DOCTORAL DISSERTATION OF THE UNIVERSITY OF BURGUNDY FRANCHE-COMTE

PREPARED AT THE UNIVERSITY OF BURGUNDY

Doctoral School n° 37
Sciences Pour l'Ingénieur et Microtechniques

PhD in Computer Science

By
Mr. Simon Tabi OBENOFUNDE

TOPOLOGY MANAGEMENT IN WIRELESS SENSOR NETWORKS

A Dissertation presented and defended at Dijon on the 26/10/2020

Composition of the Jury:

Mr. KHEDDOUCI Hamamache	Professor at the University of Lyon	President
Mr. MOSBAH Mohamed	Professor at the Bordeaux INP	Reviewer
Mrs. ABY Thérèse	Assoc. Professor, ESIEE Amiens	Examiner
Mrs. ROXIN Ana	Assoc. Professor, University of Burgundy	Examiner
Mr. TOGNI Olivier	Professor at the University of Burgundy	Thesis Supervisor
Mr. ABDOU Wahabou	Assoc. Professor, University of Burgundy	Co-Supervisor

(Sciences Pour l'Ingénieur et Microtechniques)
(Engineering Sciences and Microtechnologies)

Résumé

Titre : Gestion de la topologie dans les réseaux de capteurs sans fil

Mots-clés : efficacité énergétique, cycle d’activation, réseaux dorsaux virtuels disjoints

Résumé : La mise en réseau de capteurs sans fil s'intègre dans presque tous les domaines des activités humaines. Les moteurs de cette technologie comprennent ses domaines d'application et les améliorations des techniques de fabrication microélectroniques. Le réseau est constitué de plusieurs nœuds de capteurs de petite taille déployés dans la zone à détecter. Les nœuds ont des capacités de traitement, de communication et de détection qui leur permettent d'exécuter leur fonction de manière collaborative. Ils détectent les événements et transmettent les informations à un puits directement ou via des nœuds intermédiaires servant de relais.

Des progrès considérables ont été réalisés sur cette technologie au cours des dernières années, cependant la gestion de l’énergie n’a pas connu la même évolution. Ceci est principalement dû au fait que la batterie est la principale source d'énergie. De plus, l’environnement du réseau peut empêcher les batteries d'être rechargées ou changées après le déploiement.

Une solution classique à ce problème d'efficacité énergétique réside dans la gestion des cycles d’activation. Il s'agit d’alterner, de façon périodique ou non, les états actif et inactif des nœuds. Cela introduit des problèmes de performances réseaux en termes de disponibilité, de latence et de taux d’acheminement des paquets, car les nœuds inactifs ne participent pas aux communications. Il est donc important de trouver des solutions permettant d’utiliser les cycles d’activation tout en garantissant la disponibilité et en réduisant la latence et le taux de perte de paquets.

Dans cette thèse, nous utilisons le cycle d’activation en combinaison avec la gestion de la topologie pour prolonger la durée de vie du réseau. Nous proposons cinq algorithmes pour construire différentes topologies que nous divisons en deux classes. La première classe organise les nœuds en ensembles de manière répétitive et entrelacée. C'est-à-dire que les nœuds appartenant à différents ensembles sont intercalés de manière à assurer la continuité des communications. La seconde classe d'algorithmes organise les nœuds en ensembles successifs en couronne. Nous avons montré expérimentalement la construction des différents ensembles.

En utilisant la construction successive d’ensembles, nous proposons deux algorithmes qui construisent des réseaux dorsaux (*backbones*) virtuels disjoints pouvant être activés alternativement. Une évaluation des algorithmes fait ressortir leur efficacité, avec notamment un facteur d’approximation faible (de l’ordre de 3.5) en comparaison avec ceux des travaux de la littérature.

Nous proposons ensuite un protocole basé sur les mécanismes de sommeil et relais sur ces topologies. Les périodes d’activité/inactivité sont définies par ensemble. Les résultats expérimentaux montrent que ce protocole permet une économie d’énergie sans dégrader les critères de performance tels que la latence et le taux d’acheminement des paquets.

(Sciences Pour l'Ingénieur et Microtechniques)
(Engineering Sciences and Microtechnologies)

Abstract

Title: Topology Management in Wireless Sensor Networks

Keywords: Energy efficiency, duty-cycle, disjoint virtual backbone networks.

Abstract: Wireless sensor networking is ingratiating itself into almost every area of human endeavors. Its drivers include its usages, improvements in microelectronics and manufacturing techniques. The network is made up of multiple tiny sensor nodes deployed in the area to be sensed, with nodes having processing, communicating, and sensing capabilities that enable them to perform their function collaboratively. Nodes sense events and transmit their data to the sink directly or through intermediate nodes acting as relay.

Despite all the tremendous advances that have been made on this technology over the past few years, energy has not kept pace. This is based mostly on the fact that battery is its main source of energy. Furthermore, some applications of the network may preclude batteries from either being recharged or changed after deployment.

A renowned solution to energy efficiency is duty cycling. This is the periodic or aperiodic placing of a node in an active and an inactive state. This introduces network performance issues of availability, latency, and packet delivery ratio, all linked to the fact that once a node is inactive or off, it is unavailable to communicate. It is therefore important to look for means of still applying duty cycling yet not losing out in availability, latency, and packet delivery ratio.

In this dissertation we employ duty cycle on topology management to extend the network lifetime. We propose five algorithms to build various topologies that we divide into two classes. The first class enables nodes to arrange themselves into repetitive and interleaving sets. That is, nodes in the same set repeat themselves on the ground such that a set spans the entire area to be sensed. The second class of algorithms arranges nodes in continuous successive sets with members of a set covering a transmission range. We demonstrate the set formation experimentally.

