dernier noeud transmetteur.
- 0A : Longueur en octets du paquet utile, 0A indique que la charge utile du message est de 10 octets.
- 00 : L'identifiant du groupe.
- 01 : Permet d'identifier le type de message reçu.

la suite de la trame représente les mesures physiques de l'environnement.

- 0F CC : La valeur du voltage interne du capteur.
- 1D 3F : La valeur brute de la température.
- 02 27 : La valeur brute de l'humidité.
- 00 BD : Le taux de luminosité.
- 01 6D : Le taux des radiations infra-rouges.

2.2.8 Décodage des trames

Les données brutes reçues doivent être analysées et converties de leur format en HEX en valeurs réelles [12]. L'application chargée d'exploiter la base de données du bridge SG1000 décode les données brutes de la façon suivante :

— **Le voltage interne du micro-contrôleur**

$$Tension = valeur_brute / 4096 * 1.5V$$

— **L'Humidité**

$$RH = (c2 + (c3 * result)) * result + c1$$

avec les constantes :

$$c1 = -4$$

$$c2 = 0.0405;$$

$$c3 = -2.8 \exp(-6);$$

— **La température**

$$RHtc = (T - 25) * (t1 + t2 * result) + RH$$

Telque :

$$t1 = 0.01$$

$$t2 = 0.00008$$

$$T = -39.6 + t1$$

RH : la valeur de l'humidité décryptée précédemment.

$result$: la valeur brute du capteur.

— **La luminosité et radiations IR**

$$Lumi = 0.625 * 106 * I * 1000$$

$$Rad_IR = 0.769 * 105 * I * 1000$$

Telque :

$$I = Tension / 100000$$

2.3 Conception du réseau

Pour la réalisation du réseau, nous présentons une vue globale de l'ensemble des différentes parties constituant notre réseau.

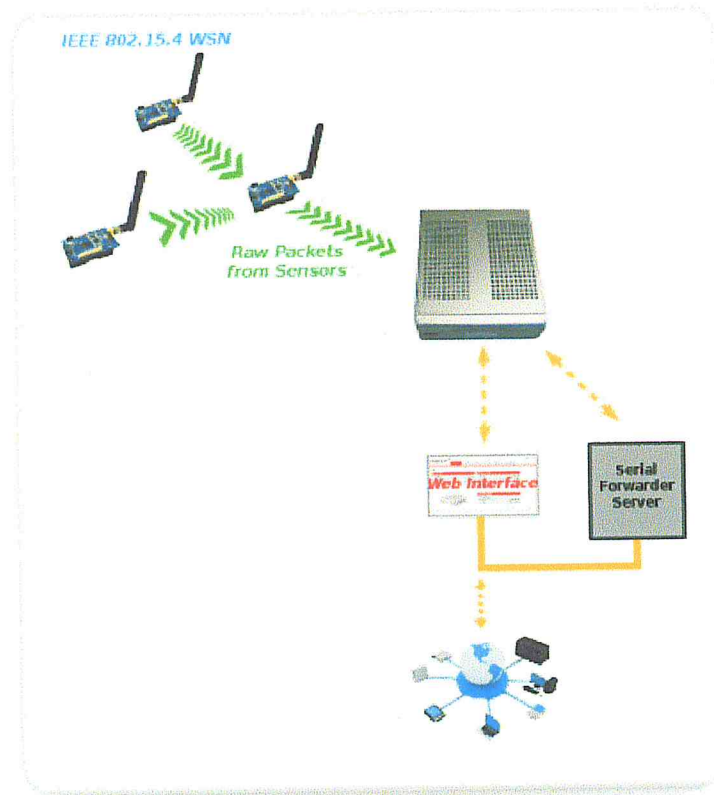


FIGURE 2.9: Représentation globale du réseau de capteur sans fil

Notre réseau est constitué de deux parties principales :

- Une première partie qui est représentée par le réseau de capteurs sans fil qui est chargé de la collecte des mesures physiques.
- Une deuxième partie qui est constituée de l'application qui est chargée de l'exploitation de la base de données.

2.4 Configuration des architectures du réseau

Dans les réseaux de capteurs, la communication de l'information à la station de base peut se faire de différentes manières selon l'architecture utilisée.

2.4.1 Architecture en étoile

Une réseau en étoile est un réseau dans lequel tous les nœuds sont directement reliés à une base centrale (station de base). Chaque nœud communique directement avec la base centrale sans nœuds intermédiaires.

Dans un réseau en étoile, il est souvent facile d'ajouter ou de supprimer des nœuds. Chaque entité est reliée à un bridge central dans une configuration en étoile. Les données sont transmises directement d'une entité au bridge central qui gère tout acheminement de données dans le réseau.

La figure suivante illustre cette première solution :

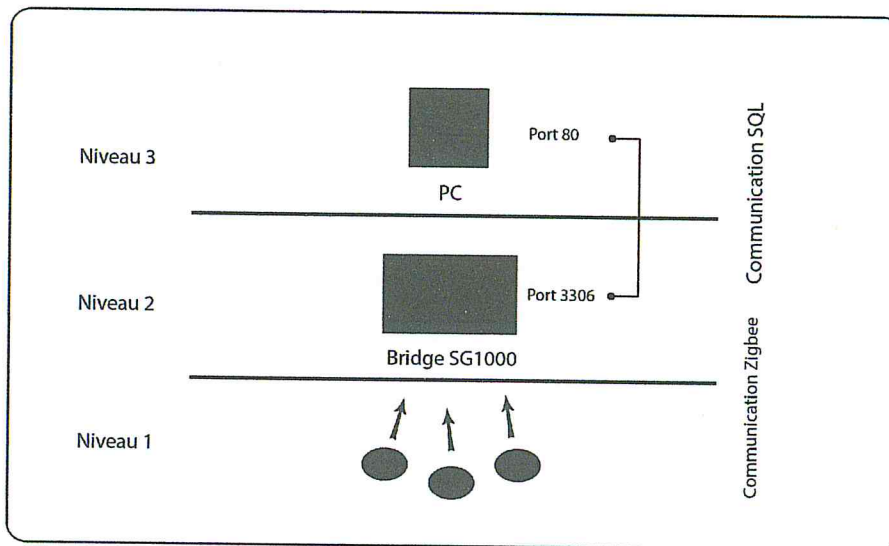


FIGURE 2.10: Architecture en étoile

L'inconvénient majeur de cette topologie provient du fait que lorsque le bridge central est panne, l'ensemble du réseau ne peut plus fonctionner.

Notre choix s'est porté en premier lieu sur cette architecture car elle offre les avantages suivants :

- Chaque liaison est indépendante des autres.
- L'évolution du réseau ne nécessite pas de modifications du réseau existant. Il suffit donc seulement de rajouter des nœuds au réseau.
- Les modifications (ajout des nœuds) sont rapides et peu coûteuses.
- Implémentation et maintenance logicielle et matérielle plus facile .

Les étapes suivantes expliquent le fonctionnement de cette architecture :

- Les capteurs qui composent le réseau ont le même rôle : chacun d'eux mesure périodiquement la température, l'humidité, etc. Et la communique par la suite au bridge SG1000.
- Le bridge reçoit les données des nœuds capteurs et les enregistre dans une base de données.
- L'application au niveau du PC récupère les données enregistrées par le bridge, les décrypte et les affiche à l'utilisateur.

2.4.2 Architecture en multi-saut

Un réseau sans fil multi-saut est un ensemble d'entités mobiles et/ou fixes inter-connectées par une technologie sans fil formant un réseau dynamique temporaire avec ou sans l'aide de toute administration centralisée ou de tout support fixe. Ces réseaux prennent une place de plus en plus importante au vu du nombre croissant d'applications disponibles sur ces réseaux.

Multi-saut - Version 1

Une communication multi-saut dans laquelle deux nœuds distants souhaitent échanger des données doivent utiliser plusieurs sauts. Des nœuds intermédiaires relaient donc les paquets de la source à la destination.

Cette version du multi-saut présente un avantage majeur qui consiste à couvrir une zone géographique plus importante.

Cette architecture s'articule autour de certains capteurs intermédiaires qui jouent le rôle des relais entre les nœuds capteurs et le bridge SG1000.

Son fonctionnement se résume comme suit :

- Les capteurs qui composent le réseau ont le même rôle : chacun d'eux mesure périodiquement la température, l'humidité, etc. Et la communique par la suite au relai.
- Le relai reçoit le paquet des mesures et les transmet à son tour à un autre relai jusqu'au bridge.
- Le bridge reçoit les données des relais et les enregistre dans une base de données.
- L'application au niveau du PC récupère les données enregistrées par le bridge, les décrypte et les affiche à l'utilisateur.

La figure suivante illustre les différents niveaux de cette solution.

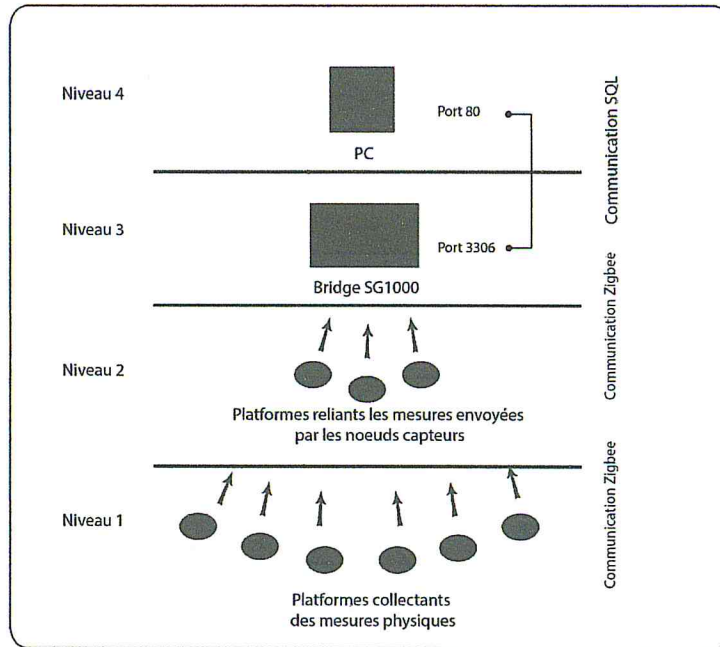


FIGURE 2.11: Architecture en multi-saut à 4 niveaux - v1

Multi-saut - Version 2

La deuxième version de l'architecture proposée précédemment consiste à exploiter les capteurs intermédiaires au maximum en les programmant de façon à ce qu'ils puissent gérer les fonctions entre la collecte et le renvoi des données.

L'inconvénient majeur de cette architecture se manifeste à travers une consommation excessive d'énergie. En effet, les nœuds se trouvant au deuxième niveau exploitent le support radio entre envois des mesures internes et réception et renvoi des données externes. Une telle performance signifie un effort plus important du capteur. Cela implique une consommation plus importante d'énergie.

Cette deuxième version du multi-saut présente par contre une meilleure performance et une exploitation efficace du matériel. Son fonctionnement se résume comme suit :

- Les capteurs mesurent périodiquement la température, l'humidité, etc. Et la communique par la suite au capteur suivant.
- Le capteur reçoit le paquet des mesures et les transmet à son tour à un autre relais jusqu'au capteur suivant.
- Le bridge reçoit les données des nœuds et les enregistre dans une base de données.
- L'application au niveau du PC récupère les données enregistrées par le bridge, les décrypte et les affiche à l'utilisateur.

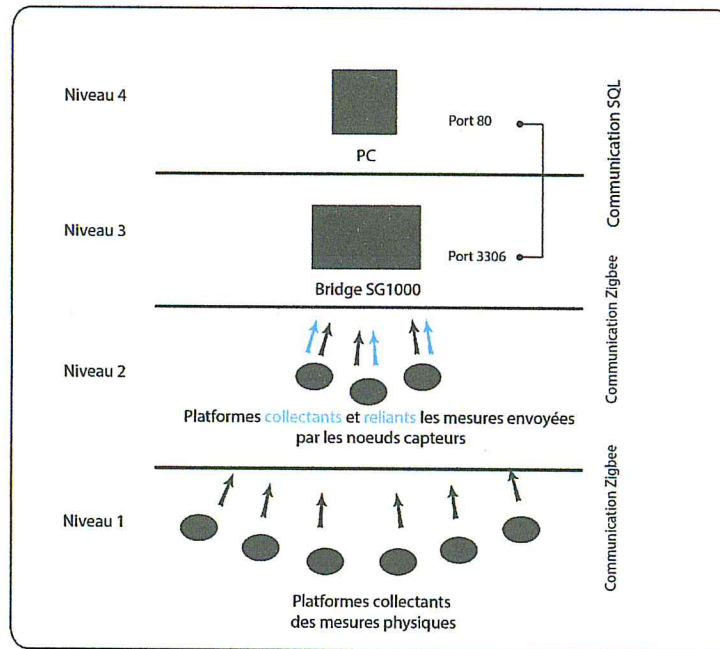


FIGURE 2.12: Architecture en multi-saut à 4 niveaux - v2

2.5 Configuration des noeuds

2.5.1 Choix du langage de programmation NesC

Pour programmer ces capteurs, notre choix s'est porté sur une variante du langage C connu sous le nom "NesC" (voir l'annexe). Ce langage utilise une architecture basée sur des composants, celle-ci vise à réduire la taille mémoire du système et les applications développées. Une application est considérée comme un ensemble de composants ayant un but précis. Chaque composant correspond à un élément matériel (LEDs, timer, ADC, etc) et peut être réutilisé dans différentes applications.

Un composant est constitué de trois éléments essentiels :

Les interfaces : Spécifient un ensemble de fonctions à mettre en application par le fournisseur des interfaces (commandes) et par l'utilisateur des interfaces (événements). Ces fonctions sont précédées par des mots-clés respectifs "command" ou "event". L'utilisation des mots-clés "use" et "provide" au début d'un composant permet de savoir respectivement si celui-ci fait appel à une fonction de l'interface ou redéfinit son code. De plus, tous les composants possèdent l'interface StdControl car sa tâche est l'initialisation, le démarrage et l'arrêt des composants.

Les modules : Définissent les éléments de base de la programmation. Ils utilisent une ou plusieurs interfaces. Par ailleurs, il est à noter que le processus d'exécution repose sur les tâches et les mécanismes d'interruption. De ce fait, les modules permettent aussi d'implémenter ces tâches.

Les configurations : Constituent un ensemble de modules et d'interfaces ainsi que des liaisons entre les composants de l'application déployée dans les capteurs.

2.5.2 Les Nœuds capteurs

Le fonctionnement de la partie embarquée sur les nœuds capteurs est relativement simple :

- Au démarrage du capteur, un temporisateur périodique est lancé suite à l'écoulement d'un certain laps de temps. Ce temporisateur actionne les procédures dédiées à la collecte des mesures physique à partir de la zone de déploiement des capteurs. Ces mesures sont envoyées par la suite vers une station de base (bridge sg1000) dédiée au stockage (Bridge sg1000) sous forme de trames.

la figure suivante illustre le processus effectué par les noeuds capteur .

