

République Algérienne Démocratique Et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université d'Ibn Khaldoun – Tiaret

Faculté des Mathématiques et de l'Informatique

Département Informatique

Mémoire de Master

Thème

**Adaptation et la simulation d'un algorithme de type DTN
(Delay Tolerant Network) pour les réseaux de capteurs sans fil**

Pour l'obtention du diplôme de Master II

Spécialité : Génie Informatique

Option : Systèmes Embarqués et Temps Réel

Réalisé par : OUARED Khaled

Dirigé par : Mr. BENGHENI Abdelmalek

Année Universitaire 2014-2015

Remerciements

Nous remercions d'abord Dieu tout puissant qui nous a donné la force et la patience pour accomplir ce modeste travail.

*Nous tenons à remercier mon encadreur **Mr BENGHENI Abdelmalek** pour l'encadrement, l'apport scientifique, le soutien incommensurable et l'aide appréciable consentie.*

*Nous tenons à remercier tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail, en particulier Mr **OUARED AEK**.*

Nos remerciements aux membres du jury, d'avoir accepté de juger notre travail.

Dédicaces

Je dédie ce modeste travail à :

✚ *Mes très chers **parents** pour leur soutien, leur patience infinie et leurs précieux conseils et encouragements,*

✚ *A mon frère et ma sœur pour leur patience et inquiétude.*

A mes très chers amis pour leur soutien et sympathie

Khaled

Résumé

Les réseaux tolérants aux délais (DTNs) sont des réseaux où la connectivité est intermittente en raison des difficultés rencontrées dans l'environnement à savoir : le climat, la mobilité, les pannes d'énergie, ...etc. Pour parvenir à livrer les données malgré tous ces défis, une nouvelle architecture réseau a été développée. Celle-ci consiste en l'ajout d'une couche supplémentaire dans la pile protocolaire OSI, au-dessus de la couche transport. Cette nouvelle politique est appelée : le protocole bundle, dont le principal rôle est le stockage de données jusqu'à ce qu'une occasion de transmission se présente, grâce à la technique Store & Forward et le concept de transfert de garde.

Du fait que cette technologie soit récente, alors divers problèmes sont encore au stade de recherche, parmi lesquels nous citons le problème du routage.

Les protocoles de routage dans les réseaux tolérants aux délais ne cherchent pas à trouver le chemin le plus court tel que c'est le cas dans les protocoles classiques, mais ils jouent plutôt sur l'augmentation du taux de délivrance de données. Les techniques utilisées pour cela, peuvent être divisées en deux grandes familles. Certains protocoles adoptent l'approche de « réplication », qui consiste en la duplication des messages dans le réseau en fournissant de multiples copies afin d'augmenter la probabilité de transmission. D'autres se basent sur « la connaissance », qui consiste à récolter des informations sur l'état du réseau et gérer les envois de manière efficace. Mais quel que soit la politique suivie, ces protocoles de routage rencontrent souvent des obstacles liés à la consommation de ressources.

L'objet de notre travail est alors d'étudier les algorithmes de routage de type DTN déjà proposés dans la littérature et essayer d'adapter et de simuler un algorithme existant par l'utilisation d'un simulateur pour les réseaux de capteurs sans fil.

Mots-clés : Réseaux de capteurs sans fil (RCSFs), Réseaux tolérants aux délais, DTN, le protocole bundle, connectivité intermittente, Store & Forward, les protocoles de routage DTN.

Sommaire

Introduction Générale	1
Chapitre I : Généralités sur les réseaux de capteurs sans fil	
Introduction	3
1. Définition	3
2. Architecture d'un réseau de capteurs sans fil.....	3
3. Les capteurs.....	4
3.1. Introduction.....	4
3.2. Architecture physique d'un capteur	4
3.2.1. Unité de capture.....	5
3.2.2. Unité de traitement	5
3.2.3. Unité de communication	5
3.2.4. Unité d'énergie	6
3.3. La pile de communication (Pile protocolaire)	6
3.3.1. La couche physique	7
3.3.2. La couche liaison de données	7
3.3.3. La couche réseau	7
3.3.4. La couche transport.....	8
3.3.5. La couche application	8
3.4. Consommation d'énergie dans les réseaux de capteurs sans fil	8
3.4.1. Energie de capture	9
3.4.2. Energie de traitement	9
3.4.3. Energie de communication.....	9
3.5. Le niveau de gestion d'énergie	9
3.6. Domaines d'application.....	10
3.6.1. Les applications militaires.....	11
3.6.2. Les applications liées à la sécurité.....	11
3.6.3. Les applications environnementales	11
3.6.4. Les applications médicales.....	12
3.6.5. Les applications industrielles	12
4. Conclusion	13
Chapitre II : Le routage dans les réseaux DTN	

Introduction.....	20
1. Les réseaux tolérants aux délais (DTN).....	20
1.1. Définition.....	21
1.2. Architecture d'un DTN	22
1.2.1. Les entités de communication.....	24
1.2.1.1. Les nœuds DTN	25
1.2.1.2 Les régions DTN	26
1.2.1.3 Les tuples	27
2. Algorithmes de routage.....	28
2.1. Routage non Géographique	29
2.1.1. Epidemic Routing	29
2.1.2. Calculs de prédictibilité	29
2.1.3. Stratégie de la transmission.....	31
2.1.4. Gérer la file d'attente	33
3. Les algorithmes de routage de type DTN pour les réseaux de capteurs sans fil	33
4. Conclusion.....	34
Conception et implémentation	35
Chapitre III : Conception et implémentation	36
I. Conception de l'algorithme adapté «algorithme épidémique»	36
INTRODUCTION.....	36
I.1. Formulation du problème de routage	36
I.2. Modélisation des réseaux DTNs	37
I.2.2 Stockage des messages.....	37
I.3. L'algorithme de routage adapté	37
I.4. La Conception UML de l'algorithme adapté.....	38
I.4.1. Identification des primitifs de l'algorithme.....	38
I.4.2. Déroulement de notre algorithme	40
I.4.2.2. Diagramme d'état de transition d'un nœud	Erreur ! Signet non défini.
I.4.2.2. Diagramme d'état de transition d'un paquet	Erreur ! Signet non défini.
I.4.2.3 Diagramme de Classes.....	42
I.4.2.3.1 Description des classes.....	43
I.4.2.4. Diagramme de séquence	44
II. Implémentation et Expérimentation.....	45

II.1. Implémentation de Notre application	45
II.2. Description des choix techniques d'implémentation.....	45
II.3. Présentation de l'application	45
II.3.1 Interface d'accueil application.....	46
II.3.2 Interface RCSF	46
II.3.3 Interface paramétrage.....	47
II.3.4 Interface simulation & visualisation	48
II.4. Les séries d'expériences	48
II.4.1 Durées d'une Connexion	48
II.4.2 Nombre de paquets reçus en fonction de la taille des files d'attente	49
Nous étudions l'influence de la taille de la file d'attente sur les performances.	49
II.5. Conclusion	50
Conclusion et Perspectives	51
Bibliographies	52

Liste des abréviations

OSI	Open System Interconnection
PRoPHET	Probabilistic ROuting Protocol using History of Encounters and Transitivity
UDP	User Datagram Protocol
UWB	Ultra WideBand
TCP	Transmission Control Protocol
TCPCL	TCP Convergence Layer
TTL	Time To Live
RF	Radio Frequency
RTT	Round-Trip delay Time
LAN	Local Network Area
IP	Internet Protocol

Liste des figures

Figure 1 : Architecture simplifiée d'un réseau de capteurs [AMZ11]	4
Figure 2 : Architecture d'un capteur [HAM07b]	5
Figure 3 : Pile protocolaire dans les RCSFs [BEN09]	7
Figure 4 : Répartition de la consommation d'énergie dans un capteur [POM04]	10
Figure 5 : Transfert en utilisant un nœud intermédiaire	21
Figure 6 : Store-and-Forward Message Switching.....	22
Figure 7 : Emplacement de la couche Bundle dans la pile protocolaire [WAR03]	23
Figure 8 : La couche Bundle présente dans toutes les régions [WAR03]	23
Figure 9 : Les options de livraisons offertes par la couche bundle. [KRI08]	24
Figure 10 : l'appartenance des nœuds aux différentes régions [WRVS07]	25
Figure 11 : Les différents rôles d'un nœud DTN [WFW03]	26
Figure 12 : Adressage d'un noeud dans un DTN [WRVS07]	28
Figure 13 : Schématisation du problème du routage dans les réseaux DTN	36
Figure 14 : Le Diagramme des cas d'utilisation de notre application.....	40
Figure 15 : Diagramme d'activité de notre algorithme adapté	41
Figure 16 : Diagramme d'état de transition d'un nœud	Erreur ! Signet non défini.
Figure 17 : Diagramme d'état de transition d'un paquet.....	Erreur ! Signet non défini.
Figure 18 : Diagramme de classe	42
Figure 19 : Diagramme de séquence	44
Figure 20 : Interface d'accueil application	46
Figure 21 : interface RCSF	47
Figure 22 : Interface paramétrage	47
Figure 23 : Interface simulation & visualisation.....	48
Figure 24 : Durées d'une Connexion	49
Figure 25 : Nombre de paquets reçus en fonction de la taille des files d'attente.....	49

Liste des Tableaux

Tableau 1 : Liste des Opérations de l'algorithme	39
Tableau 2 : Description des classes.....	43

Introduction Générale

Les réseaux de capteurs sans fil (RCSFs) viennent au secours de l'environnement et de l'industrie grâce aux récents développements réalisés ces dernières années dans les domaines de la microélectronique et des technologies sans fil (Wireless). Réalisés à partir de petites entités électroniques, ils représentent aujourd'hui une solution crédible quand il s'agit d'observer, de contrôler et éventuellement de traiter des phénomènes physiques tels que la température, la pression, les mouvements ou encore la luminosité. Les applications dédiées à de tels réseaux sont nombreuses et de plus en plus variées, allant de la défense à la surveillance, de la santé à l'environnement et le commerce.

Les constituants (ou nœuds) de ces réseaux sont généralement déployés en grand nombre en vue de collecter et de transmettre des données environnementales, d'une manière autonome, vers un ou plusieurs points de collecte. Les caractéristiques de ces nœuds impliquent diverses contraintes en matière de puissance de calcul, d'espace de stockage mémoire, d'énergie et de communication. Les nœuds dans un RCSF sont souvent amenés à opérer dans des milieux difficiles d'accès, ceci rendant leur recharge en énergie délicate, voire impossible.

Les DTN fonctionnent en mode déconnecté, il n'y a pas en permanence de connexion de bout en bout. Les contacts n'étant pas permanents, il faut réduire au maximum l'interactivité des dialogues entre les terminaux. Les échanges effectués au moyen de TCP/IP, par exemple, sont peu performants dans ce contexte.

Les nœuds ou les passerelles doivent mettre les paquets dans des tampons et les envoyer au prochain saut le plus approprié lorsque la connexion vers ce nœud devient disponible.

Dans cet environnement, les états des connexions changent fréquemment, il n'existe pas a priori de chemin stable entre une source et une destination, la propagation d'informations de routage n'est pas non plus efficace. Donc, les algorithmes traditionnels sont inefficaces, voire ne fonctionnent plus dans les DTN. Il est important de trouver des algorithmes de routage efficaces dans ces situations.

L'objectif de ce travail est d'étudier les algorithmes de routage de type DTN déjà proposés dans la littérature et essayer d'adapter et de simuler un algorithme existant par l'utilisation d'un simulateur pour les réseaux de capteurs sans fil. Ce document s'articule autour de trois chapitres.

Le premier chapitre décrit les généralités sur les réseaux de capteurs sans fil (RCSFs), l'architecture d'un capteur et présente les principes et les caractéristiques des réseaux de capteurs ainsi que ses domaines d'application.

Dans le deuxième chapitre, nous présentons le routage dans les réseaux DTN (Delay tolerant network).

Dans le troisième chapitre nous exposons la conception détaillée de l'algorithme adapté et la démonstration de notre outil prototype à base de java ainsi les résultats de simulation.

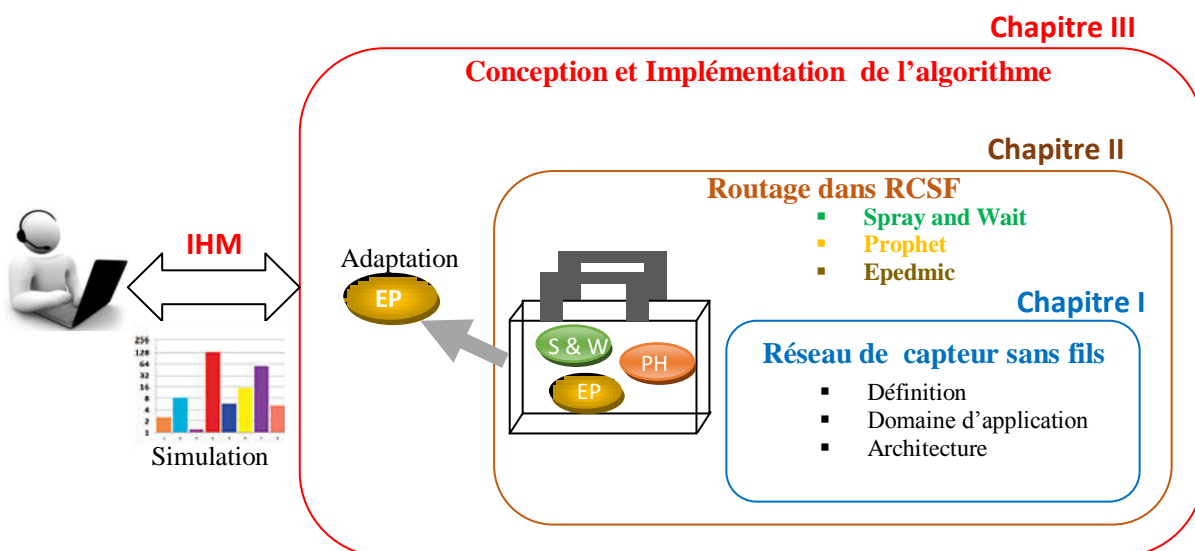


Figure I.1 : Organisation de notre Mémoire

Chapitre I

Généralités sur les réseaux de capteurs sans fil

Chapitre I : Généralités sur les réseaux de capteurs sans fil

Introduction

Dans cette section, on s'intéressera aux réseaux de capteurs sans fils, tout en détaillant leurs aspects de base (fonctionnement, architecture, pile protocolaire et domaines d'application).

1. Définition

Un capteur sans fils est un composant physique autonome, doté de moyens de traitement et de communication de l'information, en plus de leur fonction initiale qui consiste à relever des mesures.

Un réseau de capteurs sans fil –RCSF- (en anglais: Wireless Sensors Network -WSN) est un type particulier de réseaux ad-hoc, constitué d'un ensemble de nœuds capteurs communiquant entre eux à l'aide d'une architecture sans fils multi-sauts afin de surveiller un phénomène physique bien précis (température, pression, mouvement, etc.). La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils peuvent être aléatoirement dispersés dans une zone géographique, appelée « champ de captage » correspondant au terrain d'intérêt pour le phénomène capté [CAS11].

2. Architecture d'un réseau de capteurs sans fil

Un RCSF consiste en un grand nombre de capteurs (des centaines voire des milliers) déployés en densité dans un terrain d'intérêt dit « zone de captage ». Ces capteurs ont la capacité de collecter les données sur le phénomène surveillé et les transmettre par l'intermédiaire d'un routage multi-sauts vers un nœud spécial appelé « puits » ou « station de base » (en anglais : sink). Ce dernier transmet ensuite les informations collectées via Internet ou bien satellite à un ordinateur dit « Gestionnaire de tâches » accessible par l'utilisateur (voir la figure 1).