Building on the continuous set formation we propose two algorithms that build disjoint virtual backbone networks, with the disjointedness used for activity scheduling. We then measure the performances of the algorithms notably the approximation ratio and find it quite low (in the order of 3.5) compared to what is obtained in the literature.

Finally, we propose a sleep and relay protocol that works on these topologies. Nodes sleep in sets and the activeness is relayed between sets. We evaluate the performance of this protocol and confirm that it actually leads to increase energy savings while not deteriorating other network performance metrics, like latency and packet delivery ratio.

Résumé étendu

1. Introduction

Les changements dans notre environnement peuvent affecter positivement ou négativement nos vies. Il est donc important de pouvoir surveiller, réagir ou contrôler leurs effets. Cela implique de recueillir et d'analyser des informations relatives à ces changements. Les capteurs sont des dispositifs permettant de collecter des données physiques concernant l’environnement où ils se trouvent et les convertir en forme numérique pour analyse. Les analyses peuvent avoir lieu à un endroit éloigné de l'endroit où la détection s'est produite, tout cela avec peu ou pas d'intervention humaine. Cela signifie que les nœuds doivent posséder des capacités de détection, de calcul et de communication, en plus de l'alimentation. Les nœuds peuvent également avoir des composants supplémentaires dépendants de l'application.

Un réseau de capteurs sans fil (*Wireless Sensor Network*, *WSN*) est une interconnexion de nœuds de capteurs coopératifs communiquant à l’aide de technologies sans fil. Un nœud qui détecte un événement et a donc des informations à envoyer est considéré comme une **source**, tandis que le nœud destinataire qui agit comme une station de base et généralement connecté à un réseau d'infrastructure, est appelé le **puits**. Les domaines d'application des WSN sont nombreux et continuent de croître. Ils comprennent la surveillance de l'environnement, le suivi des ressources, la e-santé, la surveillance des infrastructures critiques, la sécurité urbaine, les réseaux intelligents, etc.

2. Problématique de recherche

Des progrès considérables ont été réalisés dans le domaine des WSN au cours des dernières décennies, cependant la gestion de l’énergie n’a pas connu la même évolution. L'efficacité énergétique, qui concerne à la fois approvisionnement et dépense efficace d'énergie, ne s'est pas améliorée au même rythme que les autres sous-systèmes WSN. Bien que de nombreux travaux aient été effectués récemment sur de nouvelles méthodes d'approvisionnement en énergie, la batterie reste la source la plus viable. Ceci est principalement dû à sa maturité technologique et à sa taille raisonnable. Mais toute batterie finira par se décharger complètement et le nœud deviendra inutile si la batterie ne peut pas être rechargée ou remplacée. L'environnement de déploiement et le nombre et la taille des nœuds rendent difficile, voire dangereux, l'accès aux nœuds après le déploiement pour le changement ou la recharge de la batterie. D'autres techniques pour l'approvisionnement en énergie dans ces réseaux incluent la récupération d'énergie et le transfert d'énergie.

La consommation énergétique est un élément critique dans les WSN. Cela implique une gestion rigoureuse et efficace du réseau. Nous pouvons généralement classer les protocoles ou mécanismes de conservation d'énergie dans les WSN en utilisant les trois approches principales suivantes :

- Approche basée sur les données
- Approche basée sur la mobilité
- Cycle d’activation (*duty cycle*)

Les approches basées sur les données impliquent de réduire la quantité de données circulant sur le réseau. Celles-ci peuvent nécessiter une agrégation à différents points, des schémas de codage, d’atteindre un seuil avant détection, etc. Les approches de mobilité impliquent un mouvement des nœuds qui peut entraîner une réduction de la distance entre une source et un puits, une réduction du nombre de sauts, etc. Le cycle d’activation consiste à alterner l'état d'un nœud entre marche et arrêt de façon cyclique.

Dans ce travail, nous nous concentrons sur le sous-système de communication qui est l'un des plus gros consommateurs d'énergie. Nous nous intéressons aux mécanismes d’activation de ce sous-système en utilisant le duty cycle.

3. Nos contributions

Cette thèse porte sur la gestion de la topologie des WSN en vue d’améliorer la consommation énergétique et la disponibilité du réseau. Cela devrait permettre de prolonger la durée de vie du réseau. Nous proposons différents algorithmes de gestion de la topologie que nous combinons avec le duty cycle. Nos contributions portent sur :

- la proposition d’algorithmes pour la formation d’ensembles en couronnes
- la définition d’algorithmes pour la formation de backbones virtuels disjoints
- la mise en œuvre d’un protocole basé sur les mécanismes de sommeil et relais

3.1 Algorithmes pour la formation des ensembles en couronnes

Nous proposons cinq algorithmes de gestion de la topologie qui sont regroupés en deux classes : ensembles répétitifs et ensembles continus.

3.1.1 Algorithmes de formation d'ensembles répétitifs

Cette famille, composée de trois algorithmes, permet de construire des ensembles de manière répétitive et entrelacée. Autrement dit, dans la plage de transmission de n'importe quel nœud, les membres de tous les autres ensembles sont trouvés. Ainsi, il y a entrelacement d'ensembles dans la zone de couverture de telle sorte que les membres de chaque ensemble forment un lien entre la source et le puits, assurant une connectivité (et une couverture) totale du réseau. Ceci est illustré sur la figure i.