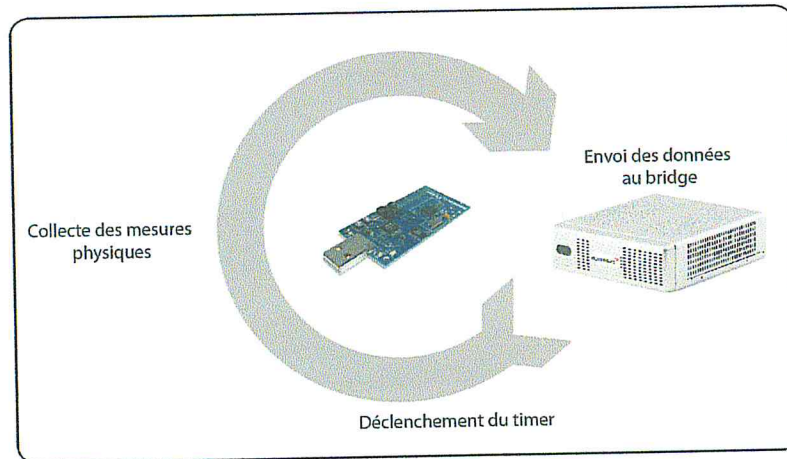


FIGURE 2.13: Interaction nœud capteur-bridge SG1000

fonction de lecture

Pour récupérer les paramètres de l'environnement (Température par exemple), le capteur déclenche une fonction de lecture à travers l'événement *Temperature.readDone()*.

les données lues sont placées dans un paquet « data » qui sera envoyé par la suite via Radio. La figure suivante montre la fonction qui permet de mesurer la température :

```
event void Temperature.readDone(error_t result, uint16_t value) {  
    data.temperature = value; // remplissage du packet  
    if (++numsensors == MAX_SENSORS) {  
        call RadioControl.start(); // appeler l'interface Radio  
        si c'est le dernier sensor  
    }  
}
```

FIGURE 2.14: La fonction *Temperature.readDone()*

Par la suite, la tâche *sendThlMsg()* est responsable de faire un transtypage (cast) du paquet data vers un type prédéfinie *message_t*.

```

task void sendThlMsg() {

    THL_msg_t* aux = (THL_msg_t*)
    call Packet.getPayload(&auxmsg, sizeof(THL_msg_t));

        aux -> vref          = data.vref;
        aux -> temperature = data.temperature;
        aux -> humidity     = data.humidity;
        aux -> photo        = data.photo;
        aux -> radiation    = data.radiation;
        aux -> id = 0xAAA3;

    if (call ThlSend.send(AM_BROADCAST_ADDR,
    &auxmsg, sizeof(THL_msg_t)) != SUCCESS) {

        post sendThlMsg();
    }
}

```

FIGURE 2.15: Transtypage des données

L'interface Radio - émetteur

l'interface Radio permet l'envoi des paquets de type "message_t".
la figure suivante montre la fonction qui permet d'activer l'interface Radio :

```

event void RadioControl.startDone(error_t err) {
    if (err == SUCCESS) { // Radio On avec succès
        call Leds.led2On();
        post sendThlMsg(); // envoi du message
    }else {
        call RadioControl.start();
    }
}

```

FIGURE 2.16: Déclenchement de l'interface radio au niveau du capteur

2.5.3 Les Nœuds relais

Un nœud relai est un système, une organisation ou un nœud qui fait le lien entre deux ou plusieurs agents partageant le même objectif. Ce type de système est utilisé principalement dans les réseaux sans fil comme moyen pour étendre la zone de couverture du réseau. Dans ce qui suit nous présentons les parties principales du programme embarqué sur les relais.

La réception des données envoyées par les nœuds capteurs se fait grâce à l'évènement "Receive.receive()".

Les instructions suivantes permettent de comprendre le processus permettant le transtypage et le renvoi des paquets.

```
event message_t* Receive.receive(message_t* msg, void* payload, uint8_t len){
    if (busy!=FALSE){
        if (len == sizeof(THL_msg_t)) {
            call Leds.led0On();
            busy = TRUE; // radio occupée
            btrpkt2 = (THL_msg_t*)(call Packet.getPayload(&pkt, sizeof(THL_msg_t)));
                // packet a envoyé
            btrpkt3 = (THL_msg_t*)payload; // packet recu
                //transtypage du paquet reçu
            btrpkt2 -> vref      = btrpkt3 ->vref;
            btrpkt2 -> temperature = btrpkt3 -> temperature ;
            btrpkt2 -> humidity    = btrpkt3 -> humidity ;
            btrpkt2 -> photo       = btrpkt3 -> photo ;
            btrpkt2 -> radiation   = btrpkt3 -> radiation ;
            btrpkt2 -> id         = btrpkt3 -> id;
                //renvoi du paquet
            if (call AMSend.send(AM_BROADCAST_ADDR,&pkt, sizeof(THL_msg_t)) == SUCCESS){
                busy = FALSE; // libérer le canal radio
                call Leds.led0Off();
            }
        }
        call Leds.led1Toggle();
    }
    return msg;
}
```

FIGURE 2.17: Réception , transtypage et renvoi des données

2.6 Conclusion

Dans ce chapitre, nous avons présenté les outils de développement utilisés (Capteurs, Kit Pro Advanticsys...) ainsi que les différentes architectures que nous proposons pour notre projet. Nous avons également présenté la configuration des capteurs destinés à la collecte des mesures et les relais.

Dans le chapitre suivant, nous allons aborder la partie de conception et de réalisation de notre application de surveillance.

Chapitre 3

Conception et réalisation de l'application de surveillance

3.1 Introduction

Au niveau de ce chapitre, nous présenterons, de manière abstraite, la méthode conceptuelle avec laquelle nous avons entrepris notre application sur le plan structurelle et technique. Ceci afin de faciliter sa compréhension et démontrer son homogénéité. Nous décrirons aussi toutes les étapes de modélisation ainsi que tous les acteurs impliqués.

Il est à noter que la méthode pour laquelle nous avons orienté notre conception repose sur le langage de référence Unified Modeling Language (UML).

Dans cette section, nous allons apporter plus de détails à notre solution en clarifiant les aspects techniques. Pour ce faire, nous avons utilisé UML (Unified Modeling Language) qui représente un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes. Ces diagrammes donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc. Dans notre travail, nous avons utilisé les diagrammes de cas d'utilisation, de classe et de séquences.

3.2 Analyse et spécification des besoins

La spécification de besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les besoins de notre application. Nous distinguons des besoins fonctionnels qui présentent les fonctionnalités attendues de notre application et les besoins non fonctionnels pour éviter le développement d'une application non satisfaisante ainsi de trouver un accord commun entre les spécialistes et les utilisateurs pour réussir le projet.

3.2.1 Spécification des besoins fonctionnels

Après une étude détaillée de système, cette partie est réservée à la description des exigences fonctionnelles des différents acteurs de l'application. Ces besoins se regroupent dans les diagrammes des cas d'utilisation.

Cette application doit couvrir principalement les besoins fonctionnels suivants :

- Gestion des utilisateurs et des privilèges.
- Gestion des sites de déploiement.
- Gestion et affectation des noeuds capteurs.
- Visualisation des données (Température, Humidité etc...) collectées par les noeuds capteurs en temps réel.
- Configuration et paramétrage des différents seuils.
- Visualisation graphique de l'historique des trames se trouvant au niveau du bridge SG1000.

3.2.2 Spécification des besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et ce qui concerne notre application, nous avons dégagé le besoins suivants :

- La convivialité de l'interface graphique : l'application doit fournir une interface conviviale et simple pour tout type d'utilisateur car elle présente le premier contact de l'utilisateur avec l'application et par le biais de celle-ci on découvrira ses fonctionnalités.
- La fiabilité : les données fournies par l'application doivent être fiables.
- La sécurité : l'application doit respecter la confidentialité des données.
- L'intégrité : l'application doit garantir l'intégrité et la cohérence des données à chaque mise à jour.

3.3 Conception du système

3.3.1 Diagramme de cas d'utilisation (use cases)

Les diagrammes des cas d'utilisation représentent les interactions entre l'application et les acteurs externes. Ces diagrammes permettent aux utilisateurs d'exprimer la manière dont ils voudraient interagir avec l'application et le type d'informations [17] qu'ils veulent échanger dans l'application.

Un diagramme de cas d'utilisation contient les acteurs et les cas d'utilisations. Des relations peuvent exister entre ces derniers tels que la relation d'extension, les relations d'utilisation et les relations d'inclusion.

Les acteurs représentent un rôle joué par une personne ou un élément qui interagit avec l'application.

Acteur	Rôle
Administrateur	Il est l'utilisateur privilégié du système. Son rôle est, superviseur du système d'information dans sa globalité, il fédère les autorisations d'accès et met en place toute la procédure liée à la sécurité des données.
Observateur	L'un des acteurs principaux qui interagissent avec le système. L'observateur est autorisé à contrôler la variation de mesures captées par le réseau des nœuds de capteurs.

TABLE 3.1: Identification des acteurs

Description des cas d'utilisation

Cas d'utilisation 'Ajouter Nœud'

Pour l'ajout d'un nouvel nœud, l'administrateur l'enregistre dans la base de données. Il se doit donc de créer une fiche :

- a. L'administrateur doit demander au système une nouvelle fiche (vierge).
- b. Le système lui demande de la remplir (l'administrateur remplit les champs).
- c. L'administrateur valide les informations saisies.

Cas d'utilisation 'Réglage des Seuils'

Quand un Administrateur veut saisir de nouveaux seuils. Il se doit donc de :

- a. Demander au système une nouvelle fiche de Réglage.
- b. Le système lui demande de remplir les champs.
- c. L'administrateur valide la fiche remplie.
- d. Le Système vérifie les valeurs saisies.
- e. L'administrateur confirme l'enregistrement.

Cas d'utilisation 'Affichage'

Quand un Observateur veut afficher la température. Il se doit donc de faire :

- a. Rechercher le site (zone) à afficher.
- b. Le système lit le message envoyé par le Bridge (SG1000).
- c. Le système extrait les données à partir du message récupéré.
- d. Le système trace la courbe et affiche la valeur numérique de la température.

Diagrammes de cas d'utilisation

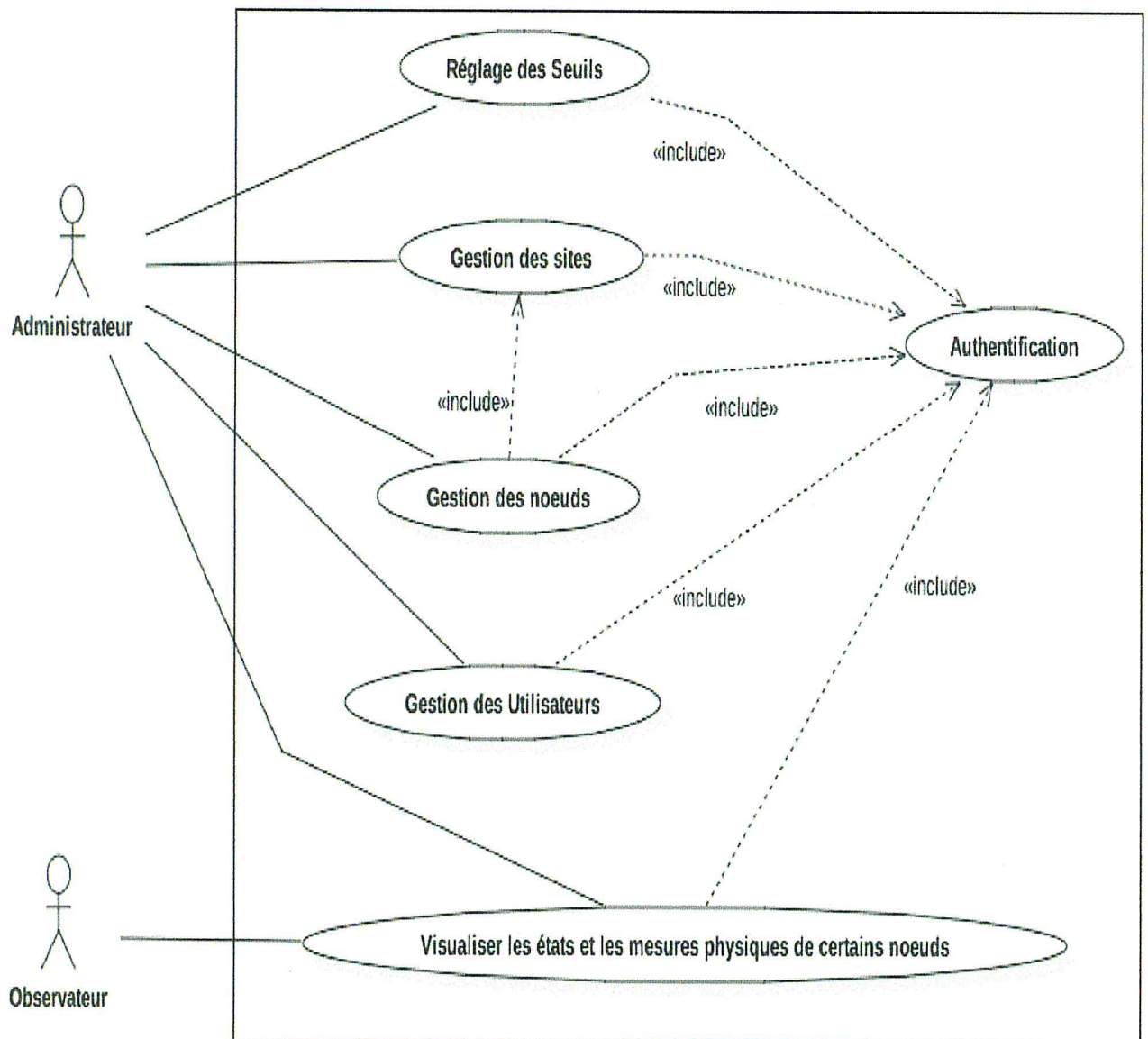


FIGURE 3.1: Diagramme de cas d'utilisation

Diagramme de cas d'utilisation : Gestion des utilisateurs

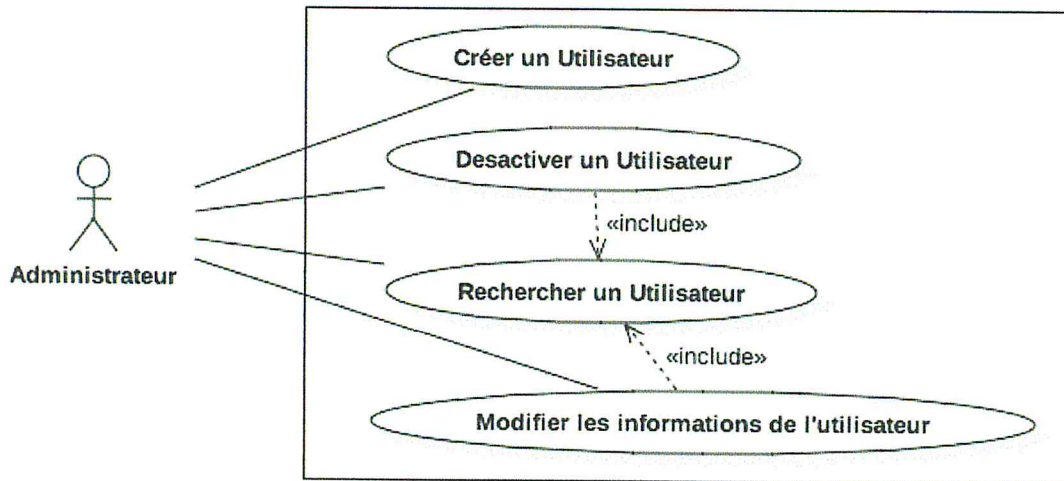


FIGURE 3.2: Diagramme de cas d'utilisation : gestion des utilisateurs

L'administrateur attribue un profil (nom d'utilisateur, mot de passe) à chaque nouvel utilisateur enregistré. Il peut aussi modifier des utilisateurs existants (informations personnelles, droits d'accès) ou supprimer les profils de certains utilisateurs (inactifs).

Diagramme de cas d'utilisation : Gestion des sites

Les sites sont gérés par l'administrateur pour représenter les différentes zones de l'environnement.