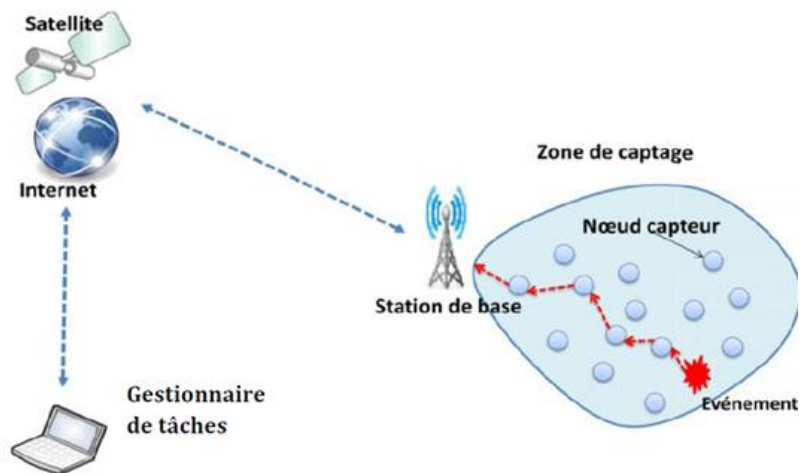


Figure 1 : Architecture simplifiée d'un réseau de capteurs [AMZ11]

3. Les capteurs

3.1. Introduction

La technologie des réseaux de capteurs est la clé de la technologie du futur. Le 'Business Week' a consacré une étude en déclarant qu'ils faisaient partie des « 21 plus importantes technologies du 21^{ème} siècle. ».

Pas chers, périphériques intelligents avec de multiples capteurs, mis en réseau grâce aux liens non filaires (abordés précédemment) et à Internet, et déployés facilement, apportent des opportunités sans précédent pour instrumentaliser et contrôler des maisons, des villes et l'environnement. De plus, les réseaux de capteurs génèrent de nouvelles possibilités pour la reconnaissance et la surveillance ainsi que d'autres applications tactiques.

Les capteurs peuvent être déployés sur le sol, dans l'air, sous l'eau, à l'intérieur de corps et de véhicules et à l'intérieur de bâtiments.

3.2. Architecture physique d'un capteur

Un nœud capteur, tel qu'illustré dans la figure 2, est composé de quatre unités de base: l'unité de capture, l'unité de traitement, l'unité de communication et l'unité d'énergie. Cependant, suivant les applications du capteur, il peut également intégrer des modules supplémentaires tels qu'un système de localisation permettant d'identifier sa position géographique (récepteur GPS, technique de triangulation), un système générateur d'énergie (cellule solaire), ou encore un système mobilisateur pour qu'il puisse se déplacer [AMZ11]. On détaillera dans ce qui suit les quatre unités de base :

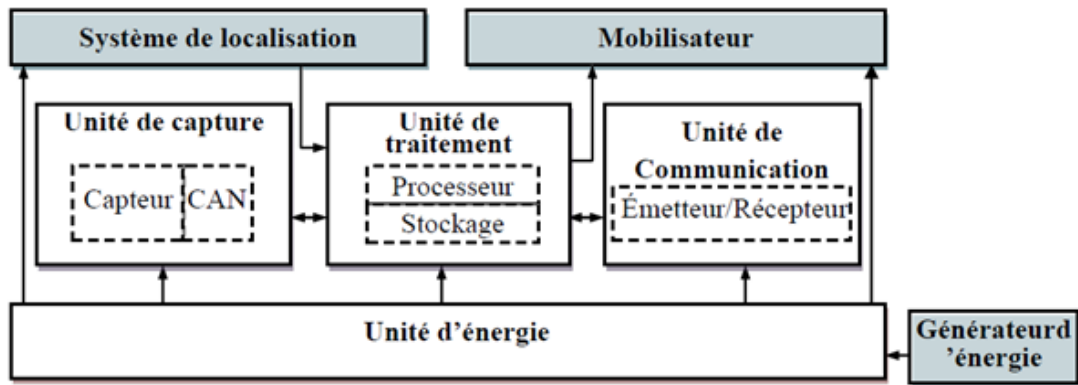


Figure 2 : Architecture d'un capteur [HAM07b]

3.2.1. Unité de capture

Cette unité est généralement composée de deux modules: un capteur et un convertisseur analogique -numérique. Le premier module est chargé de capter et/ou mesurer des grandeurs physiques à partir d'un objet cible. Il communique l'information recueillie sous forme analogique au convertisseur analogique-numérique (ADC : Analog to Digital Converter). L'ADC quant à lui, il transforme l'information en signaux numériques en vue de la transmettre à l'unité de traitement.

3.2.2. Unité de traitement

L'unité de traitement comprend un microcontrôleur dédié, associé à une petite unité de stockage. Elle fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les micro-capteurs. Cette unité fournit aux capteurs la capacité d'exécuter des calculs sur les données et les conserver selon un scénario programmé et assure également la gestion des procédures et des protocoles de communication. Il est à noter qu'il est préférable, mais non obligatoire, de disposer de moyens de reprogrammer les capteurs dans le cas d'un changement dans les exigences de l'application [DUR09].

3.2.3. Unité de communication

Elle est le plus souvent constituée d'un dispositif à base des radios fréquences même s'il arrive parfois d'utiliser des circuits optiques passifs ou actifs. Son rôle est d'assurer la connectivité du capteur au réseau en effectuant toutes les émissions et réceptions de données sur le médium sans fils. Typiquement, le module sans fil d'un nœud capteur a quatre modes opérationnels [HAC03] :

- Idle: le module est allumé et il n'y a ni émission ni réception (écoute).
- Transmit (Tx): le module sans fils est entrain de transmettre un paquet (émission).
- Receive (Rx): le module sans fils est entrain de recevoir un paquet (réception).
- Sleep: le module radio est éteint (en veille).

3.2.4. Unité d'énergie

La consommation d'énergie est un point crucial dans les réseaux de capteurs sans fils. Les capteurs sont généralement munis d'une ressource énergétique à durée de vie limitée. Toute intervention pour recharger ou changer les batteries est la plupart du temps exclue vue la nature de l'environnement. Donc avoir une meilleure gestion de la consommation d'énergie est primordial pour augmenter la durée de vie du réseau.

3.3. La pile de communication (Pile protocolaire)

On peut diviser le procédé de communication dans chaque nœud du réseau indépendamment des différentes fonctions qu'il remplit, en cinq couches standardisées. Ces dernières interagissent entre elles de telle sorte que chaque couche utilise les services de la couche inférieure, et à son tour, fournit des services pour celle supérieure. La pile de communication des RCSFs suit le modèle OSI¹, sauf qu'elle doit assurer de plus la coordination entre les différents nœuds même s'ils sont en nombre de millions. En outre, elle doit respecter les caractéristiques du RCSF, telles que la limitation énergétique et le modèle de communication many-to-one, etc.

Par ailleurs, on définit trois plans de gestion, qui doivent être suivis par les différentes couches de la pile. Tous les nœuds utilisent donc les mêmes protocoles de communication. Cela rend la normalisation des règles de fabrication du matériel et du logiciel une tâche réalisable.

¹ OSI : Open System Interconnection

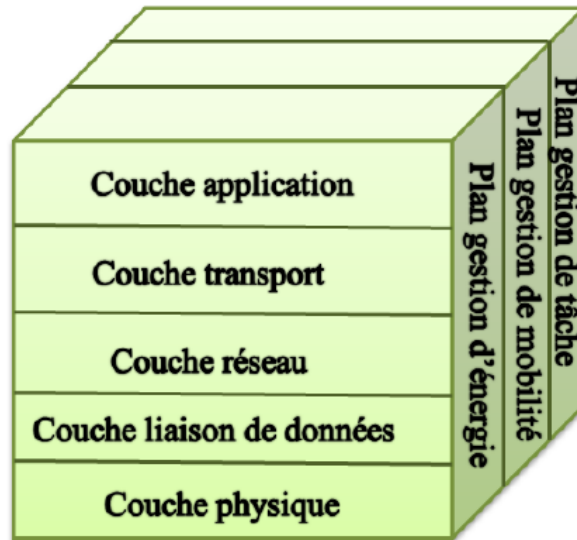


Figure 3 : Pile protocolaire dans les RCSFs [BEN09]

3.3.1. La couche physique

La couche physique gère le support de communication qui achemine les données entre les nœuds capteurs. Effectivement, elle est responsable de la sélection de la fréquence, la génération de la porteuse, la détection du signal, la modulation et le cryptage des données [AKY01, AKY02].

Il existe deux médias pouvant être utilisés pour communiquer dans les réseaux de capteurs sans fils : l'infrarouge et la radiofréquence [ACH05]. Néanmoins, la communication par radiofréquence est la plus utilisée dans les RCSFs, car les paquets transportés sont petits, les débits de transmission sont bas, et la réutilisation de fréquence est considérable [OUL10].

3.3.2. La couche liaison de données

Cette couche se charge essentiellement de multiplexer le flux de données, détecter les trames, contrôler l'accès au média de transmission (MAC : Media Access Control) et contrôler les erreurs. Elle garantit également des connexions point-à-point ou point-à-multipoints fiables. Néanmoins, le type de l'application, la topologie du réseau et les contraintes physique des nœuds, telles que l'énergie et la capacité de traitement des données, déterminent la manière d'accomplir ces fonctionnalités [ILY04].

3.3.3. La couche réseau

Le déploiement dense des nœuds dans le champ de captage et leur portée de transmission réduite, favorisent l'utilisation des protocoles de routage basés sur la

communication multi-sauts. Toutefois, les protocoles de routage multi-sauts conçus pour les réseaux ad-hoc ne peuvent pas répondre aux exigences uniques des RCSFs, notamment dans l'absence d'adressage global, la priorité absolue donnée à la conservation d'énergie et la nécessité de l'agrégation de données [BEC09]. D'où l'intérêt de mettre en place des protocoles propres aux réseaux de capteurs tenant en compte leurs aspects particuliers. En définitif, on peut dire que le rôle principal de la couche réseau est d'assurer le routage de l'information, tout en satisfaisant les contraintes liées au RCSFs.

3.3.4. La couche transport

La couche transport constitue une interface entre la couche application et la couche réseau, les principales fonctionnalités de cette couche sont [ILY04]:

- Le multiplexage/démultiplexage des messages entre les applications et la couche réseau.
- Le contrôle de haut niveau des données.
- La régularisation de la quantité des données injectées dans le réseau.

Cette couche est particulièrement indispensable si le réseau de capteurs est prévu d'être accessible à partir d'autres réseaux externes tels qu'Internet [AKY02]. Ainsi, un utilisateur peut établir une connexion avec le nœud puits via les protocoles TCP ou UDP, par Internet ou par satellite. Cependant toutes les communications entre le puits et les nœuds-capteurs doivent utiliser un protocole de type UDP à cause des limitations de mémoire dans les capteurs [AKY02]. En effet, les datagrammes UDP ont une taille réduite, ce qui économise la bande passante et l'énergie de transmission. De plus, UDP n'utilise aucun acquittement sur les données et n'effectue aucun contrôle sur la bonne transmission de l'information [BEC09].

3.3.5. La couche application

Cette couche constitue l'ensemble des applications implémentées sur un réseau de capteurs. Ces applications doivent fournir des mécanismes permettant à l'utilisateur d'interagir avec le réseau à travers des interfaces, et éventuellement, par l'intermédiaire d'un réseau étendu comme Internet. Elles doivent également rendre transparents le matériel et les logiciels utilisés dans les couches inférieures [HAM07b].

3.4. Consommation d'énergie dans les réseaux de capteurs sans fil

L'énergie consommée par un nœud capteur est due essentiellement aux opérations suivantes : la capture, le traitement et la communication de données [HCB00].

3.4.1. Energie de capture

L'énergie de capture est dissipée pour accomplir les tâches suivantes : échantillonnage, traitement de signal, conversion analogique/numérique et activation de la capture.

En général, l'énergie de capture représente un faible pourcentage de l'énergie totale consommé par un nœud.

3.4.2. Energie de traitement

L'énergie de traitement se divise en deux parties : l'énergie de commutation et l'énergie de fuite. L'énergie de commutation est déterminée par la tension d'alimentation et la capacité totale commutée au niveau logiciel (en exécutant un logiciel). Par contre l'énergie de fuite correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement. En général, l'énergie de traitement est faible par rapport à celle nécessaire pour la communication.

3.4.3. Energie de communication

L'énergie de communication se décline en deux parties : l'énergie de réception et l'énergie de l'émission. Cette énergie est déterminée par la quantité des données à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio. L'émission d'un signal est caractérisée par sa puissance. Quand la puissance d'émission est élevée, le signal aura une grande portée et l'énergie consommée sera plus élevée. Notons que l'énergie de communication représente la portion la plus grande de l'énergie consommée par un nœud capteur.

3.5. Le niveau de gestion d'énergie

Un capteur est doté généralement d'une ressource énergétique limitée sous forme d'une batterie standard de basse tension ($< 0.5 \text{ AH}$, 1.2 V). Dans la plus part des cas, il est impossible de recharger ou de remplacer cette batterie. Ainsi, la durée de vie d'un nœud va dépendre fortement de sa consommation d'énergie.

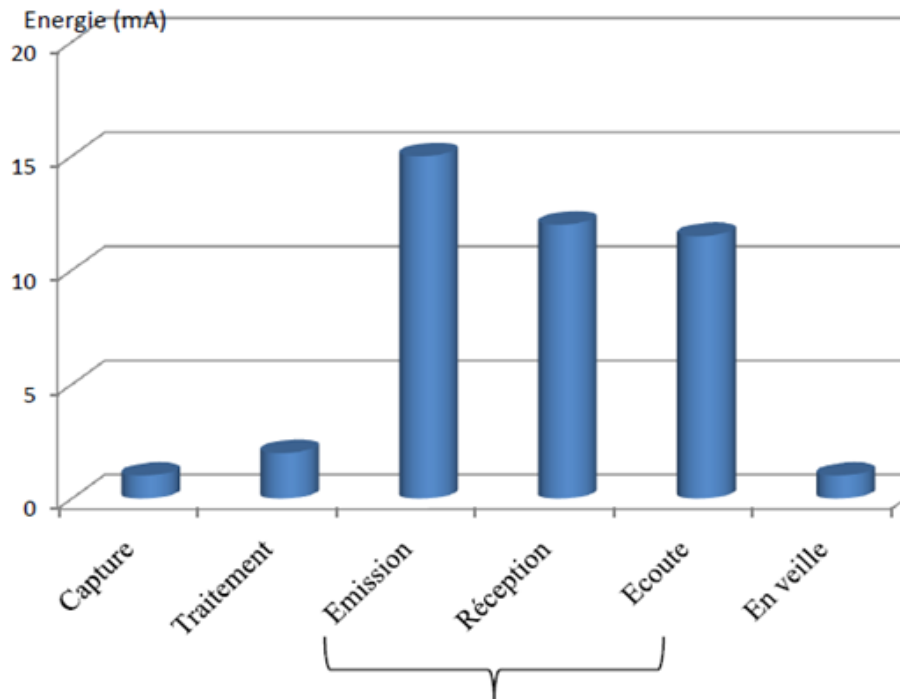


Figure 4 : Répartition de la consommation d'énergie dans un capteur [POM04]

La figure 4 illustre la répartition de la consommation d'énergie au sein d'un capteur. On constate que les opérations de communication consomment la plus grande quantité d'énergie. C'est pour cela, maximiser la conservation d'énergie lors de la communication va augmenter, de façon très significative, la durée de vie du capteur. De manière générale, le niveau de gestion d'énergie adopte des mécanismes pour contrôler et essayer d'optimiser l'énergie consommée par les capteurs tels que : la mise en veille après la réception d'un message ; la non-participation au routage des nœuds qui atteignent un seuil critique d'énergie.

3.6. Domaines d'application

La taille réduite des capteurs, leur coût réduit et la large gamme disponible (thermique, chimique, cinétique, optique, etc.) ont permis au RCSF d'envahir rapidement plusieurs domaines applicatifs. Parmi les domaines où ces réseaux offrent de meilleures contributions, on cite les domaines : militaire, environnemental, industriel, de la santé et de la sécurité. Des exemples d'applications potentielles dans ces différents domaines sont exposés ci-dessous [CHA10, HAM07b]:

3.6.1. Les applications militaires

A l'instar de plusieurs technologies, le domaine militaire fut le moteur initial pour le développement des réseaux de capteurs. Le déploiement rapide, l'auto-configuration et la tolérance aux pannes sont des caractéristiques qui font de ce type de réseaux un outil appréciable dans un tel domaine. Les RCSFs peuvent être utilisés dans le cadre militaire, par exemple, pour :

- Contrôler des forces, des équipements et des munitions;
- Analyser et surveiller un champ de bataille;
- Détecter et poursuivre des objets ennemis;
- Protéger des éléments sensibles comme les bases nucléaires et les ponts;
- Détecter et reconnaître des attaques nucléaires, biologiques et chimiques.