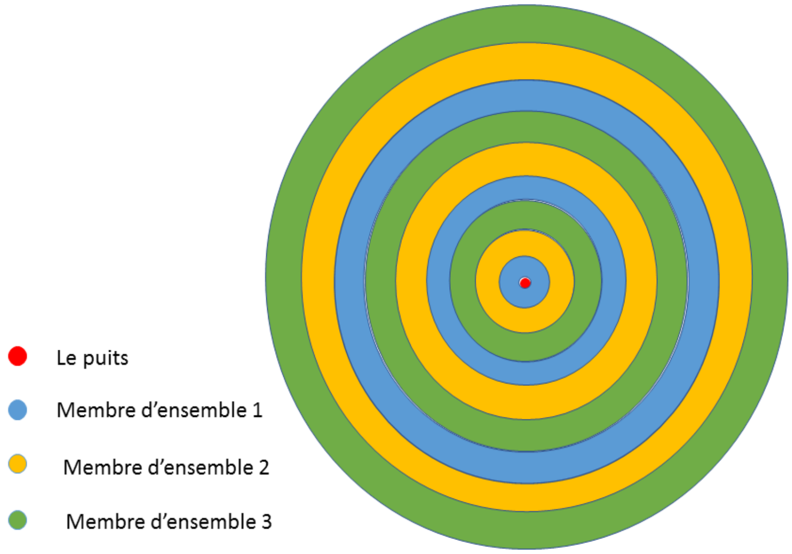


Figure i : Résultat de la formation répétitive d'ensembles

3.1.1.1 Algorithme géographique de formation d'ensembles répétitifs

En utilisant son emplacement et celui du puits, un nœud calcule indépendamment des autres la distance qui le sépare du puits. Il utilise ensuite cette distance, *D*, le nombre d'ensembles à former, *n*, et sa portée radio, *r*, pour déterminer l'ensemble, *s*, auquel il doit appartenir. Nous avons choisi d'utiliser une formule géométrique de coordonnées pour déterminer la distance, *D (P1, P2)* entre un nœud *P1 (X1, Y1)* et un autre (le puits) *P2 (X2, Y2)* comme :

$$D\ (P1, P2) = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2} \qquad \text{(i)}$$

Pour obtenir l'ensemble auquel il appartient, le nœud procède maintenant à la division de cette distance *D*, par sa portée de transmission, *r*, et prend le module de la réponse (partie entière) et le nombre d'ensembles, *n*, à former, comme présenté dans l'équation (ii).

$$s(Pi) \ = \ \left\lfloor \frac{D(Pi, puits)}{r} \right\rfloor \% \, n \qquad (ii)$$

3.1.1.2 Algorithme de formation d'ensembles répétitifs par diffusion

A l'initiation, le puits agissant comme initiateur des ensembles, diffuse un message de configuration avec portée, *r/n* (*r* est la portée de transmission radio alors que *n* est le nombre d’ensembles à former), à ses voisins. Ce message contient le numéro d'ensemble de l'expéditeur, (*s* = 0), le nombre d'ensembles à former, *n*, et un nombre de sauts (*h* = 0). Tous les autres nœuds ont leurs valeurs *s* et *h* initialisées à l'infini. Chaque nœud qui reçoit ce message compare les valeurs *s* et *h* reçues aux siennes. Si celles-ci inférieures aux siennes, il incrémente le numéro d'ensemble reçu et prend le modulo de la réponse et du nombre d'ensembles à former comme son numéro d'ensemble. Il incrémente également le nombre de sauts reçus et le considère comme le sien. Le nœud met désormais à jour le message de configuration avec ses propres valeurs et le diffuse.

3.1.1.3 Algorithme de formation d'ensembles répétitifs par diffusion à signaux multiples

La procédure commence avec le puits diffusant un certain nombre de messages en fonction du nombre d'ensembles à former. Les portées de ces messages sont un facteur du nombre d'ensembles, à former. Pour la formation de deux ensembles (n = 2) par exemple, deux messages de portée r/2 et r, respectivement, sont diffusés. Le numéro d’ensemble de ces messages est également défini en conséquence (0 et 1 respectivement). Ces messages ont les champs suivants : nombre de sauts, h et numéro d’ensemble, s, (h initialisé à 0). Tous les autres nœuds ont les valeurs de h et s initialisées à l'infini. Une fois qu'un nœud reçoit un message de configuration, il compare le nombre de sauts reçu au sien. Si celui-ci est inférieur au sien, il compare également le numéro d'ensemble reçu avec le sien. Si celui-ci est également plus bas, il adopte ces deux valeurs.

3.2 Algorithmes de formation d'ensembles continus

Dans ce groupe composé de deux algorithmes, les nœuds s'organisent en ensembles concentriques autour du puits. Les ensembles sont formés dans l’ordre croissant, selon la zone de couverture. Autrement dit, chaque ensemble occupe une plage de communication. Ainsi, le nombre d'ensembles est proportionnel à la zone de couverture. Ceci est illustré à la figure ii.

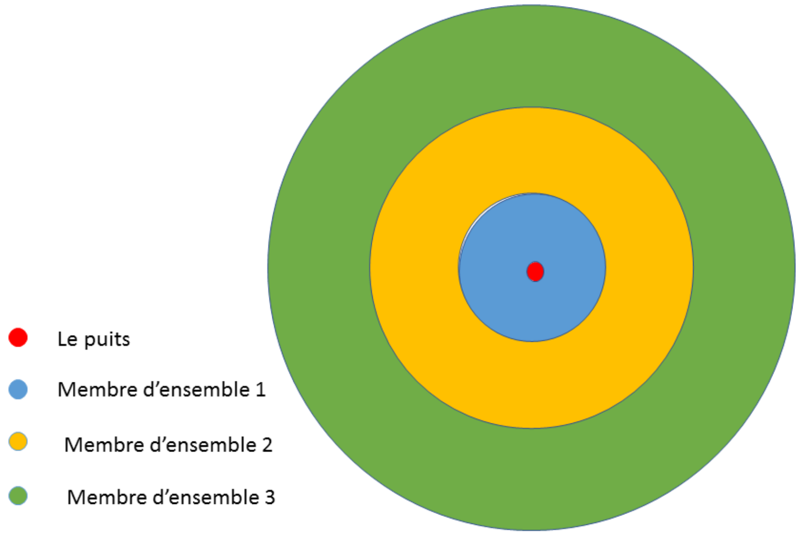


Figure ii : Résultat de la formation d’ensembles continus

3.2.1.1 Algorithme géographique de formation d'ensembles continus

Cet algorithme est un dérivé de l'algorithme géographique de formation d’ensembles répétitifs, présenté ci-dessus, où un nœud est capable de calculer indépendamment et mathématiquement l'ensemble auquel il appartient en utilisant la distance *D*, entre lui-même et le puits, et sa plage de transmission. Dans ce cas, il produit des ensembles de nœuds dans l'ordre croissant par opposition aux ensembles répétitifs.