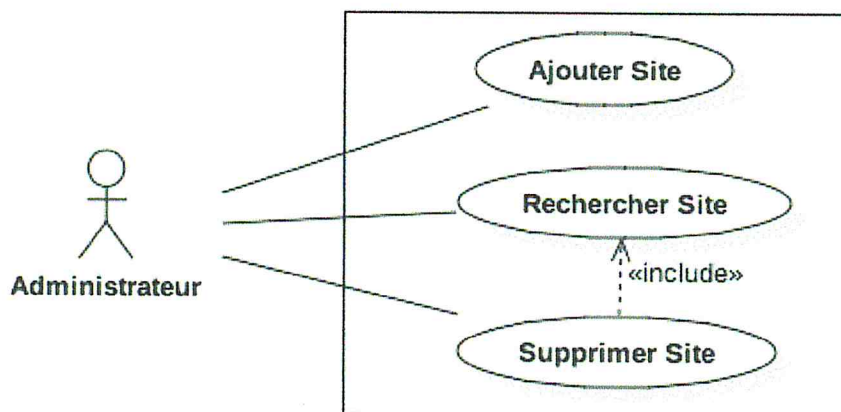


FIGURE 3.3: Diagramme de cas d'utilisation : gestion des sites

Diagramme de cas d'utilisation : Gestion des nœuds

Un Nœud est ajouté, modifié ou supprimé par un administrateur pour assurer la surveillance de l'environnement.

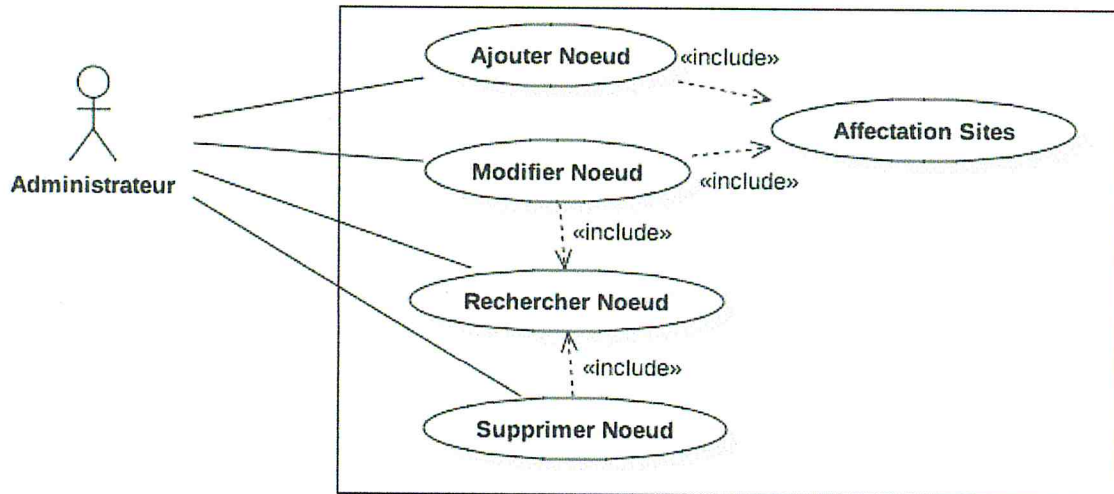


FIGURE 3.4: Diagramme de cas d'utilisation : gestion des nœuds

Diagramme de cas d'utilisation : Réglage des Seuils

L'administrateur attribue un réglage des seuils pour les Nœud affectés.

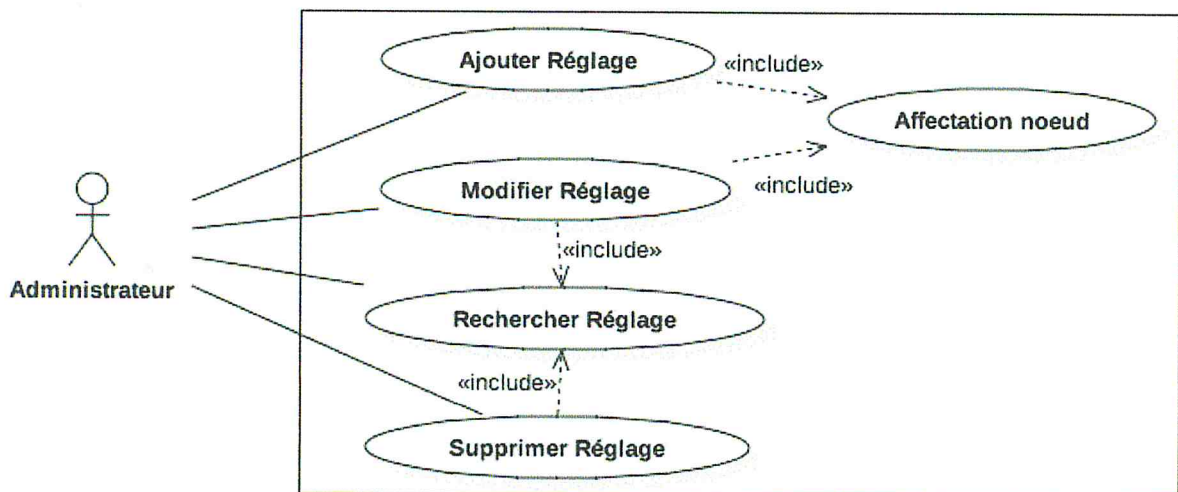


FIGURE 3.5: Diagramme de cas d'utilisation : Réglage des Seuils

3.3.2 Diagrammes de séquence

Les diagrammes des séquences documentent les interactions à mettre en œuvre entre les classes pour réaliser un résultat, tel qu'un cas d'utilisation [18]. UML étant conçu pour la programmation orientée objet, ces communications entre les classes sont reconnues comme des messages. Le diagramme des séquences énumère des objets horizontalement, et le temps verticalement. Il modélise l'exécution des différents messages en fonction du temps.

Chaque diagramme de séquence doit comprendre :

- i. Les acteurs intervenants dans la séquence de messages.
- ii. Une note contenant la description de la séquence représentée.
- iii. Les messages envoyés entre les entités (ou sur eux-mêmes) avec les paramètres pertinents.
- iv. Les lignes de vie des entités doivent montrer la création et la destruction lorsque pertinent.
- v. Les entités logicielles doivent avoir la bonne représentation UML : instance, instance nommée ou classe.

Le diagramme de séquences se base sur les concepts suivants :

- **Objet** Description d'un objet du monde réel (instance de classe). Il peut être une personne ou une chose.
- **Message** Les messages indiquent les communications entre les objets.
- **Message réflexif** Un objet peut envoyer à lui-même : c'est un message réflexif.
- **Période d'activité d'objet** Dans certains cas, il est représenté par la période pendant laquelle un objet est actif.

Diagramme de séquence : Authentification

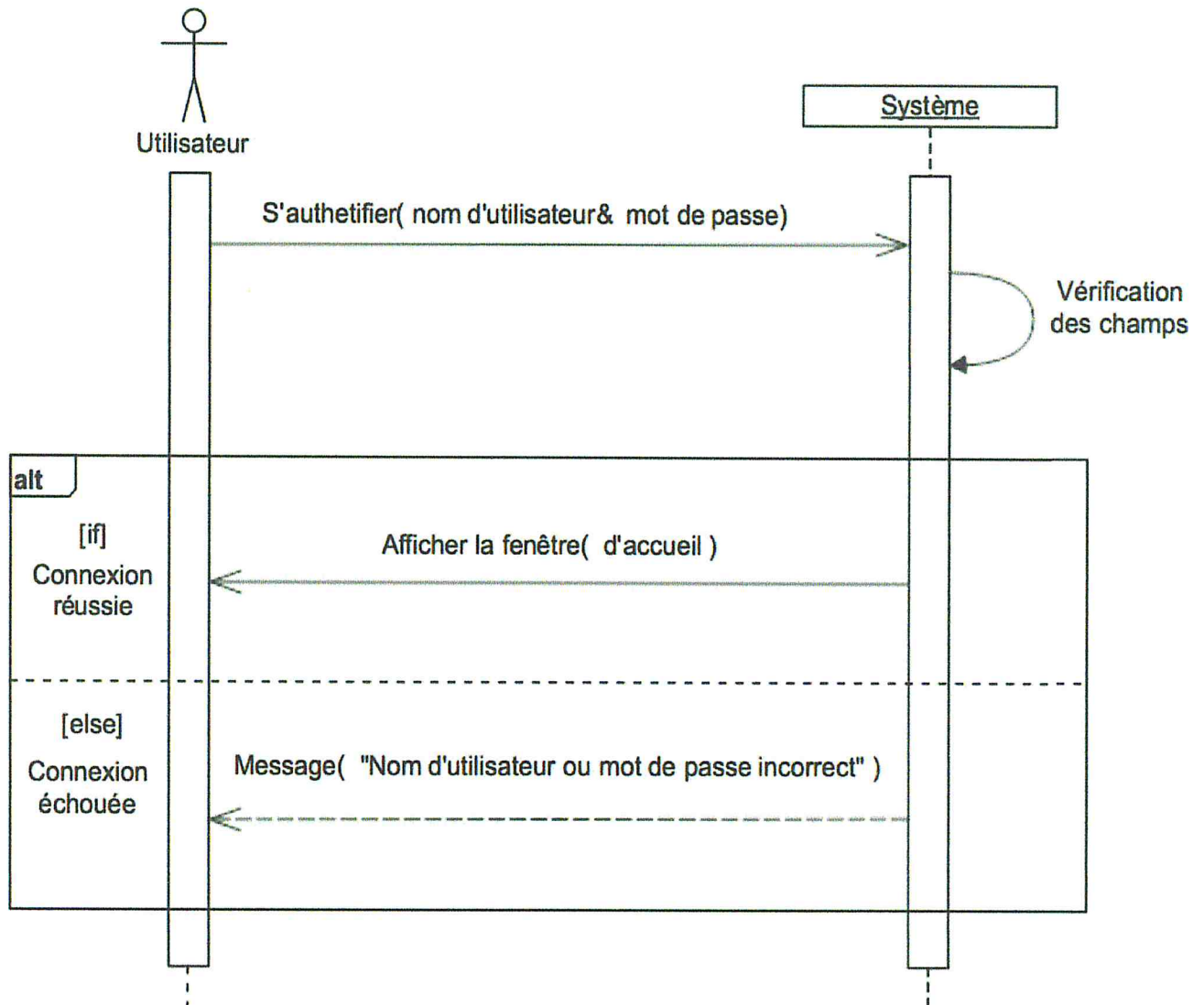


FIGURE 3.6: Diagramme de séquence : Authentification

Description du scénario :

- Le système affiche l'interface d'authentification.
- L'utilisateur introduit un login et un mot de passe.
- Le système vérifie le login et le mot de passe.
- Si les données saisies sont correctes le système affiche l'interface de l'application, sinon le système demande de répéter la saisie de login et mot de passe.

Diagramme de séquence : Créer nœud

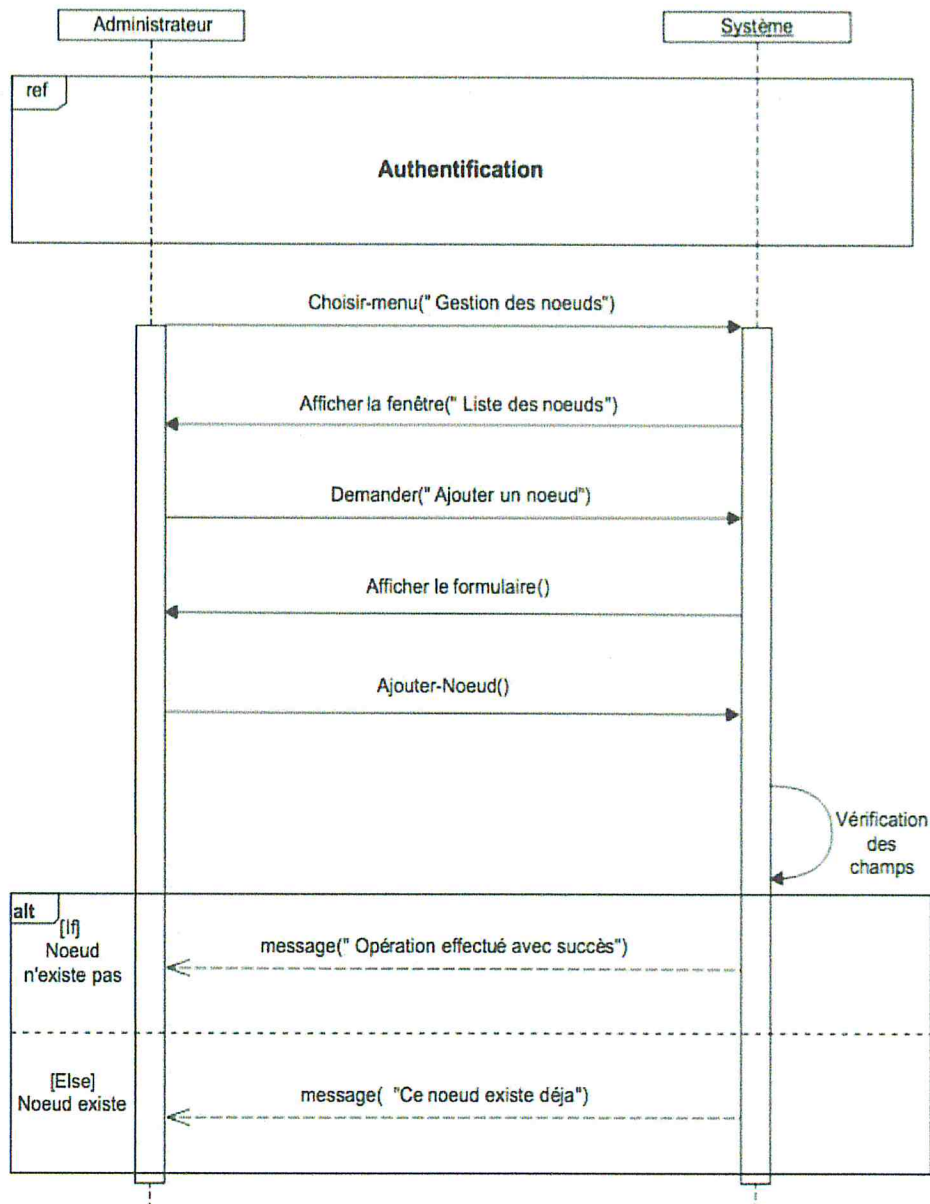


FIGURE 3.7: Diagramme de séquence : Créer nœud

Description du scénario :

- Après l'authentification, l'administrateur doit accéder au menu Gestion des nœuds et demande au système une nouvelle fiche nœud pour remplir les informations du nœud.
- Le système affiche le formulaire.
- L'administrateur saisit les informations du nœud et valide.
- Si le nœud n'existe pas dans la base de donnée le système enregistre le nouveau nœud, sinon le système renvoie un message que le nœud existe.