3.6.2. Les applications liées à la sécurité

L'application des réseaux de capteurs dans le domaine de la sécurité pourrait diminuer considérablement les budgets consacrés à la sécurisation des lieux et à la protection des êtres humains tout en garantissant des résultats plus fiables. Comme exemples de ce type d'applications, on peut citer :

- La détection des altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement, par des capteurs intégrés dans les murs ou dans le béton;
- La protection des barrages par la détection prompte des fuites d'eau;
- La surveillance des mouvements en constituant un système de détection d'intrusions distribué. Cet aspect va rendre plus complexe la possibilité de mettre hors d'usage ce système de surveillance.

3.6.3. Les applications environnementales

La surveillance de l'environnement est le domaine qui utilise le plus la technologie des réseaux de capteurs. Des applications typiques peuvent être mentionnées :

- La détection et la prévention contre les désastres comme les inondations, la pollution et notamment les feux de forêts en déployant, dans ce cas, des thermo-capteurs

Adaptation et la simulation d'un algorithme de type DTN (delay to pour les réseaux de capteurs sans fil

capables d'évaluer les risques d'un éventuel début d'incendie grâce à des relevés périodiques de température;

- Dans l'agriculture, les contrôles du climat et de l'état de la terre permettent d'augmenter la précision et l'efficacité des activités agricoles telles que l'irrigation et l'utilisation d'engrais;
- L'étude et l'observation de la faune et de la flore dans les grandes forêts inaccessibles ou dans les profondeurs sous marines où la présence de l'homme est quasi impossible;
- L'exploration et la surveillance des environnements hostiles tels que les volcans ou les régions toxiques.

3.6.4. Les applications médicales

L'utilisation des réseaux de capteurs dans le domaine de la médecine peut apporter une surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité. Ceci est rendu possible grâce à l'utilisation de micro-capteurs qui pourront être avalés ou implantés sous la peau. Parmi les applications d'avant garde de ce domaine on peut énumérer :

- Les micro-caméras qui peuvent être ingérées et sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain ;
- La création d'une rétine artificielle composée d'une centaine de micro-capteurs pour améliorer la vision.

3.6.5. Les applications industrielles

Grâce à leur grande flexibilité, les réseaux de capteurs sont largement utilisés dans le champ industriel. Parmi les cas d'utilisations, on cite :

- Le suivi et le contrôle du procédé de production pour assurer une meilleure qualité de service;
- La protection contre les catastrophes industrielles en assurant un suivi efficace des équipements et produits sensibles tout en détectant précocement les problèmes;

L'observation d'un site susceptible de subir les effets d'une pollution et la construction en temps réel d'une cartographie de sa contamination grâce à des capteurs, disséminés sur le site, capables de relever ces niveaux pollués.

4. Conclusion

Dans ce chapitre nous avons présenté les RCSFs sont de plus en plus indispensables dans tout domaine où récolter l'information à partir de l'environnement est la base pour prendre une décision. Ces réseaux spécifiques ont leurs propres caractéristiques et limitations, notamment la limitation d'énergie, du traitement et du stockage. De ce fait, appliquer les protocoles et traitements des réseaux classiques sur ce type de réseau est inefficace. Par ailleurs, de nouveaux protocoles de routage qui répondent aux limitations de la technologie des RCSFs sont mis en place. Une étude de routage dans les réseaux DTN (Delay tolerant network) fera l'objectif du chapitre suivant.

Chapitre II

Le routage dans les réseaux DTN (Delay Tolerant Network)

Chapitre II : Le routage dans les réseaux DTN

(Delay Tolerant Network)

Introduction

On qualifie de réseaux tolérants au délai ou Delay Tolerant Networks (DTN) les réseaux dont le délai de bout en bout entre deux nœuds peut devenir très grand.

Les DTN fonctionnent en mode déconnecté, il n'y a pas en permanence de connexion de bout en bout. Les contacts n'étant pas permanents, il faut réduire au maximum l'interactivité des dialogues entre les terminaux. Les échanges effectués au moyen de TCP/IP, par exemple, sont peu performants dans ce contexte.

Les nœuds ou les passerelles doivent mettre les paquets dans des tampons et les envoyer au prochain saut le plus approprié lorsque la connexion vers ce nœud devient disponible.

Dans cet environnement, les états des connexions changent fréquemment, il n'existe pas a priori de chemin stable entre une source et une destination, la propagation d'informations de routage n'est pas non plus efficace. Donc, les algorithmes traditionnels sont inefficaces, voire ne fonctionnent plus dans les DTN. Il est important de trouver des algorithmes de routage efficaces dans ces situations.

L'objectif de ce travail est d'étudier les routages déjà proposés dans la littérature, essayer d'adapter certains algorithmes et évaluer ces modifications au moyen d'un simulateur.

Ce chapitre est constitué d'une présentation des réseaux DTN² et la description des algorithmes de routage dans les DTN.

1. Les réseaux tolérants aux délais (DTN)

Les réseaux traditionnels, comme l'Internet actuel, sont basés sur l'hypothèse que certaines conditions sont remplies : un chemin de bout en bout doit exister et être stable, les nœuds doivent toujours être alimentés en énergie, la bande passante doit être suffisante.

Cependant ces conditions sont très difficiles à obtenir dans certains scénarios, comme la liaison entre des satellites et le sol, la communication entre plusieurs planètes dans laquelle

² DTN : Delay Tolerant Network

les délais de transmissions s'allongent ou, dans les réseaux sans fils, lorsque les nœuds sont placés de façon très éparse et donc ne peuvent pas toujours communiquer entre eux.

Les réseaux DTN désignent des réseaux capables de transmettre des informations de bout en bout, même lorsque le réseau n'est pas connecté en permanence. La figure ci-dessous représente un exemple de DTN.

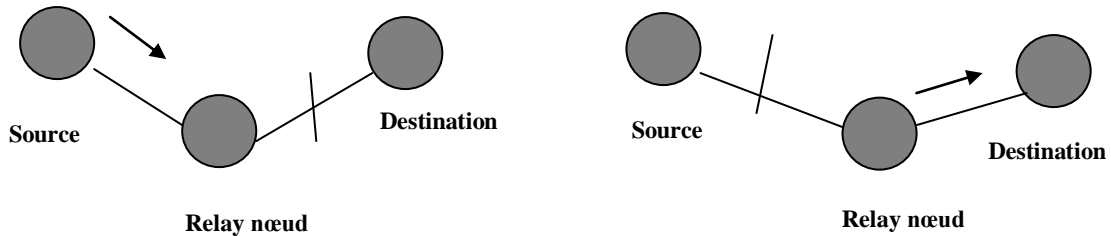


Figure 5 : Transfert en utilisant un nœud intermédiaire

En fonction de sa mobilité, soit le Nœud Relais est connecté à la source, soit il est connecté à la destination mais pas aux deux simultanément. Donc, il n'y a pas de connexion de bout en bout. Certains protocoles, TCP/IP par exemple, ne fonctionnent plus dans ces conditions.

1.1. Définition

Un réseau tolérant aux délais est un réseau de plusieurs réseaux régionaux, c'est un overlay au-dessus de ces réseaux régionaux incluant le réseau Internet. Un DTN supporte l'interopérabilité entre les réseaux [WAR03] :

- En s'accommodant de longs délais entre (ou dans) les réseaux régionaux.
- En traduisant les caractéristiques de communication entre les réseaux régionaux.

Les DTNs accommodent la mobilité et l'énergie limitée des appareils de communication sans fil.

Notons que, les technologies sans fil DTN sont diverses, nous trouvons alors :

- La radio fréquences (RF) : Une forme de communication sans fil qui permet de transmettre l'information d'un terminal à une station de base, qui à son tour la transmet à un ordinateur hôte.
- L'UWB9 : Une technologie radio ultra large bande, utilisée pour la communication haut débit sur courte distance, et avec une très faible puissance. La bande passante UWB est définie comme ayant une largeur d'au moins 500 MHz.

- La liaison dégagée optique (Laser) : Une technologie qui a un avantage économique certain sur les solutions filaires. Où la seule condition pour l'installer est de garantir une vue dégagée des obstacles entre les deux points. Ce qui impose des émetteurs/récepteurs sur des points hauts, des fixations fiables interdisant tout mouvement des matériels, et l'absence d'éléments perturbateurs tels que : le flux d'air d'une bouche d'aération, ou la poussière.

1.2. Architecture d'un DTN

Le DTN résout tous les problèmes rencontrés par les protocoles bout à bout en utilisant la méthode store-and-forward message switching. Il s'agit de transmettre, le long du chemin voulu, les informations d'une zone de stockage à une autre comme le montre la figure 2. Cette opération nécessite des zones de stockage, telles des disques durs, permettant de stocker des messages pendant des jours, et non de simples allocations dynamiques de mémoire de quelques millisecondes comme cela aurait pu être le cas avec une autre méthode. De telles capacités de mémoire sont nécessaires car il se peut qu'un lien entre deux nœuds consécutifs ne soit pas disponible pour une longue période. En outre, si une erreur se produit, le message devra être mémorisé dans ces zones afin de pouvoir être retransmis.



Figure 6 : Store-and-Forward Message Switching

L'architecture DTN met en œuvre le store-and-forward message switching par l'ajout d'une nouvelle couche protocolaire, située entre la couche transport et la couche application, appelée bundle layer dont la fonction principale est de permettre à une application de communiquer à travers différentes régions. On parle alors de bundle protocol. Plus précisément, une seule bundle layer est utilisée à travers les différents réseaux traversés, tandis que les couches qui se trouvent en dessous de la bundle layer sont quant à elles spécifiques à chaque région comme le montre la figure 7.

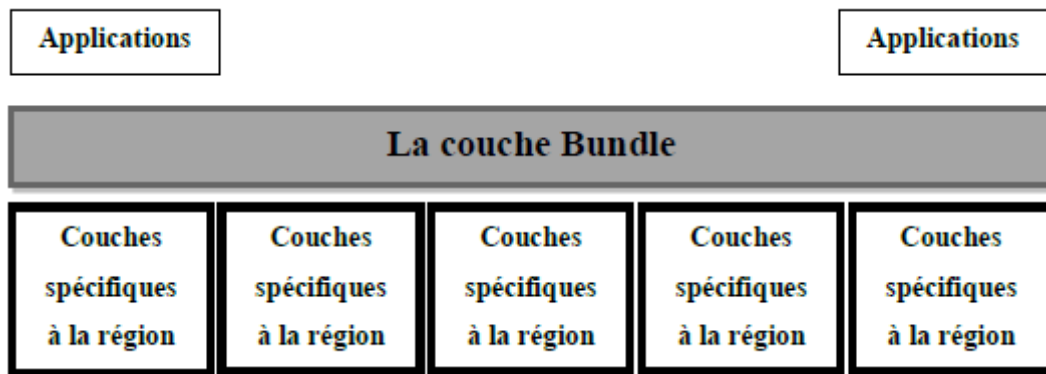


Figure 7 : Emplacement de la couche Bundle dans la pile protocolaire [WAR03]

La couche bundle est utilisée par tous les réseaux (ou les régions) qui constituent le DTN, tandis que les couches inférieures (de la couche transport à la couche physique) sont choisies selon l'environnement de chaque région (Figure 8).

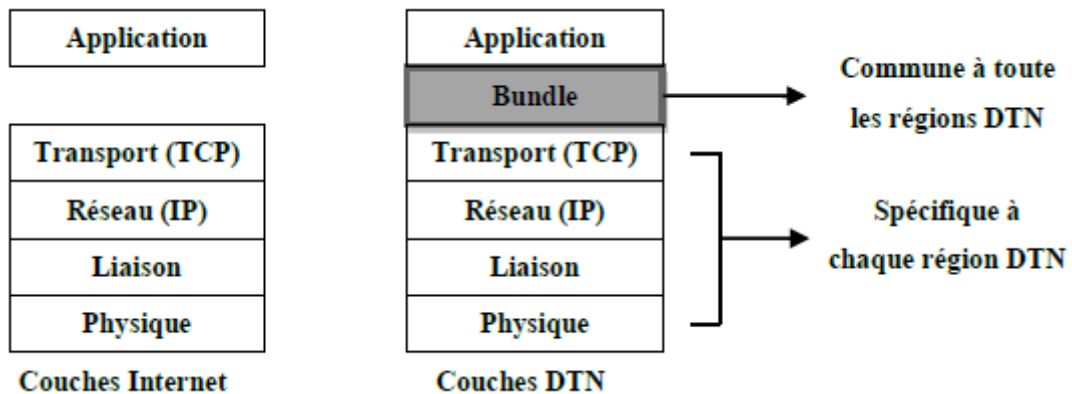


Figure 8 : La couche Bundle présente dans toutes les régions [WAR03]

Pour des transmissions de données qui peuvent être très longues, le bundle protocol fonctionne différemment de TCP et évite de multiplier les allers-retours. Ainsi, l'acquittement n'est pas obligatoire et dépend de la classe de service sélectionnée parmi les suivantes :

- Custody Transfer
- Return Receipt
- Custody-Transfer Notification
- Bundle-Forwarding Notification
- Priority of Delivery (en masse, normale, accélérée)

- Authentification (afin de vérifier l'identité de l'émetteur et pour vérifier l'intégralité du message)

La figure 9 ci-dessous met en valeur les différents types de classes de services.

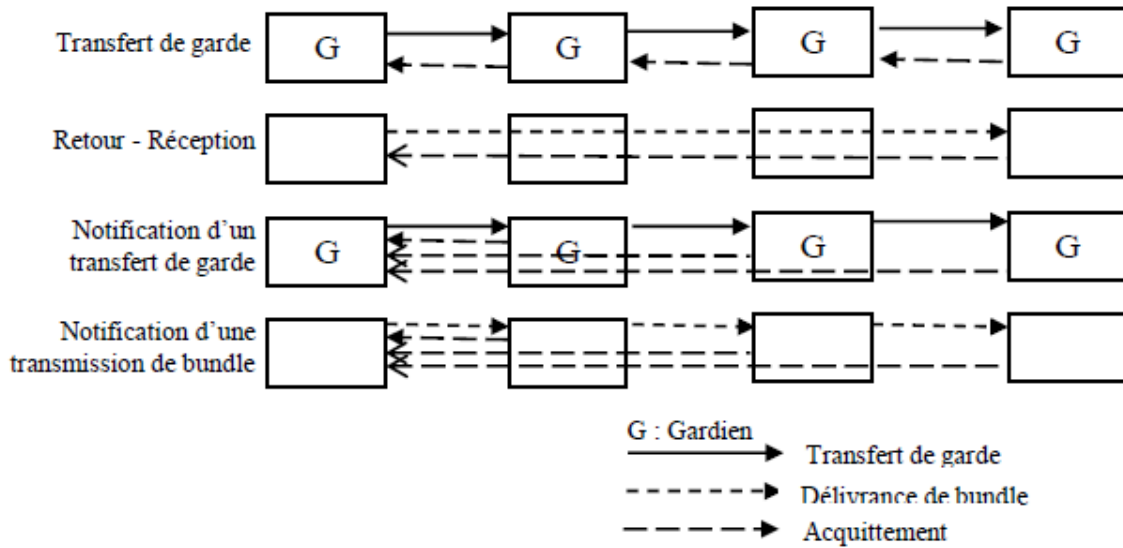


Figure 9 : Les options de livraisons offertes par la couche bundle. [KRI08]

1.2.1. Les entités de communication

Comme schématisé sur la Figure 10, un réseau DTN est composé d'un ensemble d'entités communicantes appelées : Nœuds. Ces nœuds sont répartis en régions [§1.2.1.2], où chacun d'entre eux est uniquement identifié par au moins un tuple [§1.2.1.3] contenant le nom de la région [§1.2.1.2] et le nom de l'entité. Un nœud lié à plusieurs régions doit avoir au moins un tuple [§1.2.1.3] d'identification pour chacune des régions auxquelles il appartient

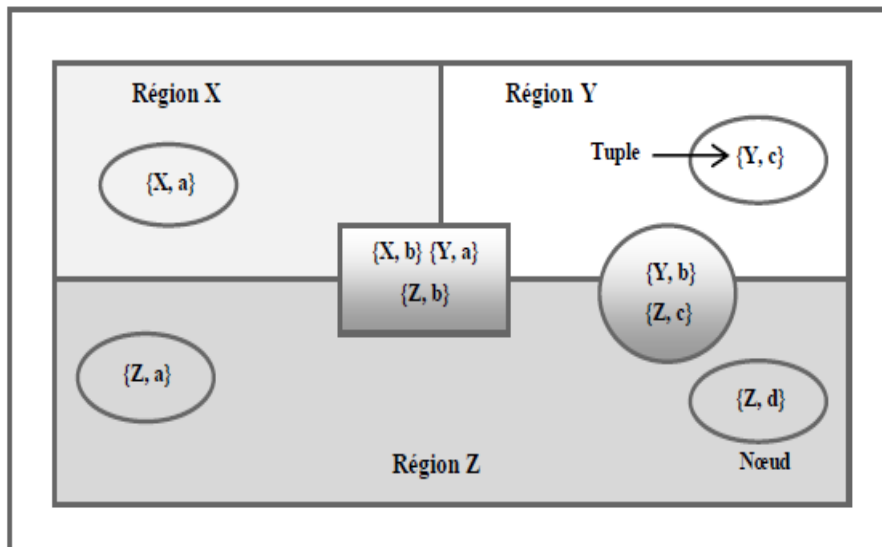


Figure 10 : l'appartenance des nœuds aux différentes régions [WRVS07]

1.2.1.1. Les nœuds DTN

Un nœud DTN est alors un dispositif pour l'envoi et la réception de messages (appelés aussi : bundles). Il peut jouer le rôle de : source, destination ou de nœud intermédiaire pour la transmission de bundles.