Pour calculer l'ensemble auquel appartient un nœud, chaque nœud *Pi* divise sa distance au puits par la portée *r* et prend la partie entière de la réponse.

$$s(Pi) = \left\lfloor \frac{D(Pi, puits)}{r} \right\rfloor \qquad (iii)$$

3.2.1.2 Algorithme de formation d'ensembles continus par diffusion

A l'initialisation, le puits jouant le rôle d'initiateur d'ensemble, diffuse un message de configuration à ses voisins à portée, *r*. Ce message contient le numéro d'ensemble de l'expéditeur, (s = 0, pour le puits), le nombre d'ensembles à former, *n*, et un nombre de sauts (h = 0, pour le puits). Tous les autres nœuds ont leurs valeurs *s* et *h* définies à l'infini. Chaque nœud qui reçoit ce message compare les valeurs *s* et *h* reçues aux siennes. Si celles reçues sont inférieures aux siennes, il incrémente le numéro de l'ensemble et le nombre de sauts reçus et les considère comme les siens. Le nœud met désormais à jour le message de configuration avec ses propres valeurs et le diffuse. Le processus se poursuit ensuite jusqu'à ce que tous les nœuds aient choisi leurs ensembles.

3.3 DVBN et DVBN+

La mise en place d’un réseau dorsal virtuel (*Virtual Backbone Network, VBN*) est une technique de gestion de la topologie qui peut considérablement améliorer la diffusion des données et l'efficacité énergétique dans un réseau de capteurs sans fil. Une façon de construire un VBN est d'utiliser la notion de théorie des graphes d'ensemble dominant connexe (CDS). Ici, un réseau WSN est modélisé comme un ensemble de points et de liens d'un graphe. Les points ou sommets représentent l'ensemble des nœuds tandis que les liens ou arêtes représentent l'ensemble des liens de communication non dirigés entre ces points. Il peut y avoir plusieurs CDS dans un réseau, mais celui avec le moins de nœuds est appelé ensemble dominant connexe minimum (MCDS). La construction d'un MCDS est NP-difficile et ne peut donc qu'être obtenue en temps raisonnable par un algorithme d'approximation. La construction d'un seul VBN n'est pas toujours assez robuste. Pour rendre le réseau plus fiable, il est souhaitable de construire plusieurs VBN. À cela s'ajoute l'efficacité énergétique liée à la programmation des activités. C'est-à-dire l'alternance d'activité entre les différents VBN.

Nous nous appuyons sur nos algorithmes de formation d'ensembles continus pour construire des VBN disjoints. L'algorithme proposé, appelé DVBN et sa version améliorée, DVBN+, fonctionnent en utilisant uniquement les emplacements ou les distances entre les nœuds, en particulier par rapport au puits et en choisissant des nœuds dominateurs, en cercles concentriques. DVBN et DVBN+ procèdent par étapes, chaque étape produisant un VBN disjoint des précédents. Le puits commence une étape en sélectionnant le premier dominateur. Ensuite, la responsabilité de sélectionner les autres est transférée à ce dominateur qui passe maintenant au suivant dans son cercle et au suivant dans le cercle d'après. Cela se répète jusqu'à ce qu'il n'y ait plus de cercles. Au tour suivant, le puits recommence avec un nœud non sélectionné et le processus se poursuit. Il s'agit de la version originale de l'algorithme. Dans la version avancée, DVBN +, la responsabilité de sélectionner d'autres dominateurs est transférée à tout autre nœud lorsqu'un Dominateur 1 ne peut pas trouver de dominateur dans le cercle suivant. Cela améliore la connectivité des nœuds au réseau dorsal.

Ces algorithmes sont simulés en C ++. L'analyse des résultats permet d'estimer le facteur d'approximation de l'algorithme à environ 3,5, ce qui est assez bon. La version avancée obtient plus de nœuds en tant que membres du réseau dorsal.

3.4 SREP

Alors que le cycle d'activation peut entraîner un gain d'énergie significatif dû principalement à l'écoute au ralenti, il impose certains inconvénients, liés au fait que pendant qu'un nœud est endormi, il est inaccessible. Ces inconvénients incluent une latence accrue, un faible taux de livraison de paquets, un manque de fiabilité et une indisponibilité du réseau pour communiquer. Nous définissons la disponibilité comme le pourcentage de temps pendant lequel un lien actif existe entre une ou des sources et le puits et nous assimilons la disponibilité au taux de livraison de paquets (PDR). La fiabilité peut également être exprimée comme la probabilité de livraison réussie de paquets d'une source à une destination.

Dans ce travail, nous visons à améliorer l'efficacité énergétique par le cycle d’activation tout en maintenant les performances. Nous proposons le protocole Sleep RElay (SREP), où les nœuds couvrant la zone à détecter sont regroupés en ensembles entrelacés en fonction de leur emplacement dans le réseau. Les nœuds du même ensemble sont en veille en même temps, relayant le sommeil à d'autres ensembles au cours d'une période de cycle d’activation. Chaque ensemble forme un chemin connecté entre l'épicentre et le puits. Un seul ensemble peut être actif à la fois.