Diagramme de séquence : Gestion des nœuds

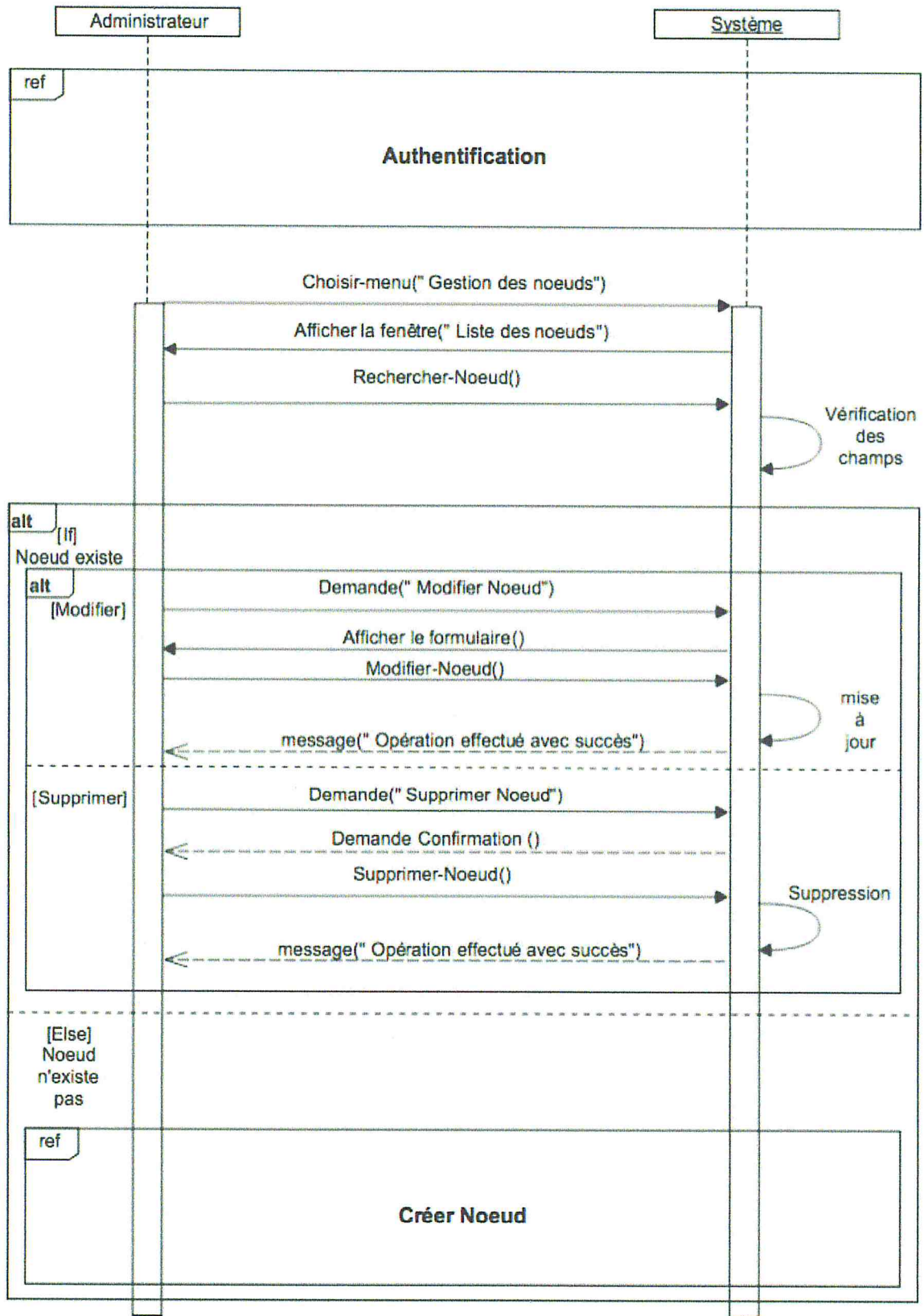


FIGURE 3.8: Diagramme de séquence : Gestion des nœuds

Description du scénario :

- Après l'authentification, l'administrateur doit accéder au menu gestion des nœuds et demande au système une recherche de nœuds.
- Si le nœud est trouvé l'administrateur peut supprimer ou modifier un nœud.
- Sinon il doit créer un nouveau nœud.

Diagramme de séquence : Créer réglage

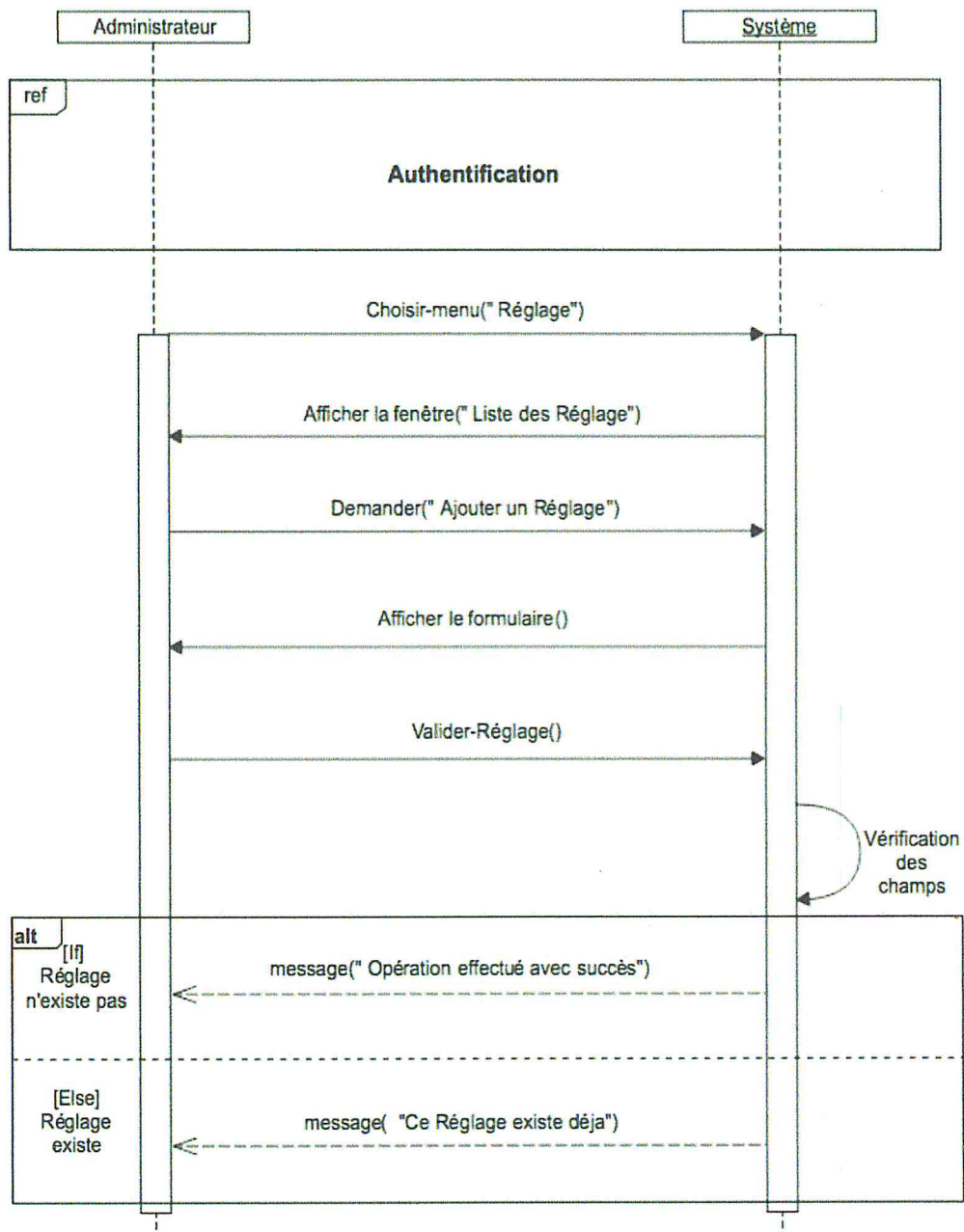


FIGURE 3.9: Diagramme de séquence : créer réglage

Description du scénario :

- Après l'authentification, l'administrateur doit accéder au menu Réglage et demande au système une nouvelle fiche réglage pour remplir les informations du réglage.
- Le système affiche le formulaire.
- L'administrateur saisit les informations du réglage et valide.
- Si le réglage n'existe pas dans la base de donnée le système enregistre le nouveau réglage, sinon le système renvoie un message que le réglage existe.

3.3.3 Diagrammes de classe

Présentation du diagramme de classes, leurs rôles, leurs attributs et leurs méthodes :

Nom de la Classe	Rôle de la classe	Attributs de la classe	Méthodes de la classe
Utilisateur	Super classe des individus du système	ID_utilis, Nom_utilisateur, Mot_de_pass, Etat, Nom,Prénom, telephone, Email, Sexe, date_naissance	Ajouter (), Modifier (), Désactiver (), Rechercher ().
Administrateur	Classe héritière de la classe 'Utilisateur'. Cette classe générale est reliée aux autres classes qui en dépendent.		Ajouter (), Modifier (), Supprimer (), Rechercher ().
Observateur	Cette classe définit les observateurs		Ajouter (), Modifier (), Supprimer (), Rechercher ().
Réglage_Seuil	Cette classe permet de représenter les différents règlements.	ID_reglage, Num_reglage, Nom_reglage, Seuil_min, Seuil_max, Type.	Ajouter (), Modifier (), Supprimer (), Rechercher ().
Site	Cette regroupe les différents sites du réseau	ID_site, Num_site, Nom_site	Ajouter (), Modifier (), Supprimer (), Rechercher ().
Noeud	Regroupe tous les Noeuds.	ID_noeud, Num_noeud, Plateforme.	Ajouter (), Supprimer (), Rechercher ().

TABLE 3.2: Description du diagramme de classe

Diagramme de classe

Le diagramme de classes représente l'aspect statique d'un système [19].

Les classes-associations permettent de décrire les liens entre classes sous forme d'une classe. Elles sont dotées d'attributs et d'opérations spécifiques à chaque instance du lien.

Une classe-association a le statut d'une classe décrivant les occurrences d'une association. Elle peut être dotée d'attributs, d'opérations, et être reliée à d'autres classes par des associations. Les flèches représentent le sens de navigation de l'association.

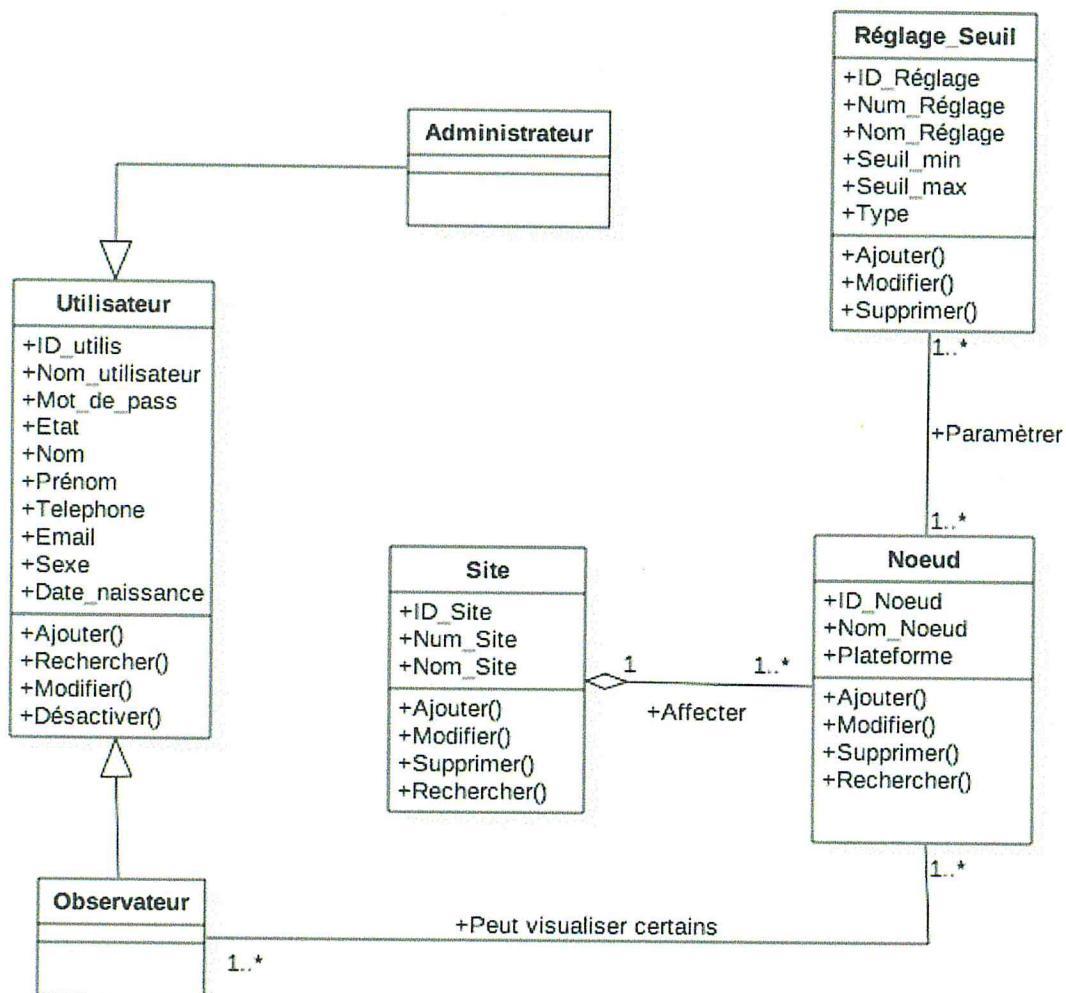


FIGURE 3.10: Diagramme de classe

Contraintes sur l'application

Une contrainte d'intégrité est une clause permettant de contraindre la modification de tables, faite par l'intermédiaire de requêtes d'utilisateurs, afin que les données saisies dans la base soient conformes aux données attendues.

Les contraintes d'intégrité attendu sur notre système sont pour la plus part défini par :

Des contraintes d'existence :

- Une alerte ne peut être lancée que si les seuils MAX et MIN sont bien définis.
- Un réglage ne peut être enregistré que si les noeuds existent dans la base de donnée.
- Une suppression de noeud ne peut être effectuée que si le noeud est existant et inactif.

Des contraintes d'unicité :

Bien évidemment toutes les clés primaires se doivent d'être uniques dans la base de données.

3.3.4 Règles de gestion

Pour concevoir le schéma de la base de données, nous allons passer du modèle objet au modèle relationnel en appliquant les règles de passage au diagramme de classe [20]. Les règles sont les suivantes :

Nous avons deux classes A et B. 'a' est un attribut de A et 'b' est un attribut de B.

Règle 1 :

Classe A $\leftarrow \rightarrow$ Classe B avec cardinalité (1, 1) : nous regroupons le contenu de A et B dans une même table.

Règle 2 :

Classe A $\leftarrow \rightarrow$ Classe B avec cardinalité (1, n) c'est-à-dire qu'une entité 'a' peut avoir plusieurs entités 'b' qui lui sont attachées : nous créons deux tables A et B. Dans la table B, nous prévoyons un attribut supplémentaire qui représente la clé de la table A.

Règle 3 :

Classe A $\leftarrow \rightarrow$ Classe B avec cardinalité (n, n) : nous créons trois tables ; table A, table B et table C qui représente une table qui fait les liens entre les éléments 'a' et les éléments 'b'. La table C aura comme attributs les clés des deux tables A et B.