Le nom du nœud DTN lui-même, par opposition à une application l'utilisant est défini dans une « région spécifique » à l'aide de l'identifiant de l'entité ou une partie de celui-ci.

Un nœud DTN peut jouer le rôle de [WFW03]:

- Hôte :

Envoi et/ou reçoit les bundles, mais ne les diffuse pas. Ce qui requiert un stockage persistant durant de longs délais dans lesquels les bundles seront alignés jusqu'à ce que les liens soient disponibles.

- Routeur :

Diffuse les bundles au sein d'une seule région DTN et peuvent optionnellement jouer le rôle d'hôte.

- Passerelle :

Diffuse les bundles entre deux ou plusieurs régions DTN et peut optionnellement jouer le rôle d'un hôte. Elle opère sur la couche transport et se base sur la commutation de messages plutôt que sur la commutation de paquets. Cependant, elle fournit l'interopérabilité entre des protocoles spécifiques pour une région et ceux spécifiques pour une autre.

Ces différents rôles peuvent être schématisés comme il est illustré dans la Figure 11 ci-dessous

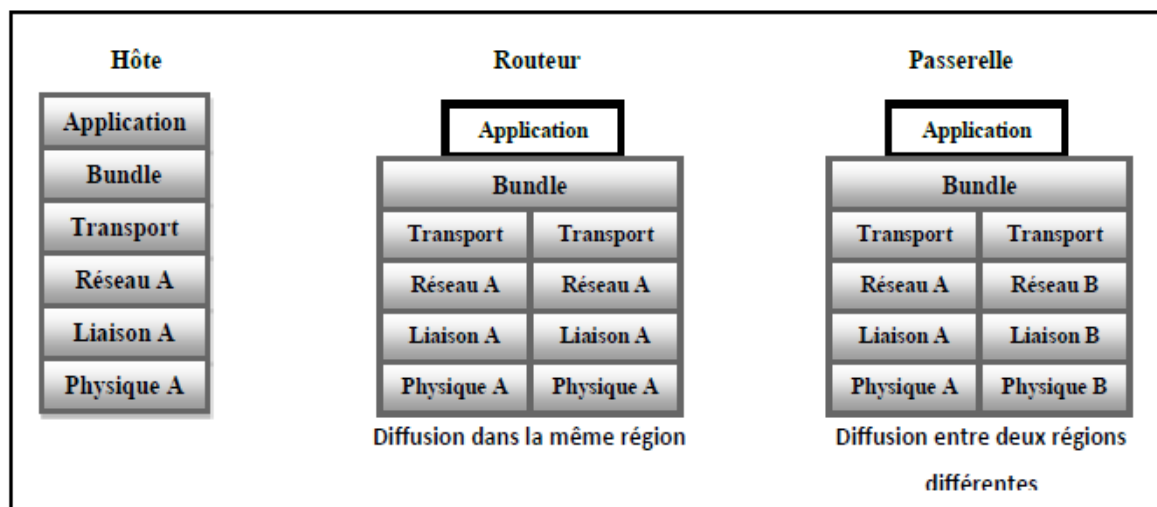


Figure 11 : Les différents rôles d'un nœud DTN [WFW03]

1.2.1.2 Les régions DTN

L'architecture DTN définit un réseau de plusieurs réseaux où chacun d'eux représente une région dans laquelle les caractéristiques de communication sont homogènes. Une région peut être le réseau Internet du globe terrestre, un réseau tactique militaire, la planète ou même un vaisseau spatial. En d'autres termes, une région est une zone qui est influencée par les familles de protocoles, les dynamiques de connexion, les politiques administratives ou de manière générale, les « régions DTN » sont délimitées en se basant sur un critère appelé : les frontières de confiance [WFW03].

Chaque région DTN a un nom unique et connu, ou que l'on peut connaître parmi toutes les autres régions du DTN. Ainsi, un référentiel pour l'ensemble des noms des régions est nécessaire, qui grâce à lui l'on peut retrouver le nom d'une région à tout moment.

Les bundles DTN (appelés aussi : messages) originaires de régions différentes de celle de destination sont transmis en premier lieu via des entités communicantes appelées : Passerelles, qui connectent la région source à une ou plusieurs autres régions. Le routage à l'extérieur de la région destination n'est fondé que sur le nom de celle-ci et non pas sur le nom complet de la destination elle-même.

Les régions DTN sont caractérisées par [CBH03]:

- Chaque région doit avoir un espace identifiant partagé par tous les nœuds de la région, et doit spécifier des conventions de nommage internes afin d’être employées pour l’identification des entités.
- Chaque nœud membre de la région est doté d’un unique identifiant tiré de cet espace identifiant. Notons que pour certains types de régions, un « nœud » peut être composé d’une collection d’éléments calculables et/ou géographiquement distribués. Un seul et unique élément s’applique sur le nœud destiné à recevoir des données provenant des autres nœuds DTN.
- Pour être considéré comme membre de la région, chaque membre potentiel de celle-ci doit être capable d’atteindre les autres membres de la même région, sans passer par d’autres nœuds DTN se trouvant à l’extérieur de celle-ci en utilisant un ou plusieurs protocoles connus au niveau de chaque nœud.
- Un nœud DTN ne doit pas nécessairement être atteint directement. Ceci peut demander une opération de Store and Forward [§ 4.4.2.2] et/ou de transmission par les autres nœuds de la même région.

1.2.1.3 Les tuples

Un tuple désigne le nom d’un nœud. Il est composé, comme illustré dans la Figure 12, de deux parties :

- L’identificateur de la région (ou nom de région).
- L’identificateur de l’entité (ou nom d’entité).

Le nom de région est nécessaire et suffisant pour router un bundle de données à sa région de destination, mais ne peut pas le délivrer au point de destination spécifique auquel il a été destiné.

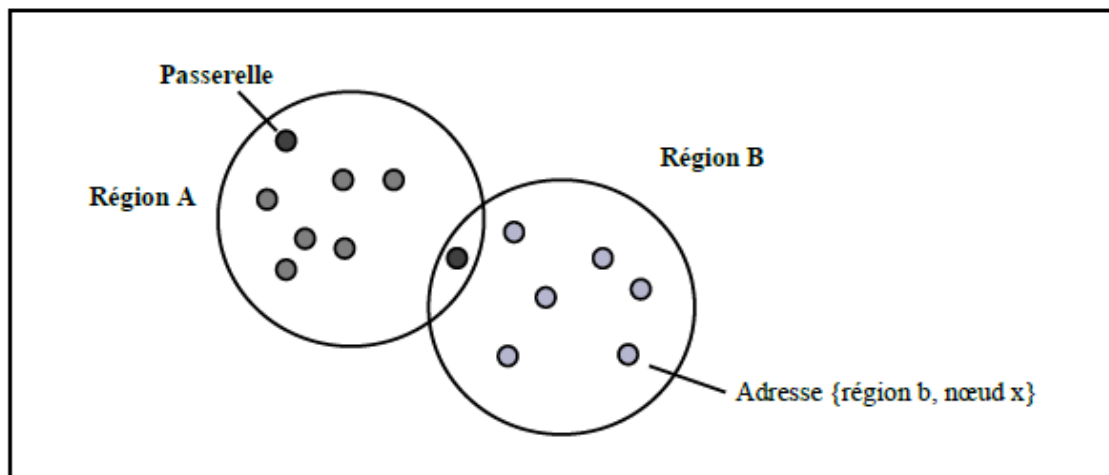


Figure 12 : Adressage d'un nœud dans un DTN [WRVS07]

2. Algorithmes de routage

Dans les réseaux traditionnels, le routage consiste à trouver le meilleur chemin vers la destination en s'appuyant sur une certaine métrique. Dans les réseaux DTN, la notion de chemin, de topologie, est différente puisque les contacts peuvent être intermittents et les nœuds mobiles. On a une topologie variable dans le temps. Les protocoles de routage traditionnels se révèlent inefficaces.

Dans les réseaux DTN, les objectifs du routage sont :

1. De diminuer le taux de perte ou d'échec dans l'acheminement des paquets
2. De diminuer le délai de bout en bout
3. De diminuer le nombre total de transmissions nécessaires à la transmission d'un paquet

Le but premier du routage est de transmettre le paquet d'une source à une destination. Donc, le taux de perte est toujours le paramètre le plus important pour le routage. Cependant, pour certaines applications, un paquet qui arrive trop tard n'a aucune valeur. On doit donc réduire le délai de bout en bout, ce qui est une tâche très difficile. La plupart des réseaux DTN sont les réseaux sans fil, c'est-à-dire que les liens sont peu fiables, la bande passante des liens est faible, les files d'attente ne sont pas importantes et les nœuds doivent économiser leur énergie. Donc, pour améliorer la performance du système, il faut réduire le nombre total de transmission de paquets et le nombre de copies de chaque paquet ainsi que minimiser la charge liée à l'échange des informations de routage.

2.1. Routage non Géographique

Ce paragraphe présente les algorithmes aléatoires et les algorithmes probabilistes. Ces deux catégories d'algorithmes sont basées sur Epidemic Routing.

2.1.1. Epidemic Routing

Epidemic Routing [AV00] est le plus connu et le plus simple algorithme dans ce domaine. Chaque nœud a une seule file d'attente. Les messages originaires de ce nœud et d'autres nœuds sont insérés dans la même file d'attente. Chaque nœud crée un sommaire en utilisant une fonction de hachage. Ce sommaire contient la liste de tous les messages possédés par ce nœud. Quand 2 nœuds A et B se rencontrent (i.e. ils peuvent communiquer directement), ils lancent une session qui s'appelle Anti-Entropie. Pendant la session, ils s'échangent leurs sommaires. Chaque nœud compare le sommaire qu'il reçoit avec son propre sommaire pour trouver les messages qu'il ne possède pas. Par exemple, le nœud A compare son sommaire à celui reçu du nœud B et détermine les messages que A n'a jamais eu. Ensuite, le nœud A demande à B de les transmettre. Ici on suppose que la file d'attente n'est pas limitée. Pour éviter de re-faire cette session entre ces 2 nœuds sans arrêt, chaque nœud possède une liste des nœuds qui contient les identifiants des nœuds rencontrés récemment.

Cet algorithme ressemble à une inondation dans les réseaux traditionnels. Il ne maintient pas de table de routage, c'est-à-dire, qu'il ne se préoccupe pas la topologie, il peut donc fonctionner sur plusieurs topologies. Quand il y a suffisamment de ressources (file d'attente, BP...), les messages parviennent à leurs destinations via le meilleur chemin, avec le délai le plus petit. On peut ajouter quelques paramètres pour contrôler la propagation, un TTL par un exemple. L'inconvénient de cette solution est évident, elle consomme trop de ressources. Sa performance est très mauvaise s'il n'y a pas suffisamment de ressources, notamment en termes de files d'attente.

2.1.2. Calculs de prédictibilité

Pour augmenter l'efficacité de cette solution, beaucoup d'algorithmes sont possibles. En général, les mouvements des nœuds dans un réseau DTN ne sont pas totalement aléatoires. Si 2 nœuds se sont rencontrés fréquemment, on peut considérer qu'il y a une grande probabilité de ces 2 nœuds se rencontrent bientôt à nouveau. On utilise ces informations de rencontre pour estimer quels nœuds sont les plus intéressants comme prochain saut d'un message.

Les algorithmes qui utilisent les informations de rencontres de nœuds et calculent la prédictibilité en fonction des destinations sont les plus nombreux et les plus intéressants.

La prédictibilité est la probabilité qu'un message arrivé à la destination via un nœud intermédiaire donné. Chaque nœud possède une liste de toutes les destinations (éventuellement tous les nœuds) avec les prédictibilités associées. Il existe beaucoup de méthodes pour calculer ces valeurs afin de créer la liste. Dans certaines méthodes, ces valeurs peuvent être plus grandes que 1.

Il y a une méthode très connue qui est utilisée par PROPHET [LDS03], [SPR04] et décrite par les équations ci-dessous :

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) \times P_{init} \longrightarrow (1)$$

$$P_{(a,b)} = P_{(a,b)old} \times \gamma^k \longrightarrow (2)$$

$$P_{(a,c)} = P_{(a,c)old} + (1 - P_{(a,c)old}) \times P_{(a,b)} \times P_{(b,c)} \times \beta \longrightarrow (3)$$

P_{init} est une constante, il s'agit de la prédictibilité initiale prise par tous les nœuds qui se rencontrent pour la première fois. Cette constante vaut 0,75 dans les simulations présentées ci-après. $P_{(a,b)}$ est la probabilité d'un nœud a rencontre un nœud b. γ ($0 < \gamma < 1$) est une constante qui définit la valeur dont doit être décrétementée la prédictibilité par unité de temps. K est la durée depuis la dernière renouvellement de $P_{(a,b)}$ par l'équation (2). Les auteurs proposent de donner à γ une valeur de 0,98. En pratique cette valeur dépend de la mobilité et de la densité de nœuds. β ($0 < \beta < 1$) est une constante reflétant l'influence de la transitivity. Il est petit, 0,25 par exemple.

Quand les nœuds a et b se rencontrent, le nœud a utilise la formule (1) et la formule (3) pour renouveler les valeurs de prédictibilité. Selon (1), a affecte à ces nœuds une prédictibilité de plus en plus grande aux nœuds qu'il rencontre souvent, Selon (3), si a rencontre b très fréquemment et si b a une très grande prédictibilité pour c (un autre nœud), a obtient aussi une très grande prédictibilité pour c. La formule (2) se calcule pour chaque période k de temps, ce qui contribue à diminuer prédictibilité des nœuds rencontré peu fréquemment par a.

Une autre méthode qui est proposée par [DFL01] ressemble beaucoup à la méthode précédente.

$$M_{t+1}(A, C) = \begin{cases} \lambda M_t(A, C) & \text{if none co-located} \\ \lambda M_t(A, C) + 1 & \text{if } C = B \\ \lambda M_t(A, C) + \alpha M_t(B, C) & \text{for all } C \neq B \end{cases}$$

On définit $M_t(A,C)$ la prédictibilité enregistrée par le nœud A pour le nœud C. M_t sera renouvelé à chaque période de temps, période de temps qui est assez courte. La nouvelle valeur est M_{t+1} . Donc, le nœud peut rencontrer un nœud au plus dans une période. λ ($0 < \lambda < 1$) et α ($0 < \alpha < 1$) sont respectivement les coefficients du temps et de la transitivité. Les valeurs de la prédictibilité sont entre 0 et $1/(1-\lambda)$. C'est-à-dire qu'il est possible d'avoir les valeurs plus grandes que 1 pour la prédictibilité.