À l'aide de NS-2, nous simulons SREP sur l'algorithme de formation d'ensembles répétitifs et confirmons que l'efficacité énergétique est améliorée sans détérioration des performances. Le protocole assure également une distribution uniforme de l'épuisement de l'énergie entre les nœuds. Autrement dit, l'énergie de tous les nœuds s'épuise uniformément.

4. Conclusion

Dans ce travail, nous avons utilisé la gestion de la topologie en combinaison avec le cycle d’activation comme méthode pour économiser l'énergie dans un WSN.

Nous avons proposé plusieurs algorithmes et protocoles qui organisent d'abord les nœuds dans différentes formations topologiques, puis appliquent le cycle d'activation. L’idée générale est de grouper les nœuds en ensembles pouvant couvrir tout ou partie de la zone d’étude. Les ensembles sont activés tour à tour, de manière cyclique, permettant à toute source d’envoyer des données au puits à tout moment. Ces algorithmes ont été validés par des simulations. Il est néanmoins important que ces simulations soient étendues à des réseaux de plus grande taille et que nos propositions soient évaluées sur de véritables bancs d'essai.

Des mécanismes de routage pourraient être associés à DVBN et DVBN+. Cette approche *cross layer* permettrait une meilleure prise en compte de facteurs tels que l’énergie résiduelle et le degré des nœuds pour la construction des ensembles et la définition de leur cycle d’activation.

DEDICATION

To

Mama Magdalene Bessem and her late husband Chief Z. O OBENOFUNDE

Acknowledgement

I will start by giving all thanks and praises to The Heavenly Majesty, The God Almighty, The maker of all things for enabling me to come to the end of this journey that was started more than forty years ago in “Banya School”, Mamfe Town, Southern Cameroon. I had known and prepared my mind from the onset that this last lap was not going to be easy. But I quickly came to the ugly realization that my assertion was a gross underestimation. The financial, administrative, and emotional huddles were almost overwhelming. God almighty has been faithful to me, providing all my needs.

A word of thanks goes to this dissertation examiners, Mr. Hamamache KHEDDOUCI (Prof) and Mohamed MOSBAH (Prof), who took time off their very tight schedule to carefully go through the manuscript. I proceed to thank my supervisors, Mr. Olivier Togni (Prof) and Wahabou Abdou (Assoc. Prof). Haven taken an M Sc. degree a long time ago, it was not easy to find supervisors in a world-class University like ours, who would still think there is something you can offer. I did not only have a wonderful academic relationship but also a friendship that extended to the point where lots of my personal burdens were shared and even borne by them. Thank you all so very much. My hope is that this relationship will continue after now.

The moral and especially financial support that I enjoyed from my friends and family was enormous. Mr. Mbi Tabetando practicalized as always, in very special ways the saying that “a friend in need is a friend indeed”. The courage and means that enabled me to kick-start this program at the financial level in which I was, came almost exclusively from him. Even while in France, he was still bearing lots of my burdens. My close-to-90-years old mother vehemently refused to relinquish her role as a mother. She, like always, solidly stood behind me financially like only a mother could do. This is not something to take for granted. Though I will never be able to repay her, I am happy to make her happy with the completion of this program.

Support from my “big” brothers Bro Besong, Mukete and Egbe, and big sisters, Srs Eli, Es and Fri were simply awesome. Egbe truly keeps justifying me addressing him as my big brother though he is younger than myself. You all (Mbi, Mama, Bro Besong, Egbe, Mukete, Srs Eli, Es, Fri, etc) refused to financially get tired of me despite all the stumbling blocks I encountered that almost pushed even me to give up on myself. I am very grateful to God for He is now empowering me with the ability to start supporting myself. This to me is a way of thanking and assuring you all that I will be fine and that your sacrifices are producing fruits. I pray to be able very soon to also start contributing to support other family members and friends, who might also need a push, like I did.

To my beautiful wife Arah and lovely children Auyaut, Ekepata, Tabi and Ononseh Akem, who have been very patient and supportive especially in prayers, I can only say thank you. Despite my absence and all the hardship that you were forced to go through as a result of this program, you showed a lot of understanding and kept encouraging me by not being wayward. I thank God our lives are becoming better after all these sacrifices.

Some other friends and family members are worth mentioning here. I am mainly talking about Hubert Besong, Bro. Egbemba Ojongmboh, Donald Besong, Sandys Egbe Tabi, Valentine Ayuketah, Felix Ojong, Obale Oben (SOBCAM), Samy Besong, Tambe Tanyi, Wandong Sidonie (Cici Bali), Aldrine Enow, Tabi Chama and a host of others for their contributions in different ways to my wellbeing in Europe these few years. Accent goes to Tabi Chama, Aldrine Enow and Tabi Sandys who went extra miles to make my life quite comfortable.

This section will not be complete if even because of space, I don't mention my "father", Mr. Michel Otero (Assoc Prof) and his wife Gabby. God has a way of bringing destiny helpers to people's path. All I did was look up vacancy announcements on the University housing website from Buea. It was sheer destiny that landed me on these people's property amongst thousands of vacant students' accommodation in Dijon. I got myself a family in a far-away land that I totally relied on. To you, I pray for God's richest blessings.

List of Figures

List of Tables

Table of Contents

Chapter 1

General Introduction

Wireless sensor networking is strongly establishing itself as a ubiquitous technology present in almost every area of human endeavor. The applications for which this technology is used are so varied and cover a very broad range, from military surveillance operations to civilian uses in medicine, route safety and engineering, just to name a few. The network is made up of numerous, cheap and tiny devices referred to as sensor nodes, composed of a sensor and a mote. The sensor is a device that detects event occurrences like changes in temperature, the presence of fire, the presence of light, etc. The mote is the component that directly receives data from the sensor, processes it, and sends to a base station through a wireless link. The network is introducing a new paradigm in computing where computing systems directly interact with the environment in which they find themselves and produce results, with little or no human intervention (Khatarkar and Kamble, 2013). Instead of humans to input the physical or environmental conditions into the computing systems, the systems themselves directly read these information from the environment and process them. The results are either directly used to adjust the physical condition or sent to a remote location for some kind of exploitation.