3.4 Le passage vers le modèle relationnel

Le modèle relationnel représente depuis plusieurs années la tendance principale du marché pour l'implémentation des bases des données, la raison est que ce modèle est fondé sur les bases mathématiques qui facilitent sa manipulation. Lors d'une conception d'une base de données relationnelle l'étape principale consiste à définir un bon schéma relationnel (dont les relations sont en troisième forme normale). C'est à partir du diagramme de classe qui décrit le modèle objet des données de notre système et à l'aide de certaines règles de passage, que nous déduisons le schéma relationnel suivant :

	Désignation	Type	longueur	Null
<i>ID_utilis</i>	Un identifiant unique incrémentale des Utilisateur	Auto incrémentatation	12	non
<i>Nom_utilis</i>	Un Nom d'utilisateur unique des Utilisateur	String	20	non
Mot_de_pass	Un Mot de passe Attribué pour chaque utilisateur	String	20	non
Etat	Etat de l'utilisateur Activer ou désactiver	Boolean	1	non
Nom	Nom de l'utilisateur	String	20	non
Prenom	Préom de l'utilisateur	String	20	non
Date_naissance	Date de naissance de l'utilisateur	Date		non
Sexe	Sexe de l'utilisateur	String	20	non
Telephone	Numéro de téléphone de l'utilisateur	Int		
Email	Email de l'utilisateur	Mail		non

TABLE 3.3: Dictionnaire des données de la classe utilisateur

3.5 Implémentation de l'application

L'IHM (interface homme-machine) représente un élément clé dans l'utilisation de tout système embarqué et conditionne pour une large part son succès. En théorie, une interface homme-machine doit être ergonomique aussi bien qu'efficace. De plus, ces interfaces doivent être faciles à utiliser et compréhensibles par les utilisateurs pour garantir un bon degré de fiabilité lors des interactions ainsi qu'un temps d'apprentissage réduit. De plus, elles doivent avoir un certain niveau d'intelligence et de standardisation pour finir par présenter les fonctionnalités de l'application avec une manière plus conviviale.

3.5.1 Présentation des outils de développement

Pour implémenter notre application nous avons utilisé les outils suivants :

Choix de MySQL Server comme SGBD

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données. Nous l'avons choisi pour notre application car il compte de nombreux avantages :

- Il gère de manière fiable une grande quantité de données dans un environnement multi-utilisateurs.
- Plusieurs utilisateurs peuvent accéder simultanément aux mêmes données.
- Empêche tout accès non autorisé et fournit des solutions efficaces pour la récupération des données après incident.

Cet outil fournit un analyseur de données OpenStreetMap pour les transformer en un nœud-arc adapté pour les applications de routage.

Choix du Langage W - Windev comme environnement de développement

Le WLangage est un langage de programmation de 5^{ème} génération inclus dans les outils de développement WinDev, WebDev et WinDev Mobile. Il est propriétaire et ne peut être manipulé qu'avec les outils PC SOFT. Le WLangage est né en 1992. avec la première version de WinDev.

3.5.2 Présentation de l'application

Notre application prend la forme d'un assistant qui est forme d'un menu situe en haut et d'une interface principale.

Interface d'authentification

Au démarrage de l'application, un ecran d'authentification sera affiché qui contient les informations à remplir pour l'authentification des utilisateurs (le login et le mot de passe) comme il est illustré dans la figure ci-dessous.

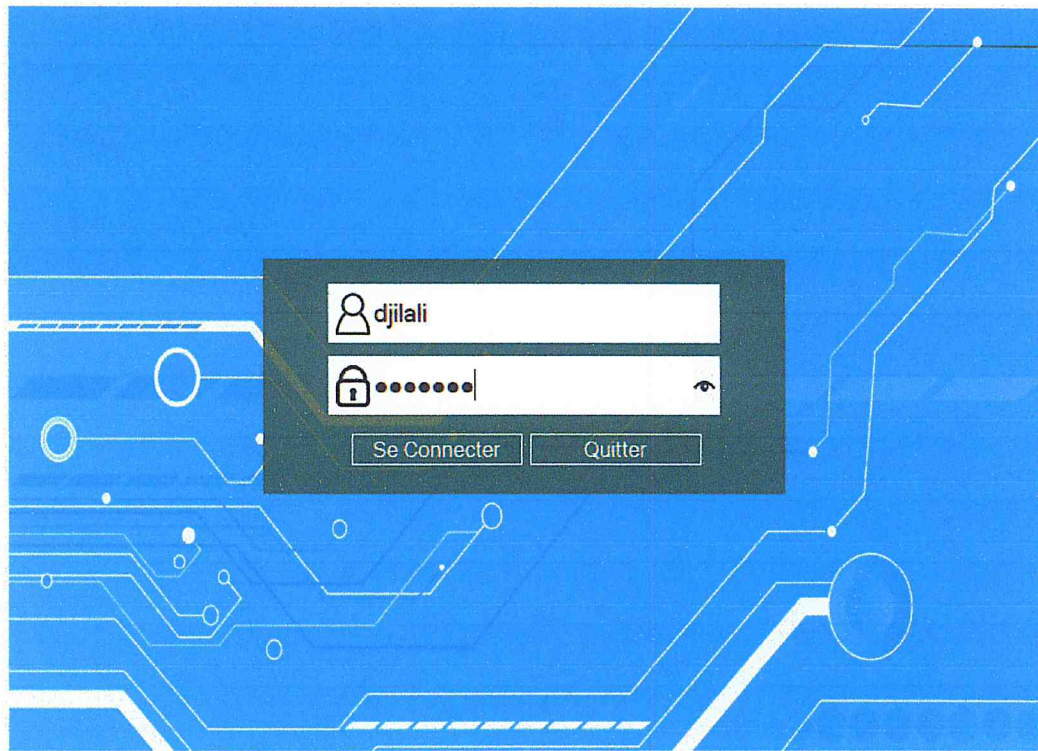


FIGURE 3.11: l'interface d'authentification

L'interface d'authentification permet d'initialiser l'application et charger seulement les fonctionnalités accessibles et les droits d'utilisation pour chaque utilisateur. En se connectant en tant qu'observateur, le menu Fichier et Affichage reste visible est accessible.

Dans ce qui suit, nous allons nous connecter en tant qu'administrateur pour pouvoir accéder a toutes les fonctionnalités de l'application.

Le menu fichier

Le menu Fichier qui est présenté dans la figure ci-dessous propose les services suivants :

- *Tableau de bord* : Présente l'interface principale.
- *Déconnexion* : Permet de fermer la session.
- *Quitter* : Permet de fermer l'application.



FIGURE 3.12: Le menu fichier

Le menu affichage

Le menu Affichage offre une multitude de services :

- *Configuration Tableau de bord* : Cette option est visible seulement pour l'administrateur, elle représente l'affichage du tableau de bord pour les observateurs.
- *Etat Noeuds* : Affichage état des noeuds.
- *Température* : Affichage de la Température.
- *Humidité* : Affichage de l'humidité.

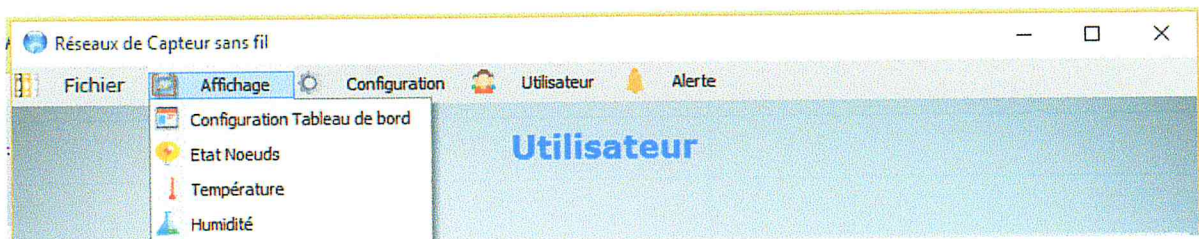


FIGURE 3.13: Le menu affichage

Le menu configuration

Passant maintenant au menu Configuration, ce menu est accessible que par l'administrateur et permet de faire :

- *Gestion des sites* : Accéder à la liste des sites.
- *Gestion des noeuds* : Accéder à la liste des Noeuds.
- *Réglage* : Accéder à la liste des Réglages.

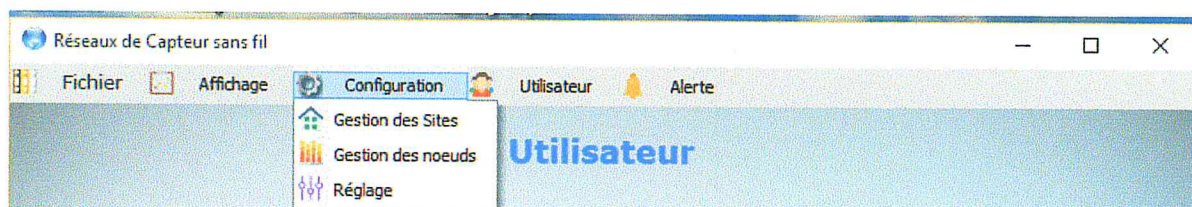


FIGURE 3.14: Le menu configuration

Le menu Utilisateur

Le menu Utilisateur est visible que par l'administrateur et permet de gérer les comptes des utilisateur.



FIGURE 3.15: Menu Utilisateur

Le menu Alerte

Enfin le menu Alerte sert à Envoyer es alertes aux utilisateurs lors d'un évènement de dépassement des seuils .

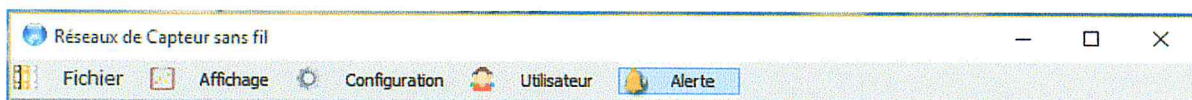


FIGURE 3.16: Menu Alerte

Interface principale

La fenêtre principale de l'application offre une vision sous forme de tableau de bord. Cette interface contient tous les panneaux de surveillance des capteurs en état de marche. Elle permet aussi d'afficher les différentes mesures prises en temps réel.

Chaque panneau indique le type des capteurs, le site associé ainsi que tous les paramètres de chacun des capteurs.

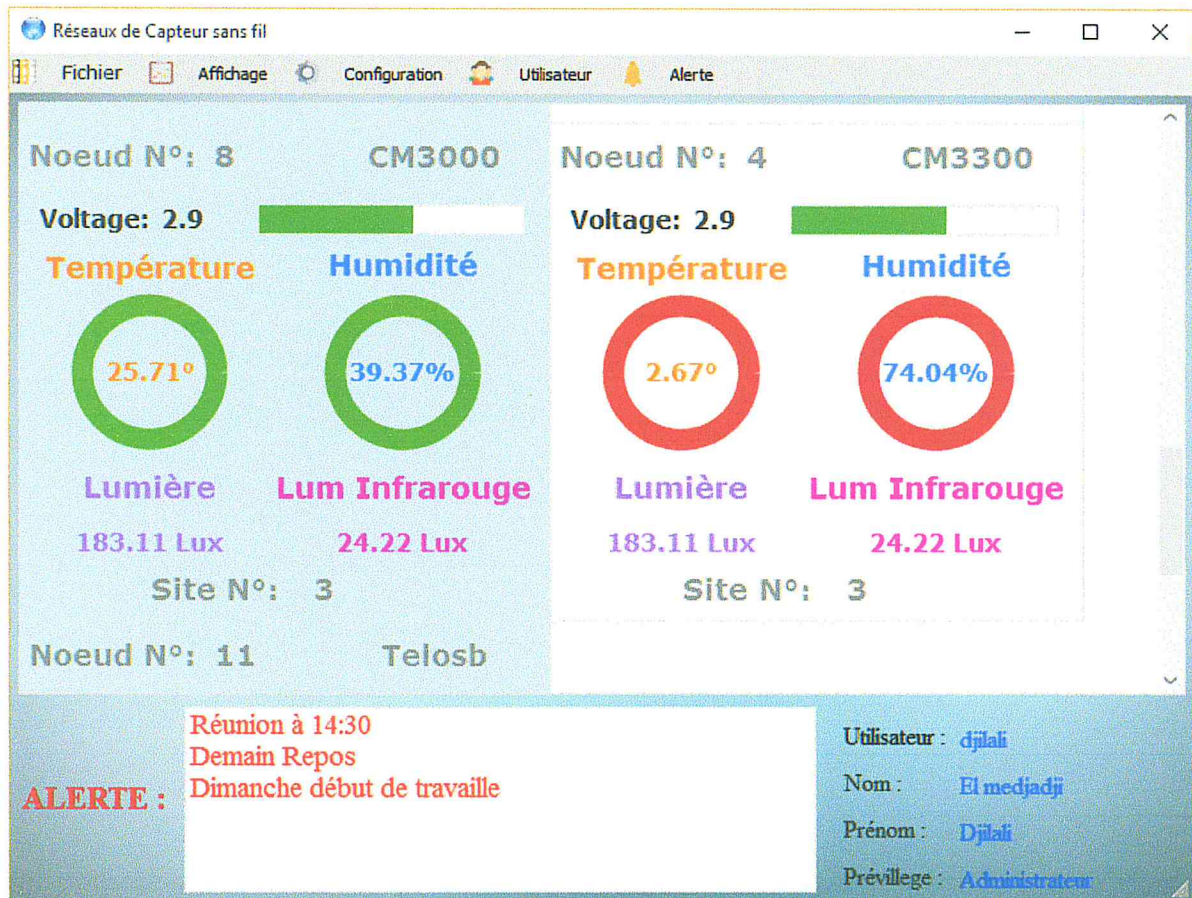
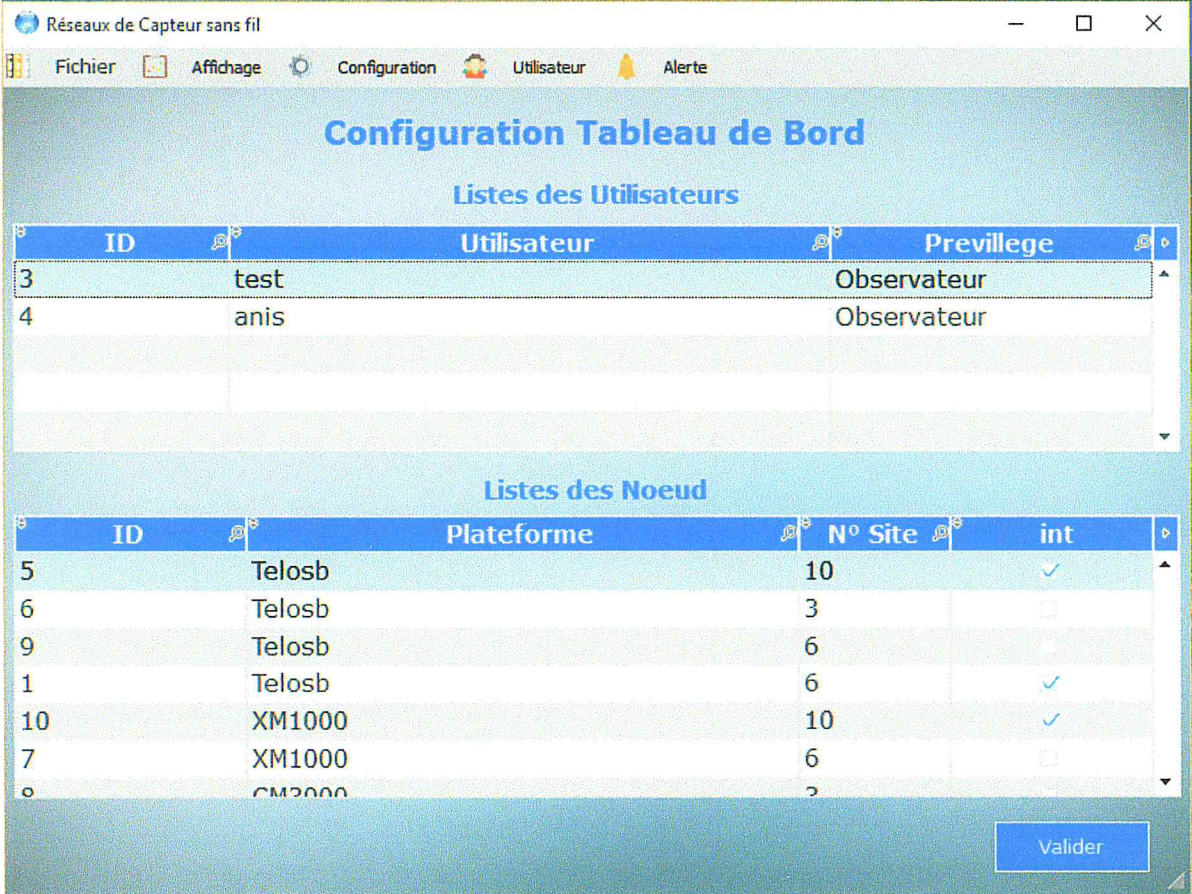


FIGURE 3.17: L'interface principale : Tableau de bord

Le panneau des configurations

Cette interface est accessible seulement par l'administrateur, elle permet de départager et d'affecter les noeuds aux personnes chargées de l'observation.