La suivante est une autre méthode qui est utilisée dans MaxProp [BGJ06]. Quand le nœud A rencontre le nœud B, A augmente sa prédictibilité pour B de $k=1$ unités, après A re-normalise toutes ses prédictibilités pour obtenir une somme de toutes ses prédictibilités égale à 1.

Prenons un exemple dans lequel le réseau comporte 4 nœuds. Au départ, A a les prédictibilités suivantes pour les autres nœuds : $P(A,B)=P(A,C)=0.3$ et $P(A,D)=0.4$. Quand il rencontre B, il modifie $P(A,B)=0.3+k=1.3$ ($k=1$), $P(A,C)=0.3$ et $P(A,D)=0.4$. Après la re-normalisation, $P(A,B)=0.65$, $P(A,C)=0.15$ et $P(A,D)=0.2$.

Dans cette méthode, La prédictibilité est modifiée uniquement au moment où des nœuds se rencontrent. Au contraire, pour les méthodes précédentes, la prédictibilité diminue lorsqu'un nœud ne rencontre aucun autre nœud pendant un certain temps. Dans [BGJ06], les auteurs prennent $k=1$, ce qui semble très grand. Quand le nœud rencontre un nœud pour lequel il a une prédictibilité faible, une telle modification de la prédictibilité est trop importante. Elle donne une très grande valeur tout de suite au nouveau nœud rencontré juste après la renormalisation. Donc un k plus petit semble plus approprié.

2.1.3. Stratégie de la transmission

On a des valeurs de prédictibilité maintenant. Comment utiliser ces informations pour obtenir une meilleure performance du routage ? Il semble évident qu'il faut transmettre les messages vers le nœud qui a une plus grande prédictibilité en fonction de la destination.

PROPHET utilise une stratégie très simple. Quand un nœud rencontre un autre nœud qui a une prédictibilité plus grande pour le message, il transmet ce message à ce nœud, et garde encore ce message pour le transmettre à d'autres nœuds dans le futur.

Le problème est que, lorsque le nœud original du message a une très faible prédictibilité pour la destination, il va transmettre ce message vers tous les nœuds qu'il rencontre, jusqu'à ce message soit supprimé par la file d'attente (souvent, le nœud original garde ses messages pendant longtemps). Ce processus prend beaucoup de ressources, même si l'on peut ajouter un contrôle afin d'éviter de transmettre le message au même nœud plusieurs fois. Par contre,

si le nœud original de message a une très grande prédictibilité, ce message va être distribué rarement, très peu de nœud pouvant avoir une prédictibilité plus grande que lui. Ce message va donc rester dans le nœud original longtemps. Par exemple, le nœud original vient de rencontrer la destination, et part plus loin. Sa prédictibilité pour la destination diminuera lentement.

Dans MV [BBL05], le nœud n'envoie que n copies d'un message au maximum à chaque rencontre. Par un exemple, si $n=8$, lorsque le nœud A rencontre le nœud B, 20 messages de A ont une prédictibilité plus petite que celle de B. Dans PROPHET, A va transmettre tous ces 20 messages vers B. Mais, ici A ne peut envoyer que 8 messages. Dans ces conditions, A choisira les 8 messages (parmi les 20 messages) qui ont plus grandes prédictibilités dans B.

L'avantage de MV est de transmettre moins de messages à chaque rencontre. Cette stratégie est plus appropriée pour les nœuds qui ne sont pas très puissants ou qui se déplacent très vite. En effet, ces nœuds n'ont pas assez de puissance, leur bande passante est faible ou la durée de rencontre est très limitée pour envoyer tous les messages.

L'inconvénient est qu'un message transporté par un nœud loin de la destination a très peu de chance d'être choisi et envoyé. Bien que ce message soit choisi dans les 20 messages, il a très peu de chances d'être choisi dans les 8 messages qui vont être transmis. Un nœud rencontré aura des difficultés à avoir une assez grande prédictibilité pour ce message permettant de lui donner la priorité devant les autres messages.

Un autre algorithme intéressant s'appelle Spray and Wait [SPR04]. Dans cet algorithme, il n'y a pas de table de routage. L'algorithme comporte 2 phases, Spray et Wait. Pendant la phase spray, le nœud original envoie rapidement L copies de message à L autres nœuds quelconques. Pendant la phase Wait, chaque nœud parmi ces L nœuds garde le message (ne l'envoie plus) jusqu'à ce qu'il rencontre directement la destination (la copie peut cependant être supprimée si la file d'attente est pleine). Donc, chaque message va ne toucher que L nœuds intermédiaires. L dépend de la dimension du réseau et du délai moyen que l'on veut obtenir.

Pour faire la phase Spray plus efficacement, Binary Spray and Wait [SPR04]. a été proposé. Le nœud original démarre avec L copies du message à transmettre. Il peut envoyer plusieurs copies du même message à un même nœud. Si un nœud possède plus d'une copie, A avec n copies par exemple, quand il rencontrera un autre nœud B qui n'a pas de copie, A pourra

envoyer $\lfloor n/2 \rfloor$ copies à B et A conserve les $(n - \lfloor n/2 \rfloor)$ autres copies. La propagation est faite comme sur un arbre binaire.

L'avantage est évident, c'est un algorithme avec un contrôle extrême fort. Il y a L copies au maximum existantes dans le réseau. De plus, une inondation est réalisée au début pour obtenir un 'Fast Start' (par apport 'Slow Start' dans PROPHET...), de cette façon les messages peuvent être propagés plus rapidement. Lorsque le nœud original est loin de la destination, le message va être difficile à envoyer à la destination (le passage à la phase Wait est très rapide). Toutes les copies sont prises par les nœuds autour de nœud original. Quand ces nœuds rencontrent un nœud qui a une prédictibilité plus intéressante, il ne peut pas lui envoyer le message s'ils sont dans une phase Wait.

2.1.4. Gérer la file d'attente

La façon la plus simple pour gérer une file d'attente est la discipline FIFO.

L'article [DFL01] compare plusieurs solutions pour gérer cette file d'attente.

1. DLR : est identique à FIFO.
2. DOA : supprime en priorité le message qui est le plus vieux dans un nœud. C'est-à-dire le message qui existe depuis le plus longtemps dans le réseau parmi tous les messages possédés par ce nœud. Ce message a en effet transité dans le réseau pendant longtemps et il y a une bonne probabilité pour que ce message soit déjà arrivé à la destination.
3. DLE : supprime le message qui a la prédictibilité la plus petite. Parce qu'il y a peu de chance de transmettre de ce message à la destination.

Dans l'article [BGJ06], les auteurs proposent d'utiliser des acquittements pour accélérer la suppression des messages qui sont arrivés à destination. En pratique cette stratégie est un peu difficile à mettre en œuvre, les ACKs ayant aussi un délai important. Cependant, la propagation de ces messages n'ajoute pas beaucoup de charge au réseau, environ 1%. Donc, l'idée semble bonne.

3. Les algorithmes de routage de type DTN pour les réseaux de capteurs sans fil

L'objectif du routage dans les protocoles traditionnels (ceux utilisés pour l'Internet) est de sélectionner le chemin qui minimise certaines métriques simples (ex : le nombre de sauts). Pour les DTNs, l'objectif n'est pas si évident.

Un objectif trivial est de maximiser la probabilité de délivrance des messages, car ces derniers peuvent, bien entendu, être perdus à cause de la création de boucles de routage, ou la perte forcée de données à cause de l'épuisement des buffers.

Vu les caractéristiques intermittentes des réseaux DTNs, les protocoles de routage s'affrontent alors aux challenges suivants :

- La non-existence d'un chemin de bout en bout => les protocoles traditionnels ne fonctionnent plus.
- Les routes sont dépendantes du temps.
- De multiples chemins pourraient être nécessaires.
- Le temps de stockage dans les buffers qui doit être long.

La réponse à ces exigences pourrait engendrer plusieurs problèmes, et une bonne approche serait celle qui tenterait de les minimiser.

Dans la littérature, diverses stratégies de routage sont proposées, chacune essaye de tenir compte de : la capacité des contacts, la capacité des buffers, l'énergie, la capacité de traitement et l'ordonnancement des contacts, et chacune d'entre elles présente aussi bien des avantages que des inconvénients. [CHB012]

4. Conclusion

Ce chapitre est constitué par une présentation des réseaux DTN, ainsi que la description du protocole Epidemic à adapter pour les RCSFs et un état de l'art sur les algorithmes de routage de type DTN pour ce type de réseaux. L'objectif était de comprendre pourquoi le protocole TCP ne suffisait pas à tous les types de communications et quels étaient les autres protocoles de communication.

Chapitre III

Conception et implémentation

Chapitre III : Conception et implémentation

I. Conception de l’algorithme adapté «algorithme épidémique»

INTRODUCTION

Les réseaux DTN ne sont "rien d'autre" que des réseaux autorisant le stockage à plus ou moins long terme (quelques minutes, heures voire jours) de messages appelés bundles dans les nœuds à l'intérieur du réseau. En absence d'une route complète entre deux nœuds, on suppose qu'on peut toujours faire transiter un message dans le réseau de proche en proche entre les nœuds accessibles dans l'espoir que l'un d'entre eux sera à un moment donné à portée du destinataire. Ainsi, le transport de l'information repose sur:

- le principe de "Store-and-forward", ce qui fait que leurs capacités de stockage sont aussi un composant à prendre en compte par le protocole de routage déployé. Notre objectif est l’adaptation et la simulation d’un algorithme capable de contrôler la capacité du buffer et de transmettre des messages avec une probabilité importante en se basant sur des informations provenant des nœuds intermédiaires.

I.1. Formulation du problème de routage dans les DTNs

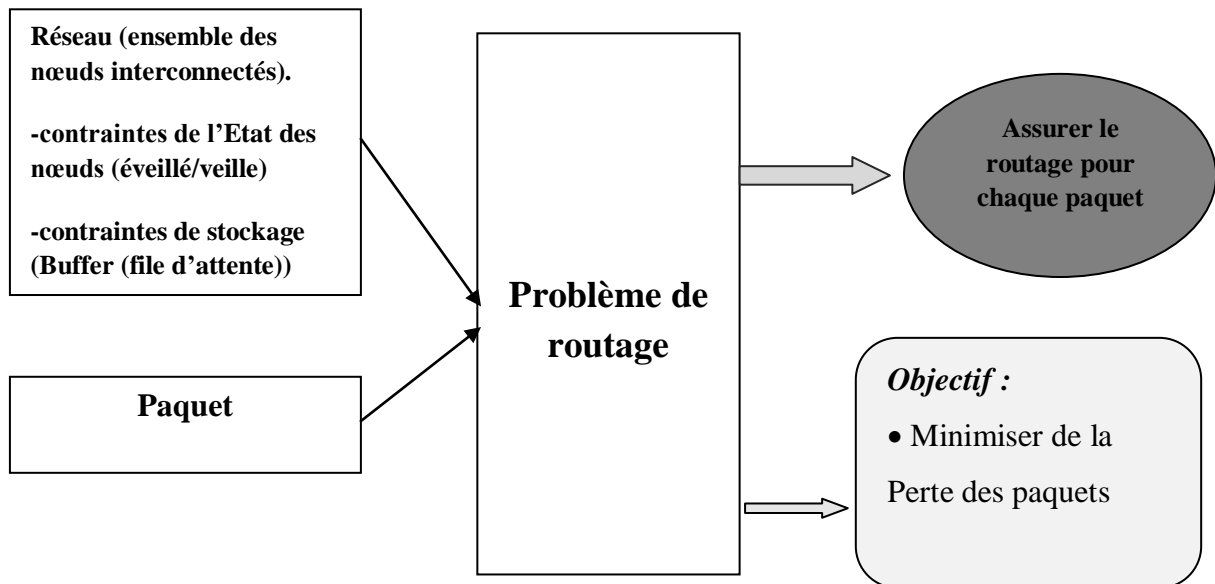


Figure 13: Schématisation du problème du routage dans les réseaux DTN

La figure au dessus représente la schématisation de notre problème de routage sous forme un système dont les entrées sont : les paramètres utilisés par le réseau, des contraintes de changement d'état d'un nœud du mode veille vers le mode éveillé et vis versa, des

contraintes à appliquer sur la file d'attente (buffer) et les différents types de paquets et la sortie de ce système est représentée par la minimisation de la perte des paquets ce type de routage.

I.2. Modélisation des réseaux DTNs

Un réseau tolérant aux délais est modélisé par un multi-graphe où une paire de nœuds peut être connectée. Ceci permet de choisir entre plusieurs types distincts de connexions physiques, afin d'effectuer la transmission de données. En outre, les capacités des liens dépendent du temps. Sa capacité atteint zéro aux moments où il n'est pas disponible.

Un réseau DTN considéré comme :

- Un ensemble de nœuds
- Interconnexion des nœuds régie par des lois probabilistes.

Afin d'implémenter des protocoles de routage pour les réseaux DTN, Warthman a proposé [FW03] un modèle de base qui se repose sur les concepts suivants:

I.2.2 Stockage des messages

Les demandes de communications sont représentées par des messages ou bundles. Un message est modélisé par quatre informations essentielles : la source du message, sa destination, l'instant où le message a été injecté dans le système ainsi que la taille de celui-ci sachant qu'elle est arbitraire.

Les nœuds dans les DTNs ont une capacité de stockage finie à long terme pour l'exploitation et l'attente de consommation par l'application au nœud destination.

I.3. L'algorithme de routage adapté

Les algorithmes épidémiques sont originellement proposés pour synchroniser les bases de données répliquées. Puis ces algorithmes ont été appliqués [2] pour l'expédition de données dans un DTN. Par analogie, la liste des messages en attente de délivrance est la base de données qui a besoin de synchronisation. Dans le routage épidémique, tous les nœuds vont éventuellement recevoir tous les messages, car l'algorithme fournit un nombre aléatoire d'échanges de données suffisantes. Son principe de base est le suivant:

- Lorsqu'un message est envoyé, il est placé dans le buffer, il est ensuite étiqueté par un ID unique.

- Quand deux nœuds se connectent, ils s'échangent leurs listes de messages avec les IDs de ces derniers. Cette liste est appelée «vecteur d'état».En utilisant ce vecteur d'état, les nœuds s'échangent les messages qu'ils ne possèdent pas.
- Et à la fin de cette opération, tous les nœuds auront les mêmes messages dans leurs buffers [EPC05].
- En d'autres termes, le routage épidémique essaie d'envoyer chaque message sur tous les chemins du réseau. Ce qui fourni une grande quantité de redondances car tous les nœuds reçoivent tous les messages, ce qui rend donc cette stratégie extrêmement robuste. En outre, il essaie tous les chemins, il livre chaque message dans un temps minimum s'il y'a des ressources suffisantes.
- Le routage épidémique est relativement simple car il ne nécessite aucune connaissance sur le réseau.
- Pour cette raison, il a été proposé pour être utilisé comme un recule si aucune autre meilleure méthode n'est disponible. L'inconvénient est qu'une énorme quantité de ressources est consommée et ceci est dû au grand nombre de copies se trouvant dans le réseau, ce qui nécessite non seulement une grande quantité d'espace buffer a mis aussi une grande bande passante et aussi suffisamment d'énergie.
- Dans cette section nous allons présenter la démarche de conception UML de l'algorithme proposé précédemment.

I.4. La Conception UML de l'algorithme adapté

Après avoir exprimé et analysé le problème en terme < entées ,sortie, objectif>, nous présentons dans cette partie l'étude conceptuelle de notre algorithme qui consiste à mettre en place un modèle de conception schématisé par le digramme de classe d'UML, qui servira de fondation pour le développement et la réalisation de l'application.

I.4.1. Identification des primitifs de l'algorithme

Le tableau ci-dessous permet d'énumérer les primitifs de l'algorithme et de les décrire :

Opération	Description
Choix de mode de Déploiement des nœuds	Permet de déployer selon un modèle aléatoire ou uniforme l'ensemble de nœud
Déploiement des nœuds	déployer les nœuds de réseau dans un plan à dimensions (X, Y)
Configuration des nœuds	Les nœuds sont configurés initialement.