A wireless sensor network (WSN), is a wireless interconnection of sensor nodes capable of sensing, processing and communicating data to a processing center for analysis or any other kind of reaction. The result from these processings could sometimes be used to adjust the condition that caused it. A node that senses an event is usually referred to as a *source* while the destination node is the *sink* or base station and is usually connected to an infrastructure network and to power supply. The processing center may be far from the area being sensed, sometimes reachable only through the Internet. The sink and the processing center could be the same.

1.1 The Research Question

Many advances have been made over the last few years on WSN technology. Despite all the tremendous advances, energy efficiency still remains a big challenge. This is because nodes are mostly battery powered with the possibility of recharging or changing them almost inexistent once they are depleted of their energy. Other sources of energy are now possible but are either yet to grow to maturity, not very appropriate vis-à-vis node sizes or the ease of network deployment. This situation is aggravated in cases where the network is deployed in inaccessible or hazardous environments. More so, the sheer number of deployed nodes may make it almost impossible to locate each for battery recharging or change after deployment. Therefore, in most applications, once a node's battery becomes drained of its energy, the node becomes useless. Thus, it is important to keep the battery alive for as long a time as possible. This will increase the lifespan of nodes and by implication, the network as a whole. Energy efficiency is thus a very critical issue in this type of network, calling on all its subsystems to be efficiently exploited.

Many mechanisms operating in diverse areas of network operation (deployment, energy sources, Operating Systems, etc.) or different layers (link layer, routing, transport, etc.) have been proposed that lead to energy efficient exploitation of WSN. Amongst these is duty cycle, which is one of the most simple and efficient and can be applied in various areas of network operation. Nodes (or their subsystems) periodically or aperiodically alternate between being active and inactive. The intention is to have nodes active only when there is need to communicate. This save a lot of energy especially from idle listening which is the most normal state that nodes in these networks find themselves. Though being quite simple duty cycle introduces a number of performance degradations based on the fact that once anode (or its communication module) is off, it cannot communicate. Increased latency and lower packet delivery ratio follow suit.

This dissertation proposes new topology management algorithms and protocols that better organize the network to conserve energy. We exploit the network in such a way that less energy is expended, thus extending the lifespan of the network. Duty cycling is combined with topology management at different levels for energy efficient network exploitation. We start with easy topologies and progress to more complex topologies (ones that are NP-hard to construct optimally). Then we look at the levels of energy gain that can be brought into the network.

1.2 Our Contributions

This dissertation entitled Topology Management in Wireless Sensor Networks is aimed at proposing topology management schemes for enhancing energy efficiency in WSN and to show how these enhance energy efficiency. It is the way that nodes arrange themselves to respond to duty cycling or its schedule that interest us. We are interested in duty cycling and its effects in terms of energy efficiency, latency and packet delivery ratio.

We propose different topology management algorithms as a means of improving energy efficiency in wireless sensor networks. Nodes organize themselves into sets of different formation. One formation is repetitive in nature while the other is continuous. These formations are then used in another proposal where the sets sleep in relay in such a way that the network performance is not deteriorated. In particular, our contributions are enumerated in the sub sections below.

1.2.1 Set Formation Algorithms

We design algorithms where nodes in the network arrange themselves into sets with different formations. In repetitive set formation algorithms, nodes arrange themselves into repetitive and interleaving sets such that one set allows a continuous path between any source and the sink with other sets interleaved in between. This configuration is directly usable in the sleep relay protocol being proposed. In the continuous set formation algorithms, nodes arrange themselves into continuous sets between the source and the sink. Nodes belonging to a single set traverse a communication range. The number of sets depends on the coverage extent of the network. This formation is the building block for constructing disjoint virtual backbone networks.

1.2.2 Disjoint Virtual Backbone Networks Construction

Two algorithms are built from any of the continuous set algorithms earlier proposed to construct disjoint virtual backbone networks, each covering the whole area. The selection of backbone nodes (dominators) is done using only node positions. In the first algorithm, only the sink and dominators with certain properties can initiate the selection of other dominators. In the second, an enhanced version, the selection role is not only limited to specific dominators but open to any other dominator who can, once the special dominator cannot. The disjointedness is aimed at ensuring activity scheduling to be used in the sleep relay protocol being proposed. We evaluate and compare the performances of the algorithms.

1.2.3 Sleep Relay Protocol

This is an energy efficient proposal that combines duty cycling and topology management. After nodes have arranged themselves into repetitive sets or into virtual backbone networks as presented above, they start sleeping in relay in such a way that members that are active span from the source to the sink. The duty cycle can thus be divided according to the number of sets and each set active in one slot of the division. This ensures that though the active time of nodes in sets is factored, the cumulative active time stays the same, not deteriorating latency and availability which is measured using packet delivery ratio. The dividing of duty cycle amongst sets can lead to huge savings in energy and prolong the lifetime of nodes and the network. This saving is confirmed using simulations.

1.3 Some Tools and Concepts

To conclude this chapter, it is important to present some basic information on some tools that have been used to evaluate or validate our proposals. Simulators have been developed in order to initially test and validate algorithms. This is so because testing an algorithm or a protocol in a real-life environment can be quite expensive and time consuming. With a simulator, an algorithm or protocol can be cheaply and quickly tested in a laboratory. Result can then be evaluated and validated before the protocol is implemented on a real network.

The theoretical computer science concept of algorithm complexity analysis will be briefly presented in this section. The introductory concepts of graph theory are left out of this chapter to be presented in Chapter 5.