The screenshot shows a web application window titled 'Réseaux de Capteur sans fil'. The main content area is titled 'Configuration Tableau de Bord' and contains two tables. The first table, 'Listes des Utilisateurs', has columns for ID, Utilisateur, and Previllege. The second table, 'Listes des Noeud', has columns for ID, Plateforme, N° Site, and int. A 'Valider' button is located at the bottom right of the interface.

ID	Utilisateur	Previllege
3	test	Observateur
4	anis	Observateur

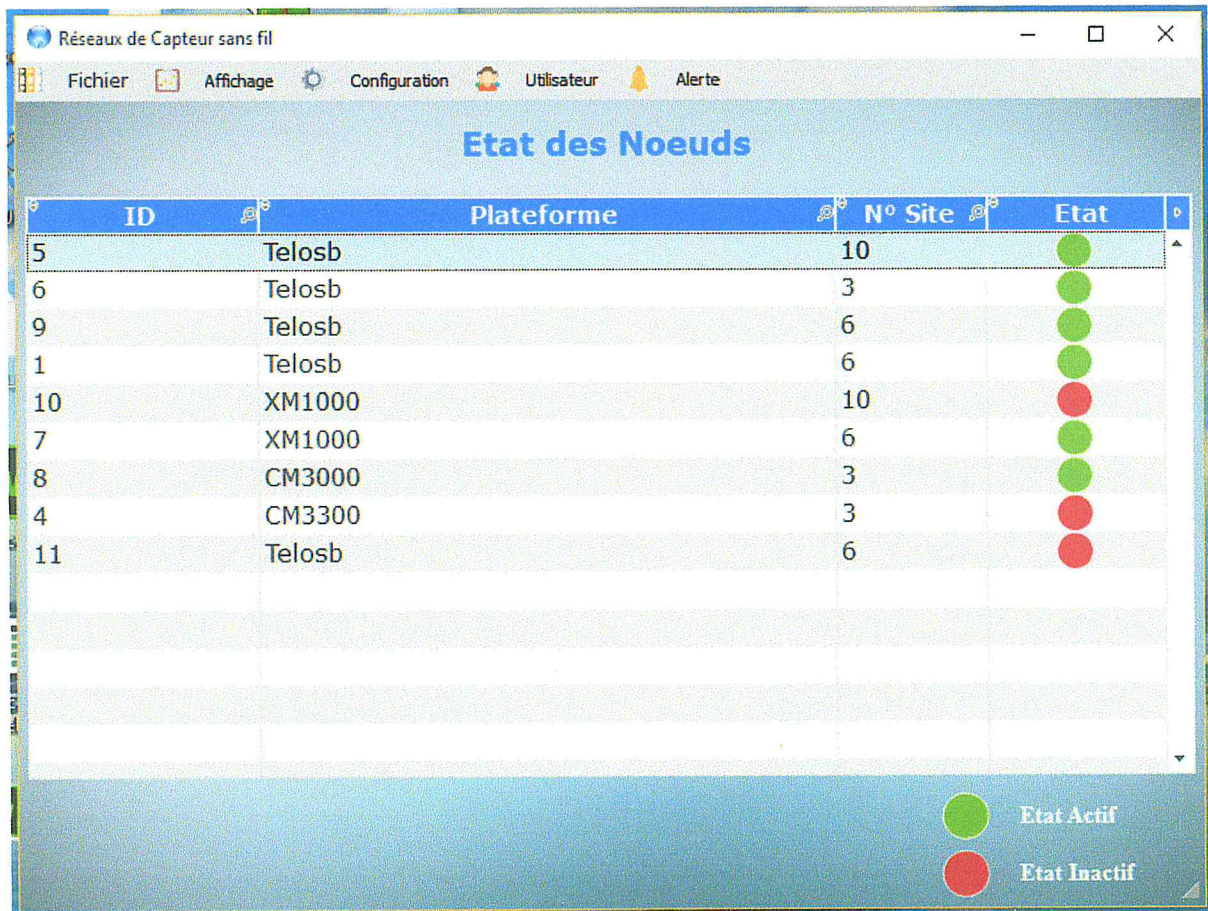
ID	Plateforme	N° Site	int
5	Telosb	10	<input checked="" type="checkbox"/>
6	Telosb	3	<input type="checkbox"/>
9	Telosb	6	<input type="checkbox"/>
1	Telosb	6	<input checked="" type="checkbox"/>
10	XM1000	10	<input checked="" type="checkbox"/>
7	XM1000	6	<input type="checkbox"/>
8	CM2000	3	<input type="checkbox"/>

FIGURE 3.18: L'interface : panneau des configurations

États des capteurs

Cette interface affiche l'état des noeuds de la façon suivante :

- Un noeud en état de marche est représenté par une couleur Verte.
- La couleur rouge est attribuée aux capteurs inactifs.



The screenshot shows a software window titled 'Réseaux de Capteur sans fil'. The main content area is titled 'Etat des Noeuds' and contains a table with the following data:

ID	Plateforme	N° Site	Etat
5	Telosb	10	● (Vert)
6	Telosb	3	● (Vert)
9	Telosb	6	● (Vert)
1	Telosb	6	● (Vert)
10	XM1000	10	● (Rouge)
7	XM1000	6	● (Vert)
8	CM3000	3	● (Vert)
4	CM3300	3	● (Rouge)
11	Telosb	6	● (Rouge)

At the bottom right of the interface, there is a legend:

- (Vert) Etat Actif
- (Rouge) Etat Inactif

FIGURE 3.19: L'interface : États des capteurs

Visualisation graphique

L'application développée permet également une visualisation des résultats se trouvant dans la base de données à travers des graphes offrant une analyse plus significative et détaillée.

La figure suivante montre une visualisation graphique des variations des températures pendant un laps de temps.

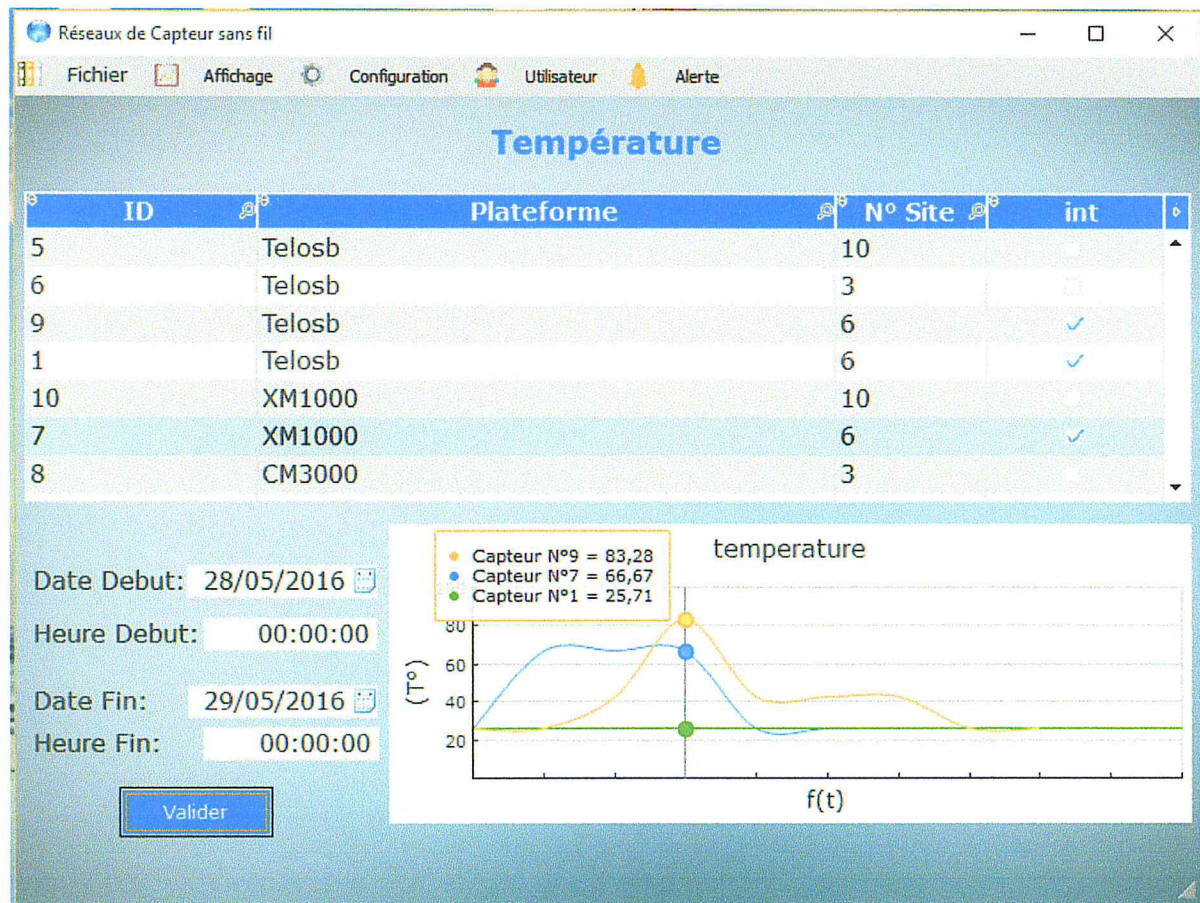


FIGURE 3.20: L'interface : l'historique des température

Gestion des Sites

Cette interface affiche la liste des Sites et permet de :

- *Ajouter* : Ajouter un nouveau site
- *Modifier* : Modifier un site
- *Supprimer* : Supprimer un site
- *Rechercher* : Rechercher un site par son Numéro ou par son Nom.

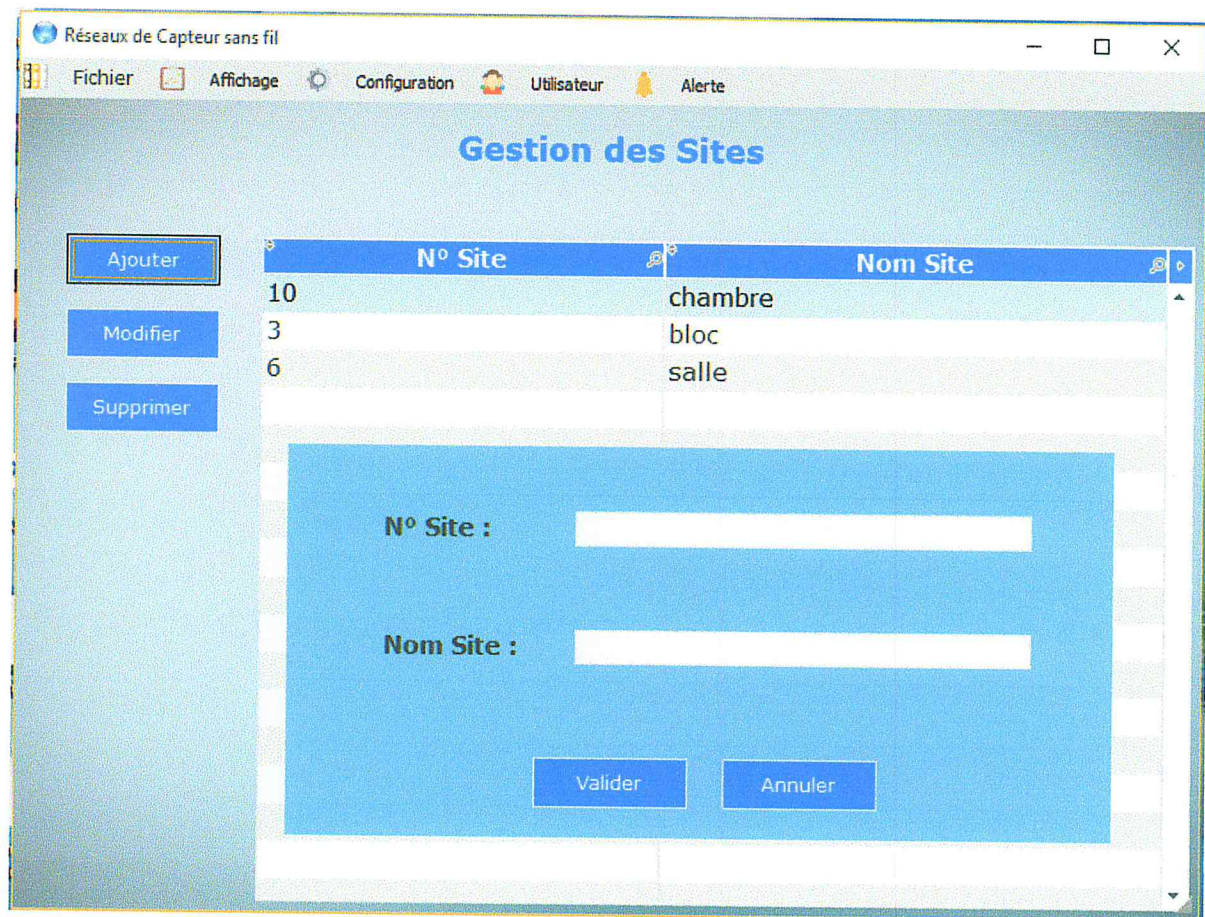


FIGURE 3.21: Gestion des Sites

Réglage

La fenêtre des réglages permet de lister tous les paramètres et les configurations techniques attribués aux capteurs.

Elle permet ainsi de supprimer, définir ou modifier les bornes des seuils afin de signaler tout type de dépassements concernant les températures et l'humidité comme le montre la figure ci-dessous.

ID capteur	Plateforme	N° Site	int
5	Telosb	10	<input checked="" type="checkbox"/>
6	Telosb	3	<input checked="" type="checkbox"/>
9	Telosb	6	<input checked="" type="checkbox"/>
1	Telosb	6	<input type="checkbox"/>
10	XM1000	10	<input type="checkbox"/>
7	XM1000	6	<input checked="" type="checkbox"/>
8	CM3000	3	<input type="checkbox"/>
4	CM3300	3	<input type="checkbox"/>
11	Telosb	6	<input checked="" type="checkbox"/>

FIGURE 3.23: Interface Fiche Réglage

Alerte

Cette interface représente un journal qui permet à l'administrateur du réseau d'adresser des consignes des messages à tout utilisateur de l'application en diffusion.

Cette interface offre aussi une signalisation sonore lors de détection d'anomalies dues au dépassement des seuils ainsi qu'une signalisation par mail.