Création de la liste des voisins	Création de la liste des voisins de chaque nœud on se basant sur la distance euclidienne et la portée de ce nœud
Sélectionner l'émetteur et le récepteur	L'administrateur doit déterminer le nœud source et le nœud de destination.
Calcul des prédictibilités	Calcul de prédictibilité de chaque nœud voisin (voir la section 2.2.2 chapitre II)
Modification de TTL et l'énergie d'un nœud	Modification de TTL d'un paquet et l'énergie des nœuds.
Le choix d'un nœud voisin	Chaque nœud sélectionne un nœud ayant la plus grande priorité dans son voisinage et insérer leur ID dans son vecteur d'état.
Gestion du buffer	plus le nombre de messages délivrés augmente l'espace buffer est consommé nécessite des algorithmes de gestion du buffer (algorithme FIFO).
Max prédictibilité	Le message est envoyé au nœud voisin qui possède une grande prédictibilité.
Envoie du broadcast	Une approche simple pour lui transmettre les données à tous les voisins qui sont découverts.
Prédire	Pour déterminer le récepteur prochain le nœud émetteur utilise les prédictions exactes sur les nœuds voisins dans le réseau.
Envoie d'ACK	L'envoi du paquet ACK indique au émetteur la réception du données par le récepteur.

Tableau 1: Liste des Opérations de l'algorithme

Dans la simulation et la configuration des paramètres d'algorithme l'outil fait alors le choix (par défaut) et affiche le résultat. Tandis que la simulation et personnalisée donne plus de liberté à l'administrateur pour des paramètres de simulation comme (la taille de paquet, émetteur / récepteur, durée d'écoute).

Les principales fonctionnalités sont : (1) le Choix de mode déploiement (2) : le Déploiement du nœud (3) La Configuration des paramètres des nœuds dans le réseau et enfin (4) Le lancement de la simulation et la visualisation des résultats théoriques sous forme des graphes.

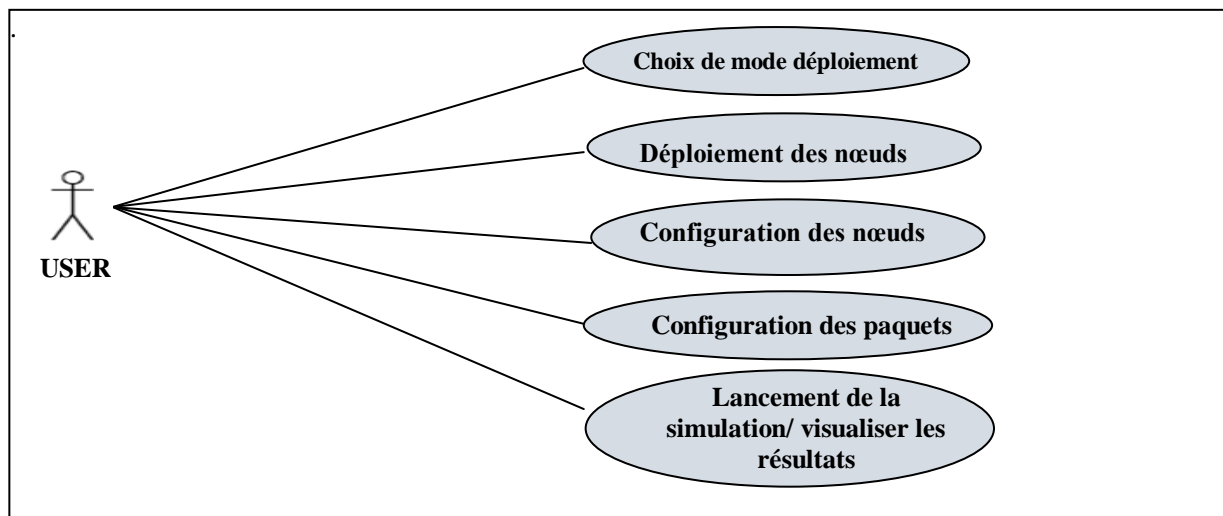


Figure 14: Le Diagramme des cas d'utilisation de notre application

1.4.2. Déroulement de l'algorithme adapté :

Dans cette étape, nous allons représenter le déroulement des opérations de l'algorithme avec un diagramme d'activité (Voir Figure 15). Les diagrammes d'activité permettent de structurer et décrire le scénario et les relations attendue par l'algorithme adapté.

1.4.2.1. Diagramme d'activité de l'algorithme adapté

Afin de représenter l'aspect dynamique dans la conception de l'algorithme adapté. Nous avons élaboré un diagramme d'activité qui décrit l'interaction entre les objets du système. Le scénario de perte de paquet est provoqué par la durée de vie TTL. Le processus de l'envoi de paquet se répète d'une manière itérative tant que le nœud récepteur est et en veille.

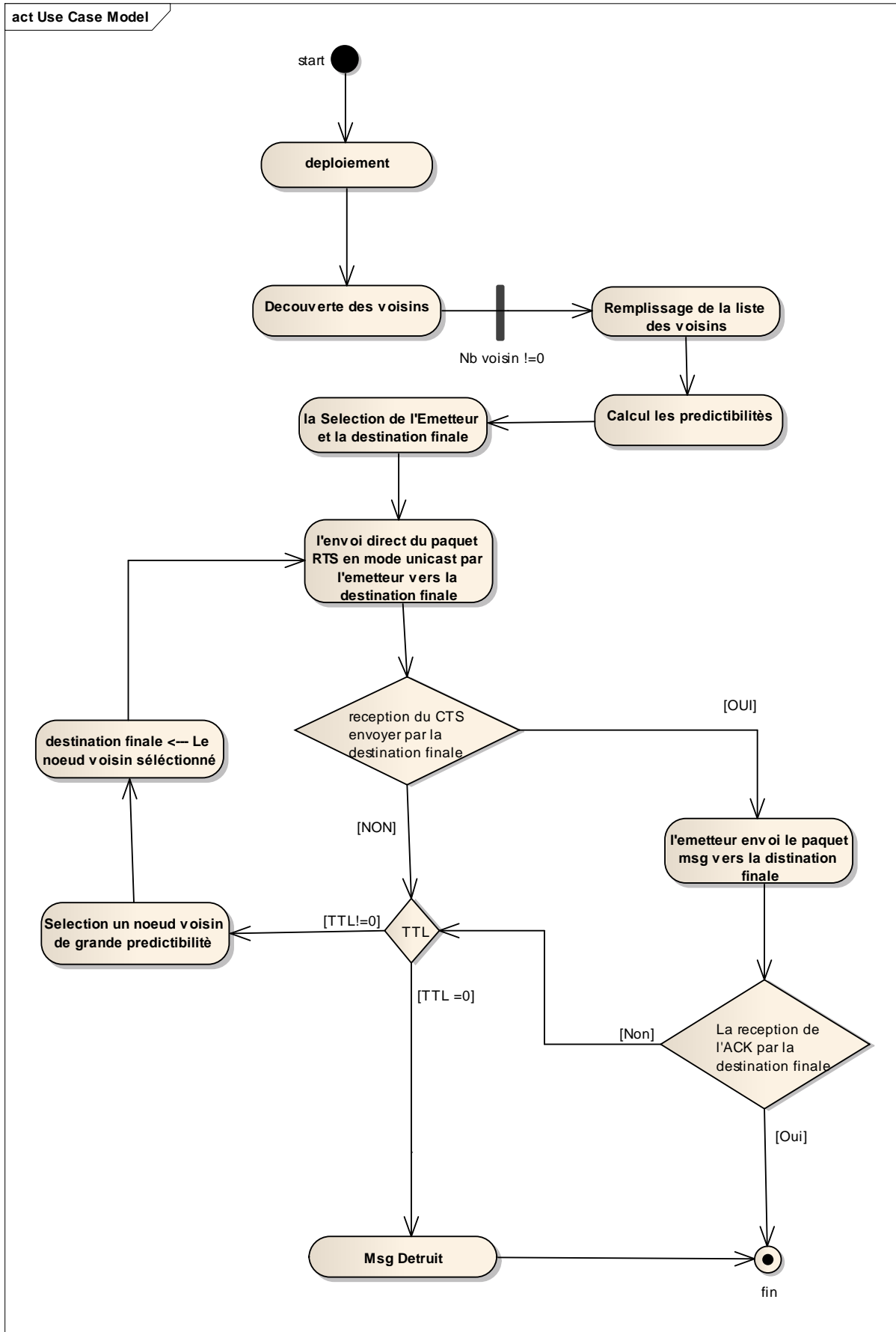


Figure 15: Diagramme d'activité de l'algorithme adapté

Les changements d'état des deux objets nœud et paquet se représenter par les deux diagrammes d'état de transition suivantes :

I.4.2.3 Diagramme de Classes

Le diagramme de classes exprime la structure statique interne du système en termes de classes et de relations entre ces dernières, ainsi que leurs attributs et méthodes. Autrement dit, il permet de fournir une représentation abstraite des objets du système et leurs liens.

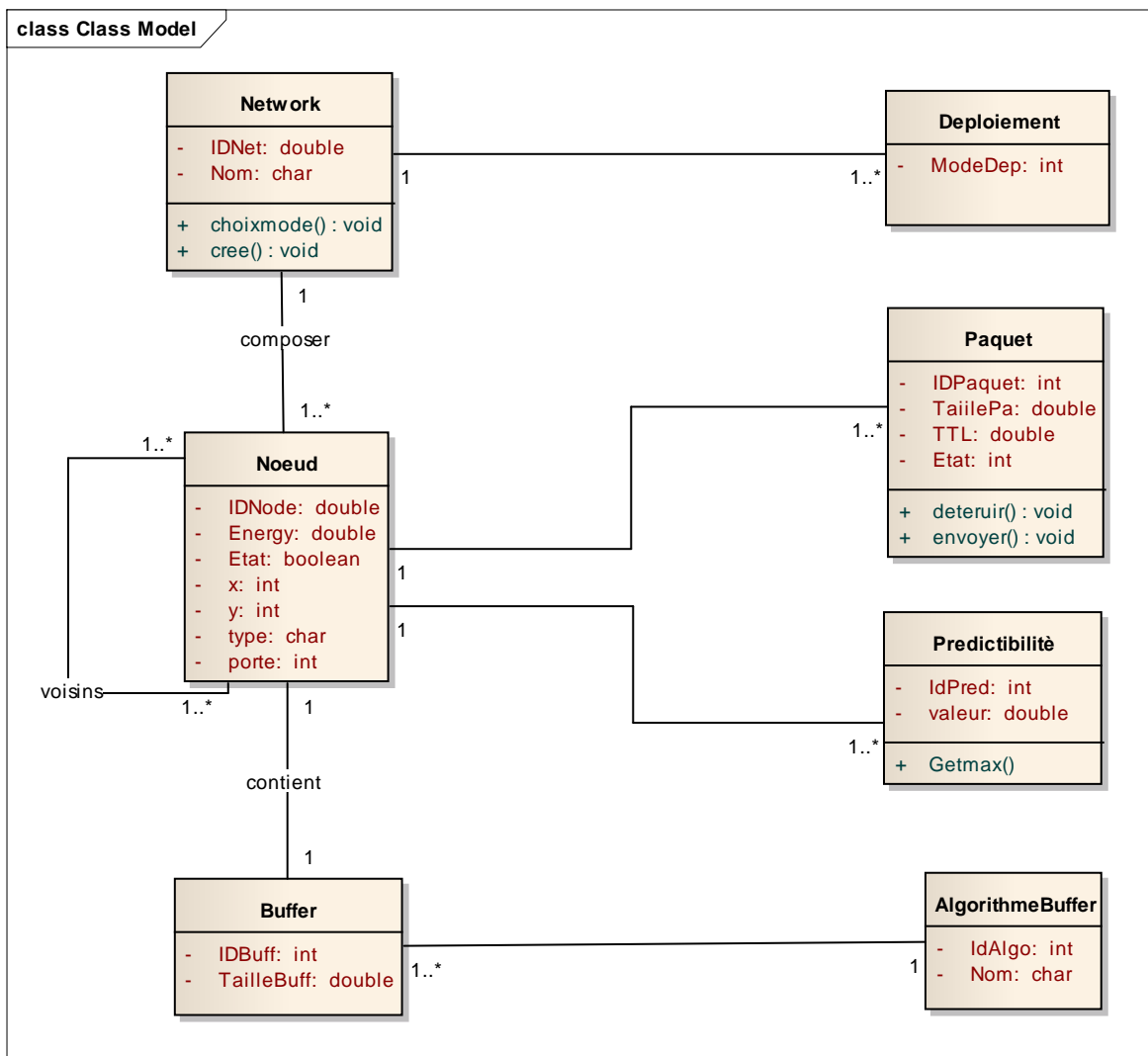


Figure 16: Diagramme de classe de notre application

I.4.2.3.1 Description des classes

Le tableau ci-dessous décrit les différentes classes composant le volet conceptuel de notre outil.

Nom de la classe	Description
Réseau	Le réseau utilisé se compose de plusieurs nœuds de capteur.
Déploiement	Nous avons de types déploiements à utilisés : déterministe ou aléatoire
Nœud	Chaque nœud est caractérisé par un ensemble de paramètres.
Prédictibilité	La prédictibilité est caractérisée par une méthode de calcul.
Buffer	Chaque Buffer (file d'attente) d'un nœud possède une capacité de stockage
Algorithme Buffer	L'Algorithme buffer est dédiée à la gestion des buffers utilisés par chaque nœud.
Paquet	Chaque paquet possède des caractéristiques spécifiques.

Tableau 2: Description des classes utilisées dans le diagramme de classe

Pour illustrer une représentation proche de notre code de programmation, nous avons décrit l'algorithme sous forme d'un diagramme de séquence qui montre les scénarios d'interaction entre les classes.

I.4.2.4. Diagramme de séquence

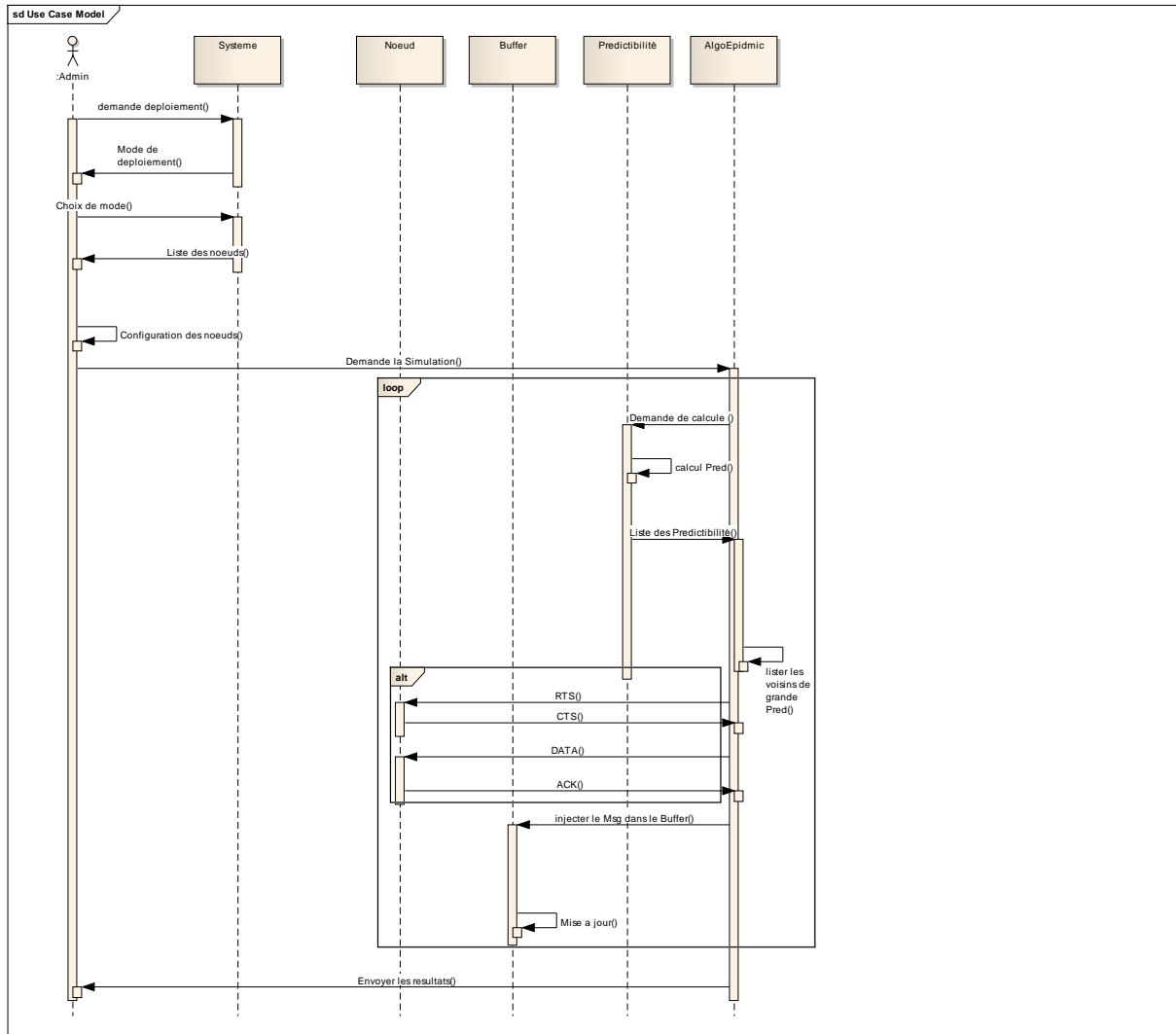


Figure 17: Diagramme de séquence

II. Implémentation et Expérimentation

II.1. Implémentation de Notre application

Après avoir exprimé les différents besoins fonctionnels et présenté la succession des étapes de conception et de modélisation de l'algorithme que nous avons proposé, nous nous sommes basé sur la conception UML afin de procéder à la réalisation de cette application qui présente un support de simulation de l'algorithme adapté.