1.3.1 NS-2

Network simulator version 2 or NS-2 (Altman and Jimenez, 2012) is a very popular discrete event simulator used both by the academic and industrial communities for research and education. Its free and open source nature has enabled it to receive enormous and free contributions for its growth from developers all over the world. NS-2 has been used to validate many protocols and standards. It started and is backwards compatible with NS-1. A version 3, NS-3 has been developed with quite a different architecture and thus non compatible with NS-2. It is focused on Internet research with emphasis on layer 2 and 3 of the protocol stack.

NS-2 is based on 2 languages:

1. An object-oriented simulator written in C++: This is the core of the simulator itself. It defines the way the simulator works and how it presents interfaces, animations, graphs, etc. Thus, to change or add functions to the simulator, one must proceed with the C++ section of the simulator.
2. An object-oriented interpreter extension of Tcl (OTcl): Tcl (Tool Command Language) is a very popular programing language with very simple syntax. It allows a very easy integration with other languages. Tcl, through the OTcl linkage, interacts with compiled C++ objects to build (describe) the network to be simulated. Here, one can describe topologies, links and the various properties of the network to be simulated.

Simulation scripts are written in Tcl and ran on the shell. They produce results that can be viewed and analyzed using accessories like a network animator (NAM) and a graph viewer (Xgraph) that comes with the simulator.

1.3.2 ViSiDiA

ViSiDiA, standing for Visualization and Simulation of Distributed Algorithms is a simulator aimed at studying the execution of distributed algorithms (Aguerre et al., 2012) and mainly used for research and education. It is made up of 3 main components:

1. A simulation core: This describes a node, its links and other properties. Each node executes a copy of the algorithm.
2. A GUI: A graphical user interface provides an easy means of writing an algorithm and for drawing graphs. It is also used to visualize simulation of implemented algorithms and to study distributed systems.
3. An API: A java-based application program interface used to write algorithms to be simulated. The API are java methods that are used to describe procedures to be executed by each thread (node).

1.3.3 Algorithm Complexity Analysis

It sometimes become necessary to compare the performance of algorithms. An algorithm is a set of steps specifying how to solve a problem. It is these steps that are coded in different ways (syntaxes) using different programming languages to constitute a computer program. It is common knowledge that there is never a single means of solving any one problem. This also applies to algorithms where there are always many different algorithms that can be applied to solve the same problem. It becomes important sometimes to decide which algorithm is best amongst these. This measure of “best” should neither depend on the programming language nor type of computer (processor) used as these tools depend on many other factors like programming paradigm of computer programs, processor speeds, instruction set architectures, ISAs, etc.

The efficiency of an algorithm depends on the time it spends in solving the problem confronted. In other words, it is a measure of how fast a program runs. This is the object of algorithm complexity analysis. There are programs known as profilers that can measure how fast an algorithm runs in milliseconds (Zaparanuks and Hauswirth, 2012). These programs can even go further to detect bottlenecks in the program for eventual optimization. Though they produce nice results, these programs are not relevant to algorithm analysis, since algorithm analysis is more concerned with comparison of algorithms.

There are some problems that are very complex to solve. Their solutions can only be approximated in polynomial time using approximation algorithms since optimal solutions are very hard to get. These problems are referred in the literature as NP-hard. The quality of a solution of an NP-hard optimization problem can be measured by its **approximation ratio** (or factor) which is for a minimization problem, the ratio of the proposed solution to an optimal solution (Vazirani, 2003).

1.4 Report Organization

This first chapter gives a brief overview of the dissertation. Some tools and concepts used in the dissertation are also presented here, including our scope of work. Chapters 2 and 3, give the state-of-the-art of various technologies related to this dissertation. In Chapter 2, we discuss duty cycling and energy management in wireless sensor networks, in some detail. In Chapter 3 we classify duty cycle protocols according to layers depending on the tasks the protocols seek to accomplish using duty cycling. In Chapter 4, we present five topology management algorithms and demonstrate some. In Chapter 5, we build upon part of the works of Chapter 4 to construct multiple disjoint virtual backbone networks. In Chapter 6, we propose SREP, a protocol in which nodes are put to sleep in relay. That is, when one set is awake others are asleep during a duty cycle. This protocol is validated with NS-2 using some of the algorithms proposed in Chapter 4. The report is concluded in Chapter 7 with a resume of our contributions and some propositions on the way forward, made.

Chapter 2

Overview of Wireless Sensor Networks

In this chapter, while we review the literature of wireless sensor network technologies in general, but concentrate on the aspects that are more directly related to our area of interest which is the application of duty cycling on topology management as an energy saving measure.

2.1 Wireless Sensor Networks

Physical properties of our environment like temperature, pressure, wind speeds, seismicity, etc., are always changing. These changes can positively or negatively affect our very lives. It is incumbent therefore on us to be able to monitor, react to or control these or their effects. This involves gathering and analyzing information about them in a timely manner. Sensors are devices that gather information around or within their vicinities and convert these into virtual (electronic) form for some action or reaction (Dargie and Poellabauer, 2010). They bridge the gap between the physical and the electronic world and are therefore special types of transducers that convert energy from some other form to electrical (digital) form for easy analysis by humans. The analyses may take place at a location far from where the sensing occurred, all these with little or no human intervention (Khatarkar and Kamble, 2013). This means that nodes should possess sensing, computation and communication capabilities, in addition to power supply. Nodes may also have additional application dependent components such as location finders, power generators and mobilizers.