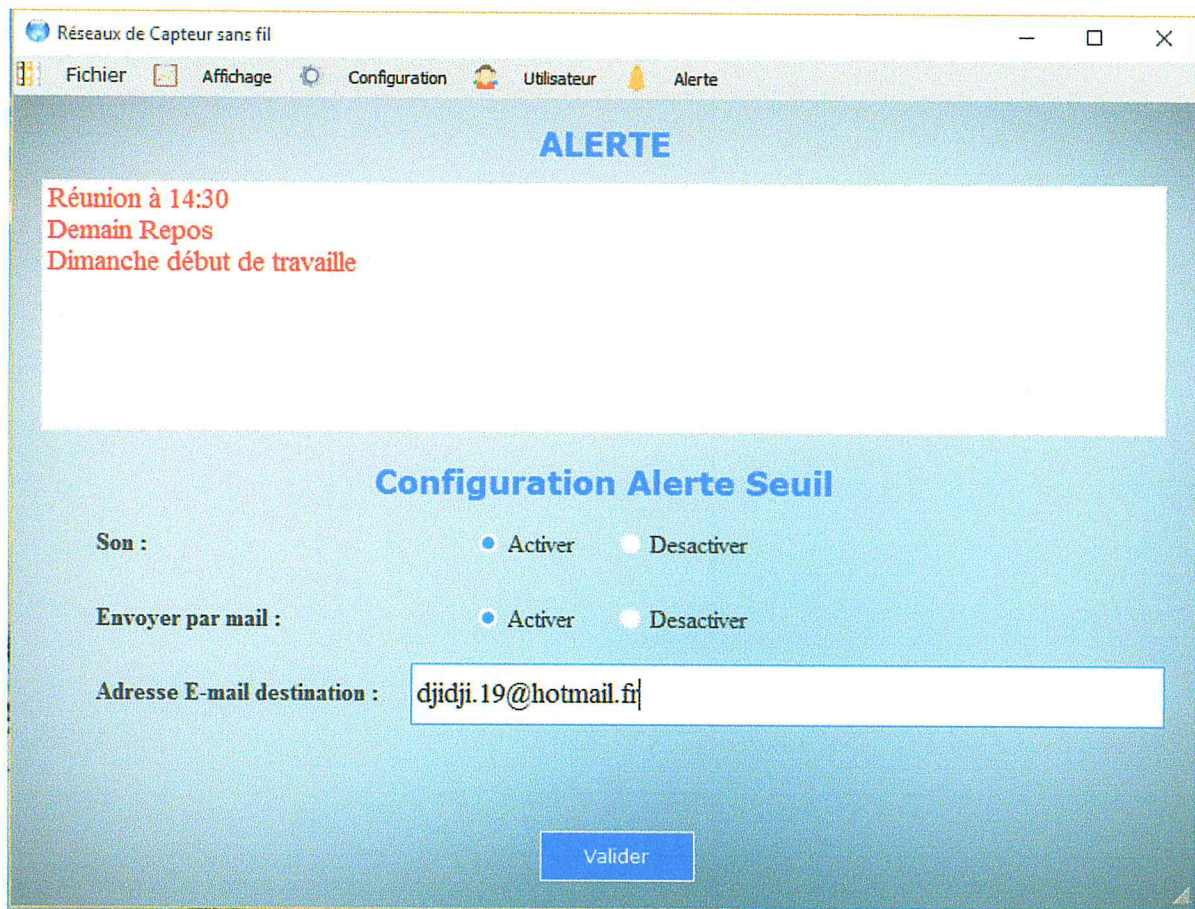


FIGURE 3.24: Interface Alerte

3.6 Conclusion

Dans ce chapitre, nous avons commencé par présenté la conception structurelle de notre application. Par la suite nous avons montré les différentes étapes de la réalisation de notre application destinée a la surveillance afin de concrétiser les résultats des mesures collectées par le RCSF précédemment.

Conclusion générale

Lorsque les capteurs sont déployés dans des zones d'intérêt ou encore déployés sur de grands espaces afin de récolter des données importantes, le choix et la conception d'une architecture adéquate est très important.

De ce fait, une bonne conception d'architecture permet d'avoir un impact sur la durée de vie du réseau tout entier et assure ainsi la fiabilité de l'information circulant à travers les nœuds du réseau.

Dans ce présent travail, Nous avons conçu et réalisé une application qui permet d'assurer une surveillance permanente d'une zone d'intérêt en temps réel à travers un réseau de capteurs sans fil.

Nous avons introduit dans un premier temps des généralités sur le domaine de la communication sans fil, nous avons aussi parler des capteurs sans fil, de leurs architecture ainsi que les différentes technologies de communication sans fil.

Ensuite nous avons présenté les différents capteurs utilisé lors du développement de notre réseau. nous avons également parlé des différentes solutions que nous proposons pour la conception et l'architecture du réseau.

La conception et la réalisation de l'application figurant a la fin de notre mémoire permet de concrétiser les résultats obtenus, et ont montré que les différentes solutions ont bien servi à résoudre le problème de la surveillance par réseau de capteurs sans fil.

Néanmoins, les réseaux de capteurs sans fil est un domaine de recherche qui reste toujours ouvert pour de nouvelles idées. Ce domaine est confronté à une multitude de défis scientifiques, notamment dans les domaines de l'optimisation et de la sécurité. Les applications basées sur ces réseaux ont souvent besoin d'un niveau de sécurité élevé car ils fournissent des services essentiels, voire vitaux.

Perspectives

Les réseaux de capteurs sans fil reste un champ de technologie très vaste et qui pose jusqu'à aujourd'hui plusieurs obstacles matériels et logiciels.

Nous pourrons par la suite améliorer notre travail en améliorant l'interconnectivité des nœuds en explorant plusieurs pistes de réflexion tel que :

- L'optimisation des chemins empruntés lors de l'acheminement des données.
- Le Cryptage et le chiffrement des données circulant à travers les nœuds du réseau.

Annexe A

Annexes

Système d'exploitation TinyOs

L'installation de tinyOS s'est avérée ne pas être aussi simple que d'installer une application normale.

Nous avons essayé plusieurs méthodes avant de réussir à l'installer correctement.

Tout d'abord nous avons essayé de l'installer sur une machine virtuelle Fedora 8, mais malheureusement pour nous, nous avons rencontré beaucoup de problèmes avec l'installation des packages RPM, le programme ne fonctionnait pas. Alors nous avons essayé de l'installer sur une machine virtuelle Windows 7, nous avons suivi un tutoriel et l'installation est terminée avec succès, mais nous avons rencontré des problèmes lors de l'exécution des tutoriels.

Alors nous sommes finalement arrivés à l'installer correctement sur ubuntu 14.04, mais cela après la résolution de plusieurs problèmes. Vous pouvez trouver les étapes d'installation de Tinyos 2.1 dans ce site :

http://tinyos.stanford.edu/tinyos-wiki/index.php/Installing_TinyOS_2.1.1

Le langage NesC

Le langage NesC (network embedded system C) est un dialecte de C basé sur des composants. NesC est orienté pour satisfaire les exigences des systèmes embarqués. De plus, il supporte un modèle de programmation qui agrège l'administration des communications, les concurrences provoquant les tâches et les événements ainsi que la capacité de réagir par rapport à ces événements.

NesC réalise aussi une optimisation dans la compilation du programme, en détectant les carrières possibles de données qui peuvent produire des modifications concurrentes au même état, à l'intérieur du processus d'exécution de l'application. Une carrière de données se produit quand plus d'un fils peuvent simultanément accéder à la même section de mémoire (concurrence d'accès mémoire entre threads), et quand au moins l'un des accès est un "write".

NesC simplifie aussi le développement d'applications et réduit la taille du code un critère important dans l'implémentation de code dans un capteur étant donné sa capacité de mémoire.

Bibliothèque

```
#ifndef TESTCM5000_H
#define TESTCM5000_H

enum{
DEFAULT_TIMER    = 5000, // 05 secondes
MAX_SENSORS     = 5,    // Nombre des sensors (Vref, Temp, Hum, Light, TSR)
TestCM5000_AM_ID = 0x07, // TestCM5000 AM ID
};

typedef nx_struct THL_msg {
    nx_uint16_t vref;
    nx_uint16_t temperature;
    nx_uint16_t humidity;
    nx_uint16_t photo;
    nx_uint16_t radiation;
    nx_uint16_t id;
} THL_msg_t;

#endif
```

Déclaration des composants

```
configuration TestCM5000C {
}
implementation {

    components MainC, TestCM5000P as App, LedsC;
    MainC.Boot <- App;           // Main
    App.Leds -> LedsC;           //Leds

                                // Radio
    components ActiveMessageC as Radio;
    App.RadioControl -> Radio;

    components new AMSenderC(TestCM5000_AM_ID);
    App.ThlSend -> AMSenderC;
    App.Packet -> AMSenderC;

                                // Timers
    components new TimerMilliC() as SampleTimer;
    App.SampleTimer -> SampleTimer;

                                // Sensors

    components new Msp430InternalVoltageC() as SensorVref; // Voltage
    App.Vref -> SensorVref;

    components new SensirionSht11C() as SensorHT;
    App.Temperature -> SensorHT.Temperature; //Température
    App.Humidity -> SensorHT.Humidity; // Humidité

    components new HamamatsuS1087ParC() as SensorPhoto; // Luminosité
    App.Photo -> SensorPhoto;

    components new HamamatsuS1087ITsrC() as SensorTotal; // Radiations Infra-rouges
    App.Radiation -> SensorTotal;
}
}
```

Implémentation et configuration des interfaces

Déclaration des interfaces

```
#include "TestCM5000.h"

module TestCM5000P @safe() {

uses {

    // Main, Leds
    interface Boot;
    interface Leds;

    // Radio
    interface SplitControl as RadioControl;
    interface AMSend as ThlSend;
    interface Packet;

    // Timers
    interface Timer<TMilli> as SampleTimer;

    // Sensors
    interface Read<uint16_t> as Vref;
    interface Read<uint16_t> as Temperature;
    interface Read<uint16_t> as Humidity;
    interface Read<uint16_t> as Photo;
    interface Read<uint16_t> as Radiation;
}
}
```


Évènements de démarrage

```
event void Boot.booted() {  
    call SampleTimer.startPeriodic(DEFAULT_TIMER);  
    // demarquer timer  
}
```

```
event void SampleTimer.fired() {  
    numsensors = 0;  
    //appel des méthodes de lecture  
    call Vref.read();  
    call Temperature.read();  
    call Humidity.read();  
    call Photo.read();  
    call Radiation.read();  
}
```

Les méthodes de lecture

```
event void Vref.readDone(error_t result, uint16_t value) {
    data.vref = value;
    if (++numsensors == MAX_SENSORS) {
        call RadioControl.start();
    }
}

event void Temperature.readDone(error_t result, uint16_t value) {
    data.temperature = value;
    if (++numsensors == MAX_SENSORS) {
        call RadioControl.start(); // demarrer radio si dernier sensor
    }
}

event void Humidity.readDone(error_t result, uint16_t value) {
    data.humidity = value;
    if (++numsensors == MAX_SENSORS) {
        call RadioControl.start();
    }
}

event void Photo.readDone(error_t result, uint16_t value) {
    data.photo = value;
    if (++numsensors == MAX_SENSORS) {
        call RadioControl.start();
    }
}

event void Radiation.readDone(error_t result, uint16_t value) {
    data.radiation = value;
    if (++numsensors == MAX_SENSORS) {
        call RadioControl.start();
    }
}
```

Les évènements et méthodes d'envoi

```
event void RadioControl.startDone(error_t err) {
    if (err == SUCCESS) {
        call Leds.led3On();
        post sendThlMsg(); // lancement radio et envoi de msg
    }else {
        call RadioControl.start();
    }
}

event void RadioControl.stopDone(error_t err) {
    if (err != SUCCESS) {
        call RadioControl.stop(); // arret radio
    }
}

task void sendThlMsg() {
    THL_msg_t* aux;
    aux = (THL_msg_t*)
    call Packet.getPayload(&auxmsg, sizeof(THL_msg_t));

    aux -> vref      = data.vref;
    aux -> temperature = data.temperature;
    aux -> humidity   = data.humidity;
    aux -> photo      = data.photo;
    aux -> radiation  = data.radiation;
    aux -> id = 0x1;

    if (call ThlSend.send(TestCM5000_AM_ID, &auxmsg, sizeof(THL_msg_t))!= SUCCESS) {
        post sendThlMsg();
    }
}

event void ThlSend.sendDone(message_t* msg, error_t error) {
    if (error == SUCCESS) {
        call RadioControl.stop(); // msg envoyé, arreter radio
        call Leds.led2Off();
    }else {
        post sendThlMsg();
    }
}
```


Évènement de réception et de transtypage - (Relais)

```
event message_t* Receive.receive(message_t* msg, void* payload, uint8_t len){
    if (busy!=FALSE){
        if (len == sizeof(THL_msg_t)) {
            call Leds.ledOn();
            busy = TRUE; // radio occupé

            btrpkt2 = (THL_msg_t*)(call Packet.getPayload(&pkt, sizeof(THL_msg_t)));
            // chargement du packet a envoyé

            btrpkt3 = (THL_msg_t*)payload; // packet recu
            //transtypage du paquet reçu

            btrpkt2 -> vref      = btrpkt3 ->vref;
            btrpkt2 -> temperature = btrpkt3 -> temperature ;
            btrpkt2 -> humidity   = btrpkt3 -> humidity ;
            btrpkt2 -> photo      = btrpkt3 -> photo ;
            btrpkt2 -> radiation  = btrpkt3 -> radiation ;
            btrpkt2 -> id        = btrpkt3 -> id;

            if (call AMSend.send(AM_BROADCAST_ADDR,&pkt, sizeof(THL_msg_t)) == SUCCESS) {
                busy = FALSE; // libérer le canal radio
                call Leds.ledOff();
            }
        }
        call Leds.led1Toggle();
    }
    return msg;
}
```

Les diagrammes de séquence (la suite)