Nous abordons dans ce chapitre la mise en œuvre de notre application qui constitue le volet pratique de notre travail. Nous commençons par lister et justifier nos choix vis-à-vis des outils de développement utilisés, puis nous allons procéder à la description du fonctionnement de l'outil.

II.2. Description des choix techniques d'implémentation

Le choix de l'IDE est souvent dicté par des préférences personnelles. Néanmoins, il est conseillé d'utiliser Eclipse ou Netbeans dans notre cas. Ces deux IDEs facilitent le développement en Java SE³ et ME⁴. Afin de procéder au développement de notre application, notre choix a porté sur l'utilisation du langage de programmation JAVA sous l'IDE Netbeans, vu que c'est un langage Orienté Objet de très haut niveau, capable de s'exécuter sur n'importe quelle plateforme grâce à la JVM (Java Virtual Machine) et possédant un ensemble complet d'API (réseau, graphique, sécurité, SGBD...).

II.3. Présentation de l'application

Nous présentons dans cette section les principales fonctionnalités de l'outil application à travers des captures écran illustrant les interfaces de ce dernier. Pour bien décrire ces interfaces, nous considérons un scénario d'exécution et de manipulation de l'outil, à partir de la configuration, en passant par déploiement de réseaux, le lancement de la simulation et la visualisation des résultats sous forme des graphes.

³Standard Edition: plateforme Java destinée plus particulièrement aux applications pour poste de travail.

⁴Micro Edition:framework Java spécialisé dans les applications mobiles.

II.3.1 Interface d'accueil application

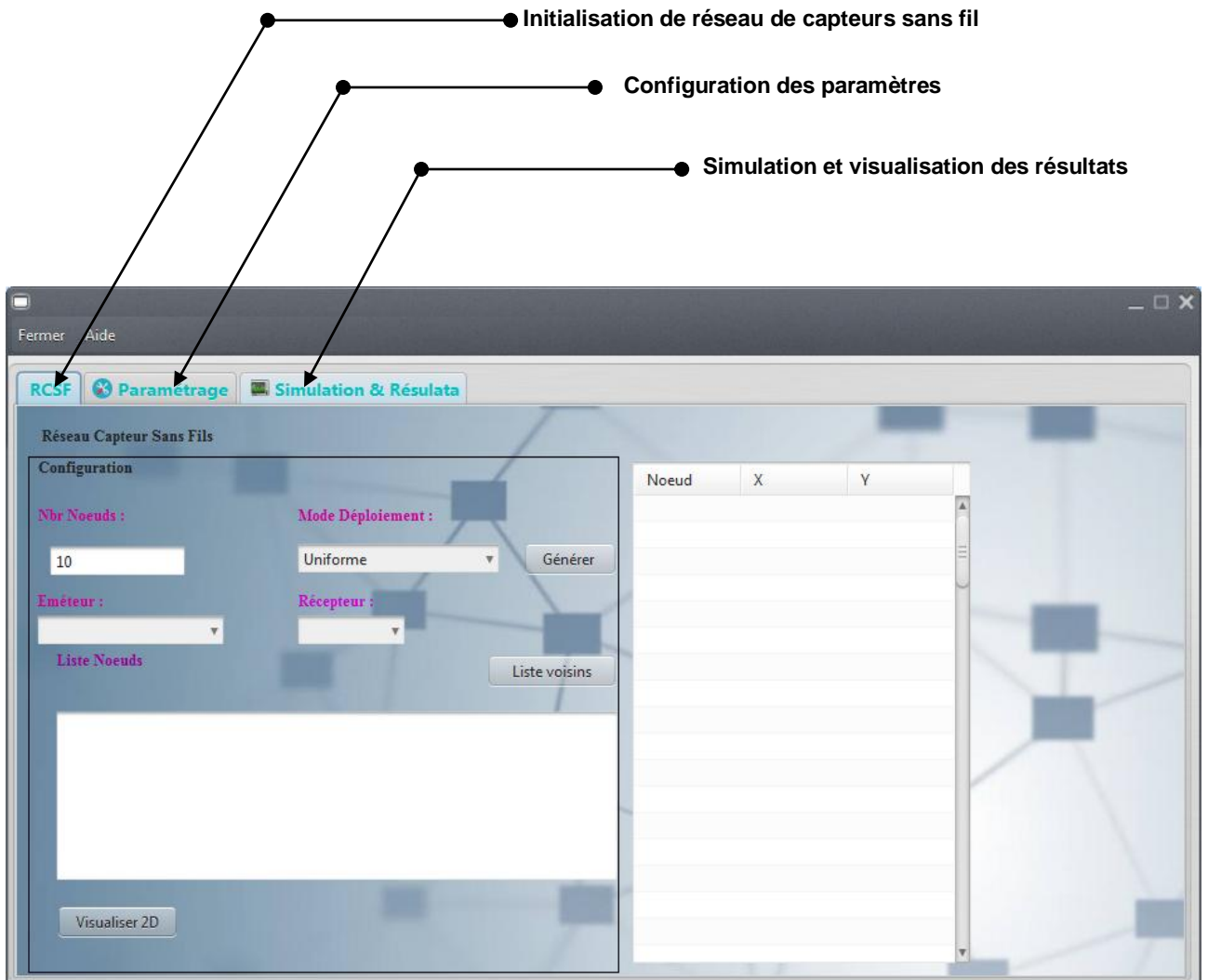


Figure 18: Interface d'accueil application

II.3.2 Interface indiquée par l'onglet RCSF

Cette fenêtre montre le scénario de déploiement des nœuds de capteurs sans fil dans un espace de deux dimensions, l'utilisateur doit spécifier premièrement le nombre de nœuds et leur mode de déploiement.

Une fois on clique sur le bouton liste voisins le système affiche la liste des nœuds voisins avec ses coordonnées (X, Y). La figure suivante illustre l'interface de déploiement du RCSF.

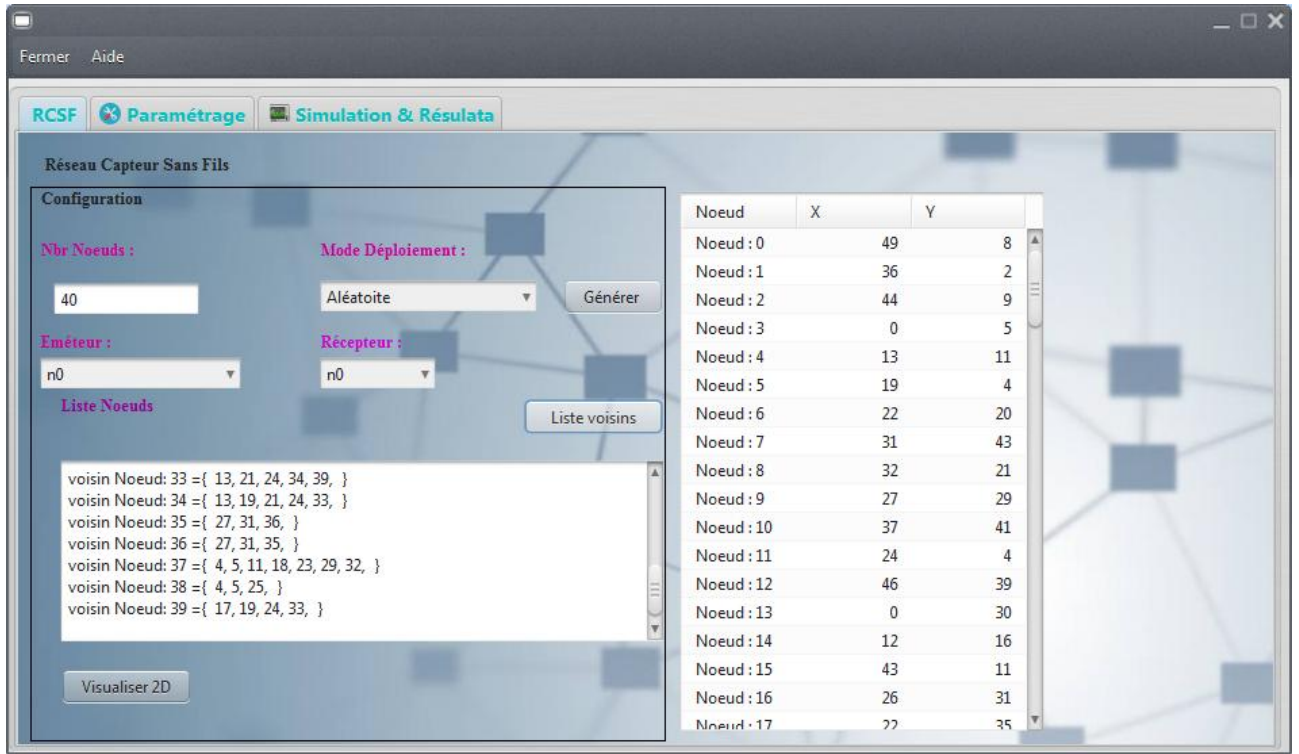


Figure 19: interface RCSF

II.3.3 Interface de l'onglet de paramétrage

Pour lancer la simulation, L'utilisateur doit procéder à introduire les paramètres nécessaires de système (Portée des nœuds, Taux d'écoute, énergie initiale des nœuds, nombre de paquets à créés, la taille de paquet et la capacité de buffer).

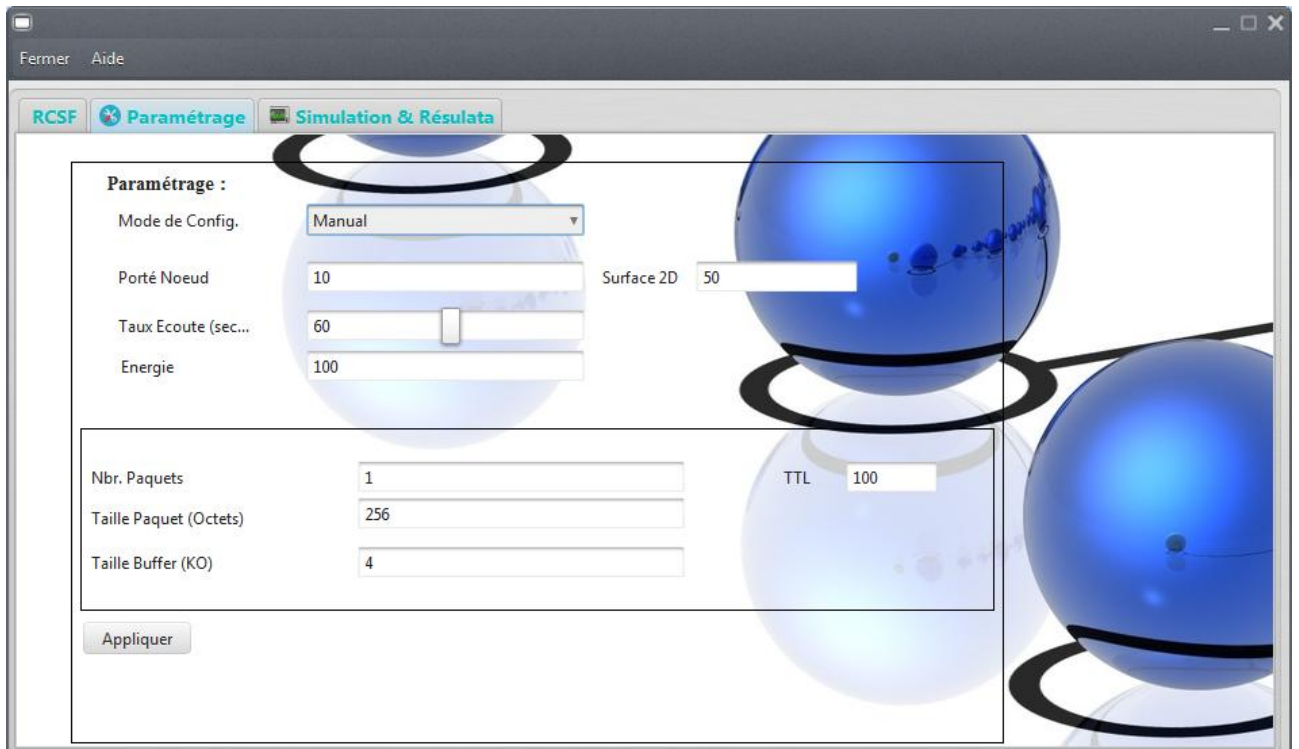


Figure 20: Interface de paramétrage

Adaptation et la simulation d'un algorithme de type DTN pour les réseaux de capteurs sans fil

II.3.4 Interface de l'onglet de simulation & visualisation

Après avoir saisi les paramètres nécessaire et déployer les nœuds du réseau, l'utilisateur lance la simulation et visualise les résultats en terme de nombre de paquets reçu, nombre de paquet détruits et l'état de chaque paquet envoyer.