Diagramme de séquence : Ajouter un site

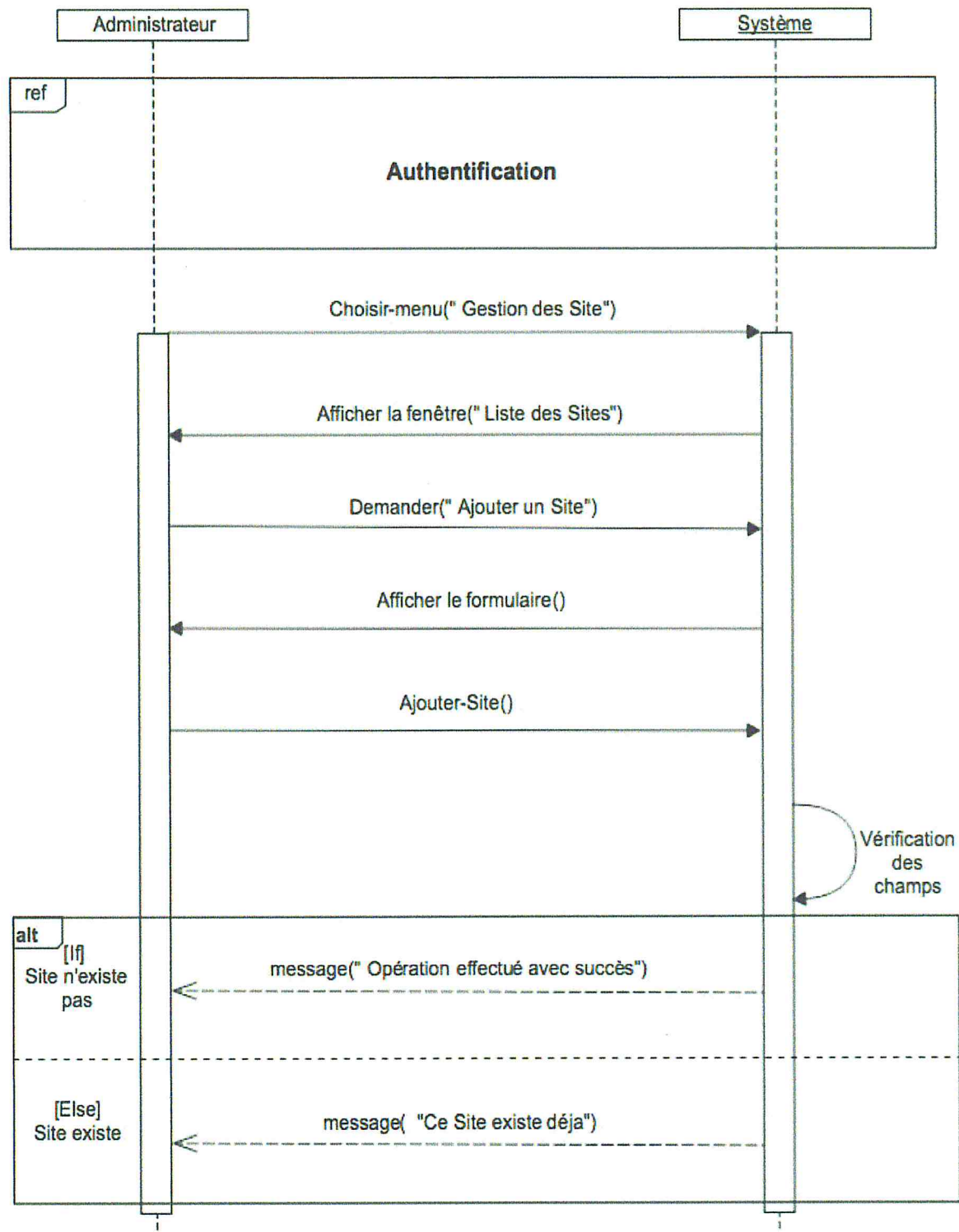


Diagramme de séquence : Gestion des utilisateurs

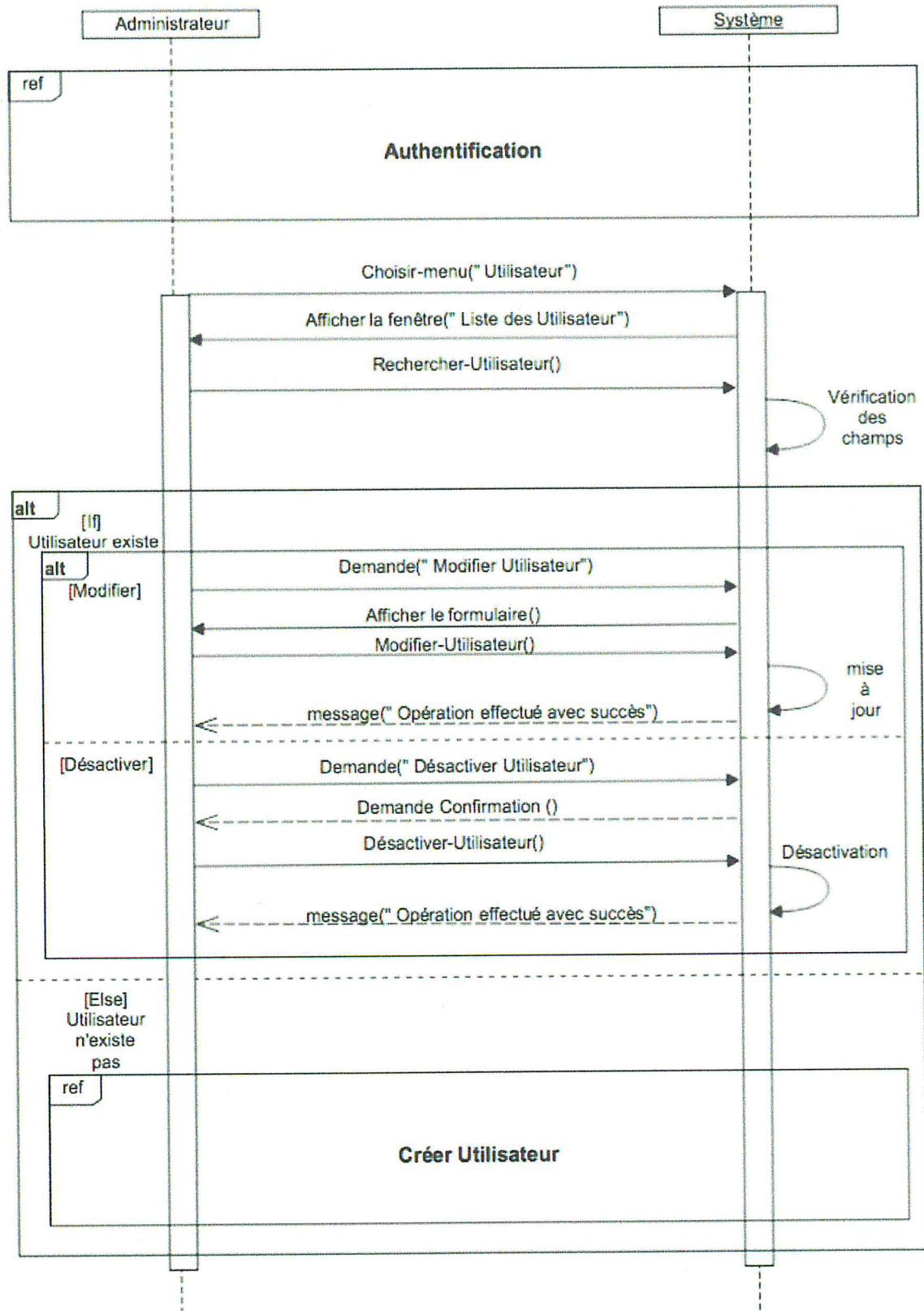


Diagramme de séquence : Ajouter un utilisateur

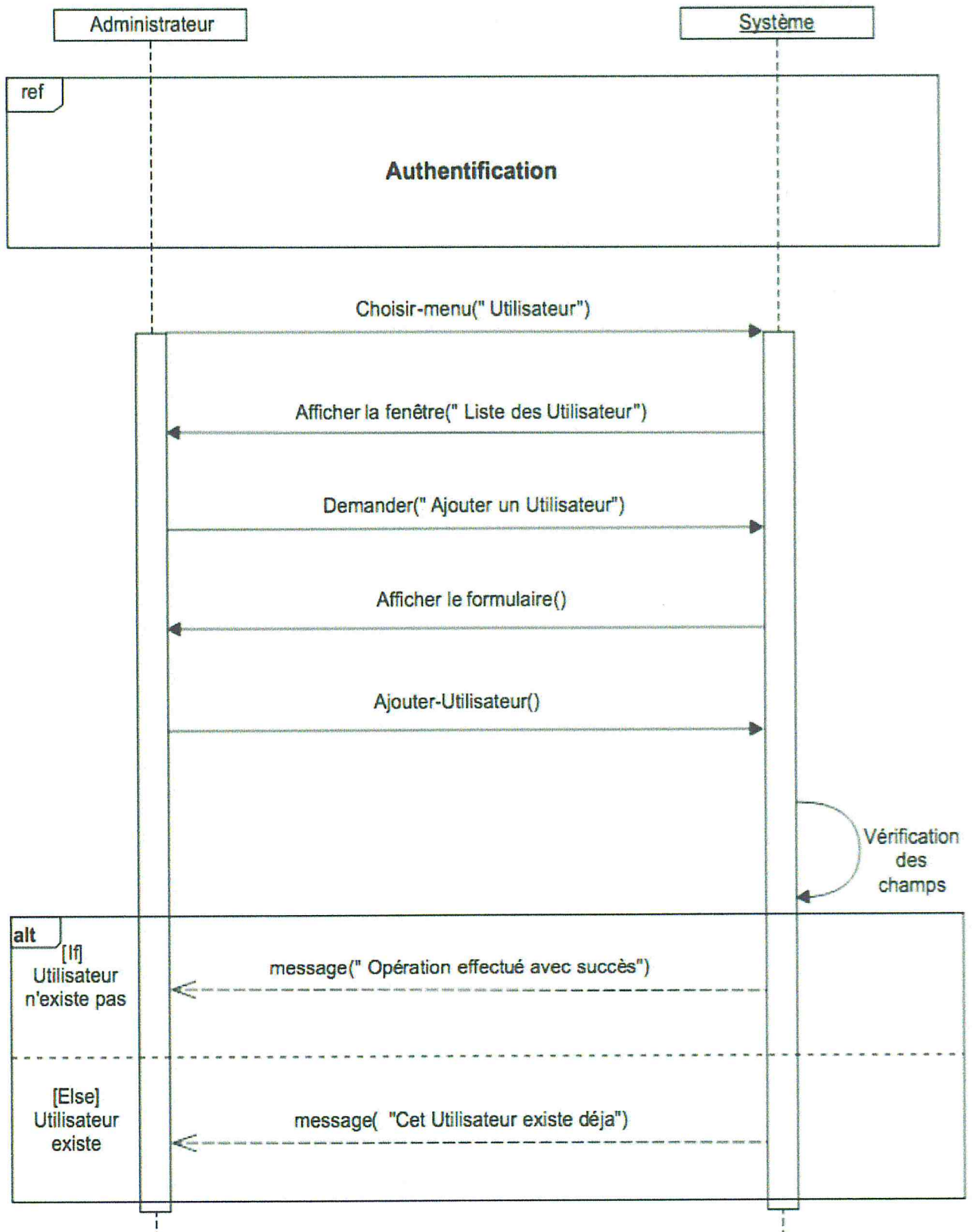
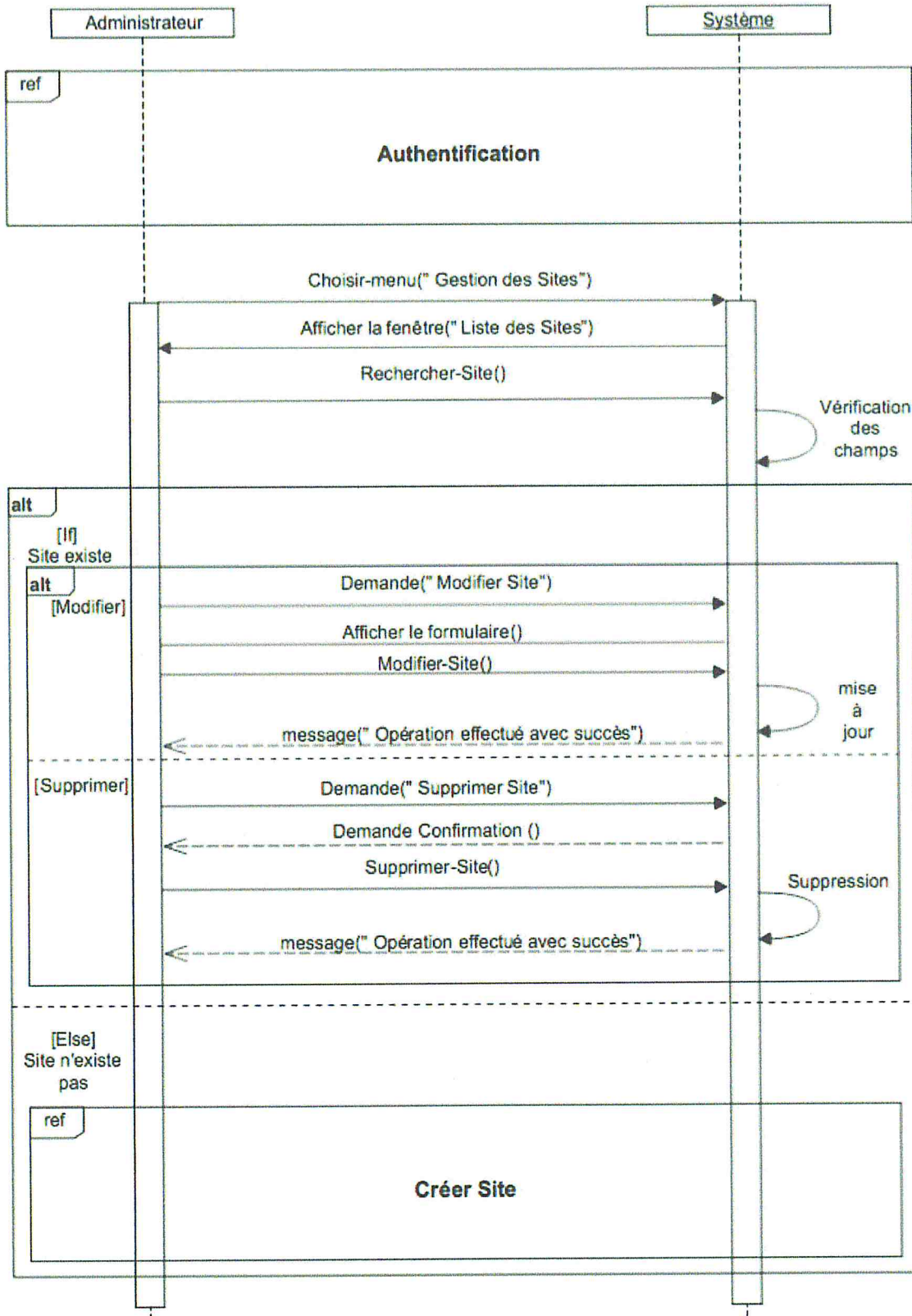


Diagramme de séquence : Gestion des sites



Bibliographie

- [1] Y. Sankarasubramaniam E. Cayirci I.F. Akyildiz, W. Su*. Wireless sensor networks : a survey. *Computer Networks* 38, 4, 2002.
- [2] Yanbo Shou. *Cryptographie sur les courbes elliptiques et tolérance aux pannes dans les réseaux de capteurs*. PhD thesis, 2014.
- [3] Salvatore D. Morgera et Ravi Sankar Ismail Butun. A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 16(1,Page 266-282), 2013.
- [4] Kamal BEYDOUN. *Conception d'un Protocole de routage hiérarchique pour les réseaux de capteurs*. PhD thesis, 2009.
- [5] A.Riddoch R.Ong et J.Hart K.Martinez, P.Padhy. *Glacial Environment Monitoring using Sensor Networks*, chapter Proceedings of the 1st Workshop on Real-World Wireless Sensor Network. 2005.
- [6] la mobilité et l'aménagement. Centre d'études et d'expertise sur les risques, l'environnement. Les transports intelligents. <http://www.transport-intelligent.net/technologies/capteurs-77/>.
- [7] Andrew S.Tanenbaum et David J.Wetherall. *Computer Networks*. Prentice Hall, 2010.
- [8] Jean-François PILLOU. Réseaux sans fil. <http://www.commentcamarche.net/contents/1312-wpan-reseau-personnel-sans-fil>.
- [9] Orange corporation. <http://www.francetelecom.com/rd>.
- [10] Alliance Zigbee. Le zigbee. <http://www.zigbee.org>.
- [11] K. J. Ray Liu Zhu Han. *Resource Allocation for Wireless Networks*, chapter Basics. Cambridge University, 2008.
- [12] MEMSIC Inc. Sg1000 advanticsys bridge. <http://www.advanticsys.com/wiki/index.php?title=SG1000f>, 2003.
- [13] MEMSIC Inc. Telosb mote platform - datasheet. http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf, 2003.
- [14] Advanticsys. Xm100 mote platform - datasheet. <http://www.advanticsys.com/wiki/index.php?title=XM1000>.
- [15] Advanticsys. Cm5000 mote platform - datasheet. <http://www.advanticsys.com/wiki/index.php?title=CM5000>.
- [16] Advanticsys. Cm4000 mote platform - datasheet. <http://www.advanticsys.com/wiki/index.php?title=CM4000>.
- [17] L. Debrauwer and F. Van der Heyde. *UML 2 : modélisation des objets*. TechNote (Nantes). Editions ENI, 2006.

- [18] P. Roques. *UML 2 par la pratique*. Collection noire. Eyrolles, 2009.
- [19] L. Debrauwer and N. Karam. *UML 2 : Entraînez-vous à la modélisation*. Les TP informatiques. Editions ENI, 2009.
- [20] P. Delmal. *SQL2 - SQL3 : Applications à Oracle*. Bibliothèque des Universités. De Boeck Supérieur, 2000.