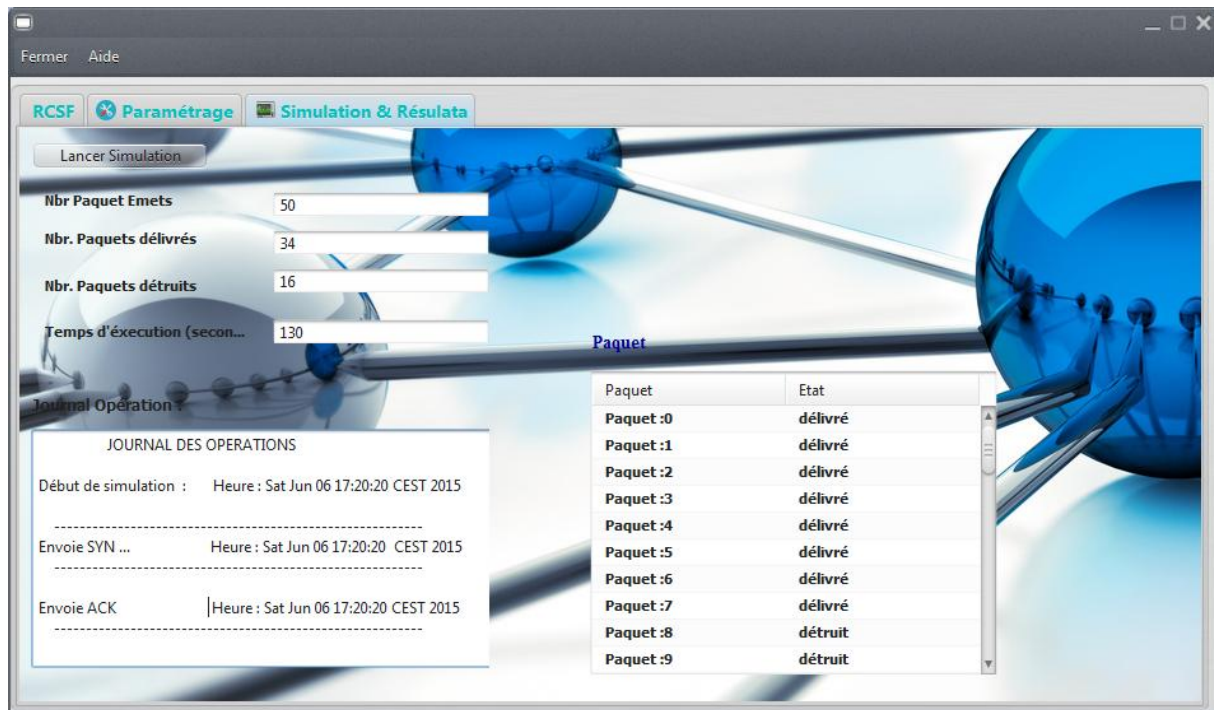


Figure 21: Interface de l'onglet de simulation & visualisation

II.4. Les scénarios d'expériences

Les scénarios des expériences s'articulent autour les points suivants :

- La durée de connexion.
- Nombre de paquets reçus en fonction de la capacité des files d'attente

II.4.1 Durées d'une Connexion

Les simulations suivantes examinent la durée de l'existence et d'absence de connexion directe (temps inter-rencontres) entre tout couple de nœuds. L'histogramme de ces temps est représenté sur la figure 24

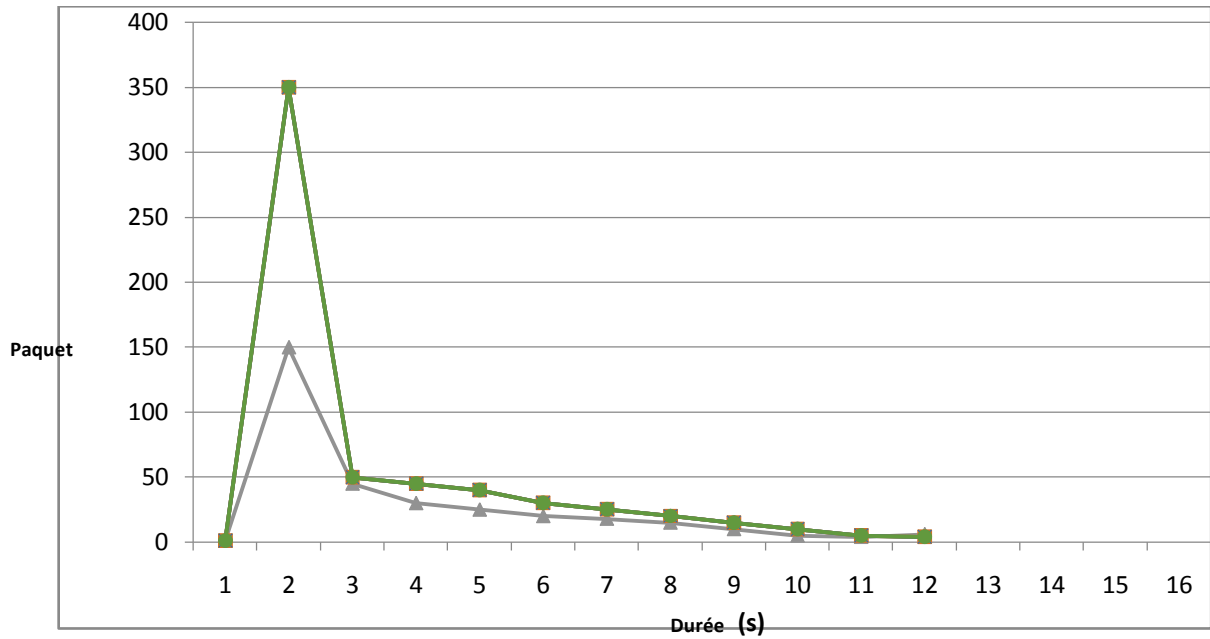


Figure 22: Durées d'une Connexion

Toutes les 4 secondes, chaque nœud génère un paquet vers une destination est aléatoire (en dehors de lui-même). L'utilisation de probabilité (prédictibilité) qui représente un seuil de confiance pour prévoir les moments d'envoi de paquets entre les nœuds. A partir de certain seuil on remarque que le nombre de paquet reçu est diminué ce qui explique que les tampons des nœuds sont tous de même taille et lorsqu'une file est pleine, on supprime le paquet qui a passé le plus de temps dans cette file.

II.4.2 Nombre de paquets reçus en fonction de la taille des files d'attente

Nous étudions l'influence de la taille de la file d'attente sur les performances.

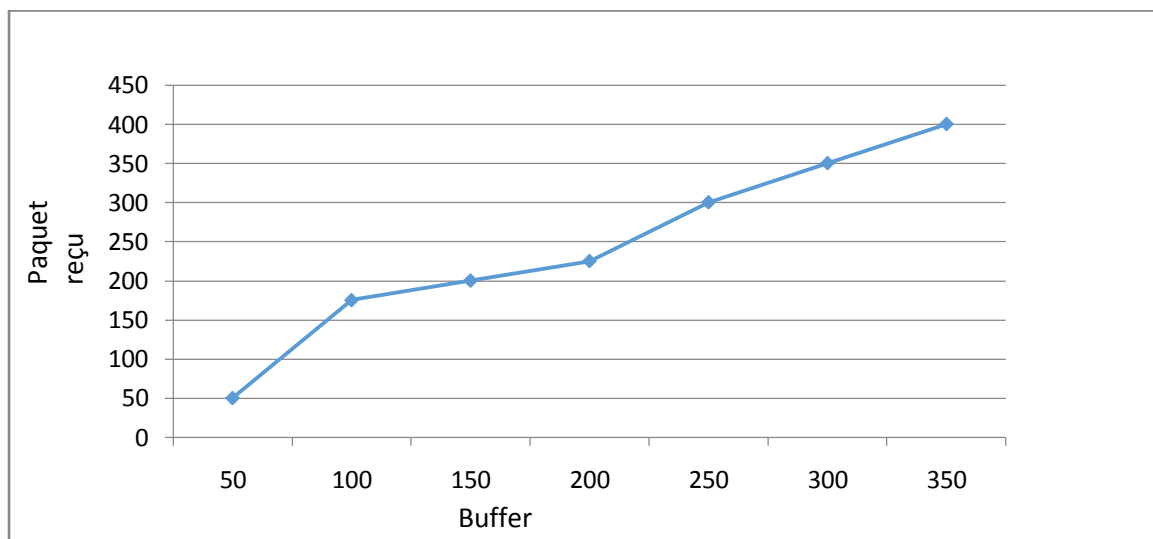


Figure 23: Nombre de paquets reçus en fonction de la taille des files d'attente

On remarque d'après les résultats obtenus illustrées dans la figure 25 par la courbe indique que le nombre de paquet reçu en utilisant Epidemic est avec succès dans le cas où la taille du buffer s'augmente progressivement. A partir de ces résultats on peut déduire que l'algorithme adapté a de bonnes performances et que la contrainte de la taille de paquet influence sur la politique de gestion du buffer (FIFO) alors l'évitement le cas où l'état du buffer plein nous permet de diminué le nombre de paquets détruits.

II.5. Conclusion

Les DTN ce sont des réseaux souffrant d'un a priori d'un grand délai de bout en bout du fait de l'absence de connectivité permanente. Les protocoles de routage traditionnels ne fonctionnent pas dans cette situation.

Nous avons adapté un algorithme qui s'appelle *algorithme épidémique*. Il consiste à diffuser les paquets le plus rapidement possibles au début, puis à transmettre les paquets selon une métrique (prédictibilité) afin de limiter strictement le nombre des copies de chaque paquet circulant dans le réseau et d'optimiser l'utilisation des ressources du réseau.

D'après les simulations réalisées, on peut dire que l'algorithme épidémique à des bonnes performances. Cependant son efficacité peut encore être améliorée.

Conclusion Générale

Avec de très larges perspectives d'applications, DTN appelle à un grand nombre de chercheurs de mener une étude. La ressource de DTN est limitée et connectés par intermittence pour faire DTN difficile. Il est d'une grande importance que nous pouvons réduire le délai de transmission et de communication et d'assurer le taux de livraison de messages.

Nous avons proposé l'adaptation de l'algorithme qui s'appelle épidémique qui est basé sur le calcul de la prédictibilité des nœuds voisins afin de sélectionner le récepteur possédant la plus grande prédictibilité et de limiter la perte de paquet grâce à l'utilisation de la file d'attente (Buffer).

D'après les simulations réalisées, tout laisse à penser que l'algorithme adapté a de bonnes performances, notamment lorsque la connexion existante entre les différents nœuds et aussi quand la taille de la file d'attente est plus grande.

Perspectives et améliorations possibles :

Certaines améliorations peuvent être apportées à notre protocole. Citons, en particulier :

- ✓ La simulation l'algorithme adapté Epidémique par un autre simulateur tel que le NS2, Castalia sous OMNET++
- ✓ La comparaison de l'algorithme adapté avec d'autres algorithmes de même type que l'Epidémique tel que le protocole spary and wait et le protocole Prophet.

Bibliographie

[ACH05] Mounir ACHIR, « Technologies basse consommation pour les réseaux ad-hoc », Thèse de Doctorat de l'Université INPG, Institut National Polytechnique de Grenoble, JUI 2005.

[AMZ11] Amel AMZIANE, Meriem TOUMI « Plateforme d'évaluation de la tolérance aux pannes des protocoles de routage dans les réseaux de capteurs sans fil », Mémoire d'ingénieur, Ecole nationale Supérieure d'Informatique ESI, 2011.

[AKY01] I.F. AKYILDIZ, Weilian. SU, Yogesh Sankarasubramaniam and Erdal CAYIRCI, « Wireless sensor networks: a survey, Computer Networks 38 », pp. 393–422, 2001.

[AKY02] I.F. AKYILDIZ, Weilian SU, Yogesh Sankarasubramaniam, and Erdal CAYIRCI, « A servey on sensor networks », IEEE Communications Magazine, pp.102-114, August 2002.

[AV00] Amin Vahdat and David Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.

[BBL05] B. Burns, O. Brock, and B. N. Levine, MV routing and capacity building in disruption tolerant networks, in Proc. Infocom, 2005.

[BEC09] Walid BECHKIT, « Un nouveau protocole de routage avec conservation d'énergie dans les réseaux de capteurs sans fil », Mémoire d'ingénieur, Ecole nationale Supérieure d'Informatique ESI, Juin 2009.

[BEC09] Walid BECHKIT, « Un nouveau protocole de routage avec conservation d'énergie dans les réseaux de capteurs sans fil », Mémoire d'ingénieur, Ecole nationale Supérieure d'Informatique ESI, Juin 2009.

[BGJ06] J. Burgess, B. Gallagher, D. Jensen, B. N. Levine, MaxProp: Routing for Vehicle-based Disruption-Tolerant Networking, in infocom 2006

[CAS11] CASAS CHICA Juan Pablo, NGUYEN Viet Minh, VU Ninh Thanh Hien, « Application de vidéo surveillance avec un réseau de capteurs multimédia sans fils », rapport de projet Master1 technologie internet, Université de Pau et des pays de l'Adour, 2011.

[CHA10] Gérard CHALHOUB, « Les réseaux de capteurs sans fil », Clermont Université IUT Réseaux et Télécommunications de Clermont-Ferrand, France, 2010.

[CHB012] Chakib BAOUICHE «Une solution tolérante aux délais pour des applications de localisation et de traçabilité a posteriori en milieux confinés» 2012

[DFL01] J. Davis, A. Fagg, and B. Levine. Wearable computers and packet transport mechanisms in highly partitioned ad hoc networks. In Proc. Intl. Symposium on Wearable Computers, October 2001.

[DUR09] Cristian DURAN-FAUNDEZ, « Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'énergie », Thèse de Doctorat de l'Université Henri Poincaré, Nancy 1 en Sciences, spécialité Automatique, Traitement du Signal et Génie Informatique, 2009.

- [EPC05] E. P. C. Jones, L. Li, P. A. S. Ward “Practical Routing in Delay-Tolerant Networks” SIGCOMM’05 Workshops, August 22-26, 2005.
- [FW03] F. Warthman “ Delay Tolerant Network (DTNs)” A tutorial. Forest warthman Warthman Associates (www.warthman.com). Version 1.1. 2003
- [ILY04] M. ILYAS and I. MAHGOUB, «Handbook of sensor networks: compact wireless and wired sensing systems», CRC Press, 2004.
- [GEL06] J. P. GELAS / ATER “Réseaux interplanétaires (IPN) et réseaux tolerant aux délais (DTN)”, Laboratoire de l’Informatique du Parallélisme, 2006
- [GYN05] J. Ghosh, S. Yoon, H. Ngo, and C. Qiao, Sociological Orbits for Efficient Routing in Intermittently Connected Mobile Ad Hoc Networks. Tech Report, Department of Computer Science and Engineering, University at Buffalo, The State University of New York, 2005.
- [HAC03] A. HAC, « Wireless Sensor Network Designs», Edition: John Wiley & Sons Ltd, 2003.
- [OUL10] Mounira OULARBI Soraya KASSAB, « HAORP (Hierarchical Adaptive Optimized Routing Protocol), Elaboration d’un protocole de routage efficace en énergie pour réseaux de capteurs sans fil », Mémoire d’ingénieur, Ecole nationale Supérieure d’Informatique, septembre 2010.
- [HAM07b] A. HAMZI, « Plateforme pour l’aide à la conception et à la simulation des réseaux sans fil », mémoire pour obtenir le grade de magister en informatique, Institut National de formation en Informatique, 2007.
- [HCB00] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, " Energy-Efficient communication Protocol for Wireless Micro sensor Networks", In proc of the Hawaii International Conference on Systems Science, vol. 8, pp. 8020, January 2000.
- [POM 04] C. POMALAZA-REZ, « Wireless Ad Hoc & Sensor Networks», University of Oulu, Finland, 2004.
- [WAR03] F. Warthman “ Delay Tolerant Network (DTNs)” A tutorial. Forest Warthman. Warthman Associates (www.warthman.com). Version 1.1. 2003
- [KRI08] A. KRIFA “An optimal joint scheduling and drop policy for delay tolerant networks”. PLANETE-INRIA. April 2008.
- [LDS03] A. Lindgren, A. Doria, and O. Schel’én. Poster: Probabilistic routing in intermittently connected networks. In Proceedings of The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), Annapolis, MD, June 2003.
- [SPR04]: T. Spyropoulos, K. Psounis, and C. S. Raghavendra, Single-copy routing in intermittently connected mobile networks, USC, Tech. Rep. CENG-2004-11, 2004.
- [SPR04]: T. Spyropoulos, K. Psounis, and C. S. Raghavendra, Multiple-copy routing in intermittently connected mobile networks, USC, Tech. Rep. CENG-2004-12, 2004.
- [SGB06] Solène ALOS Georges-Rémy FOUAD Patrick PEROUSE DE MONTCLOS PI20 : Mars Communication 2006

Adaptation et la simulation d’un algorithme de type DTN pour les réseaux de capteurs sans fil

[LFC05] J. Leguay, T. Friedman, and V. Conan, DTN routing in a mobility pattern space, in Proc. WDTN, 2005.

[LFC06] J. Leguay, T. Friedman, and V. Conan, Evaluation Mobility Pattern Space Routing for DTNs, in infocom 2006.

[MHM05] M. Musolesi, S. Hailes, and C. Mascolo, Adaptive routing for intermittently connected mobile ad hoc networks, in Proc. WOWMOM, 2005.

[XB10] **Travaux de XU Jian-bo et al.** A new data transmission protocol in Delay Tolerant Mobile Wireless Sensor Networks 2010

[ZAZ04] E. Zegura, M. Ammar, W. Zhao, A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks, in MobiHoc 2004.

[ZWW13]: Sai Zou, Wei Wang and Wenyong Wang , A routing algorithm on delay-tolerant of wireless sensor network based on the node selfishness , EURASIP Journal on Wireless Communications and Networking 2013,