It is common to find many sensor nodes working together to accomplish a given task (Boukerche and Sun, 2018). This might stem from the fact that either the environment being sensed is too large to be monitored by just one sensor, the process quite complex involving the monitoring of several physical properties or the area being monitored inaccessible. Each of these scenarios require correlation of the data collected by each sensor node to be sent to a base station for analysis, storage or further processing. Thus, there is need for a network which can be either wired or wireless. While both types of networks have their advantages and disadvantages, wireless sensor networks are favored because of cost, deployment and usage (Shi et al. 2018). Their main challenges include the wireless channel and energy efficiency. Wireless interconnection can either be by acoustic, infrared, microwave or radio links (Karray et al. 2018). Radio is favored as the others have a few more disadvantages. Infrared for example, requires line-of-sight. Because of these, communication in wireless sensor networks is mostly wireless, using radio link.

A wireless sensor network (WSN) is therefore a network of cooperative sensors nodes, wirelessly interconnected. It is a special type of ad hoc network that can be mobile or fixed. In a mobile sensor network, the nodes can move about, while in a fixed network, node movements are either non-existent or minimal. In this network, a node that senses and thus has information to send is a source, while the last one that act as a base station and mostly connected to an infrastructure network, is the sink.

The areas of application of WSNs are quite numerous and still growing. They include environmental monitoring, resource tracking, assisted living and e-health, critical infrastructures monitoring, homeland security, smart grid, etc. (Camillò et al. 2013). Technological advances especially in the fields of microelectronics have contributed immensely to these growths. Generally, wireless sensor networks have the following distinctive characteristics as compared to wireless networks in general:

- Battery-powered nodes: Sensor nodes are usually battery-powered with the batteries very difficult to be changed or recharged.
- Reliability: Nodes are usually exposed and thus prone to physical damages.
- Data redundancy: Many of the densely deployed nodes usually sense the same event, producing the same results. Thus, there might be some level of data redundancy in the network.
- Ownership: The network is usually privately owned as opposed to public ownership found in a network like the public switched telecommunications network (PSTN).
- Self-configurable: The usually randomly deployed network nodes are left to

configure themselves into a functional network.

- Frequent topology change: Network topology changes frequently due to the node failures, damage, addition, energy depletion, or channel fading.
- Application specific: A sensor network is usually designed and deployed for a specific application. Thus, their design requirements change with application.
- Many-to-one traffic pattern: In most sensor network applications, data flows from multiple nodes towards one sink, exhibiting a many-to-one traffic (convergecast) pattern.
- Data-centric application: Information delivery usually depend on the data rather than on address of nodes

In the design and application of WSN, energy efficiency is very critical (Anastasi et al. 2006). This is mainly based on its being mostly battery powered with difficulties in replacing or recharging, once depleted. Other sources of energy like energy harvesting and transference are now possible but they too have their challenges.

2.2 Sensor Node Organization

Many subsystems work together to enable a wireless sensor network perform its assigned task. We take a brief look at some of these.

2.2.1 Hardware

It is improper to discuss WSN hardware without briefly reviewing the organization or architecture of a sensor node. Figure 1 presents a typical node organization. A tiny sensor node comprises a sensor and a mote. The sensor is the device that senses or detects the occurrence of an event or a change in the environment. This information is then translated into data and passed on to the mote whose job is to process it. There can be different sensors for different purposes mounted on one mote. The mote is the computing elements of the node and sometimes referred to as Smart Dust.

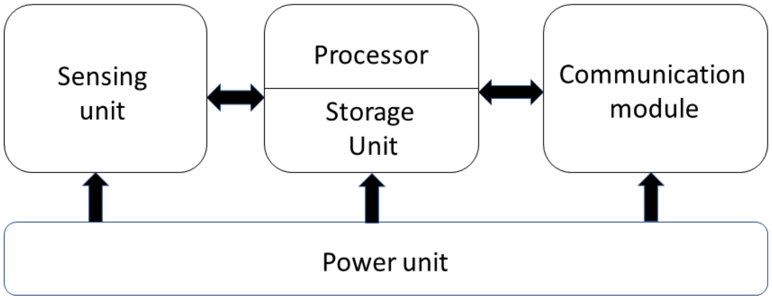


Figure 1:Typical Sensor node architecture

Node architecture depends to a large extent on the application to which it is to be applied but is generally organized as presented in Figure 1.

- **Processing unit**: It is the heart of the node and does all the processing task. For example, it does data management, routing, power control etc. It also manages sensing, transmission and every other process. The electronics of this unit is quite diverse and depends mostly on the application. It ranges from microcontrollers, field programmable gate arrays (FPGA), system on chip (SoC), multicore, complex programmable logic devices (CPLD), etc. (Karray et al, 2018). A microcontroller, for example, is a processor and memory integrated on the same chip with an input/output system.
- **Storage unit**: It generally consists of three components especially for early motes: the RAM/SRAM (static random-access memory), Flash memory and EEPROM.
- **Communication module**: This permits nodes' intercommunication and data exchange. It is a high power-consuming unit; hence, its choice, design and management are quite crucial especially to our end. This is because it is on this module that duty cycling will be applied. Duty cycle is the alternation of the activeness of a system, aimed at conserving energy. Communication is generally Radio Frequency (RF) based.
- **Sensors**: These interface the nodes to the outside world they find themselves in. They collect data from the monitored physical phenomena and translate them into electrical signal to be analyzed by the processor. The electrical signal is usually first digitized using Analog to Digital Converters (ADC). Some nodes incorporate pins and ports to interconnect other sensors and/or sensor boards. We can cite for

this type MICAz, BEAN, etc. Some others integrate and provide pins and ports for extensions as in AWAIRS, Medusa MK-2 and TelosB (Karray et al., 2018). The sensing unit could be a technological bottleneck as it is much slower than other (semi-conductor-based) components (Khemapech et al., 2014).

- **Power Unit**: This unit is generally the most critical in WSNs. It powers all components of the node and defines the major system constraint: the available power. The unit usually consists of a battery (mostly AA) whose choice is determined by chemistry (Khan et al., 2015). Power is dissipated when the system is in use. For many applications, recharging or replacing the battery of thousands of nodes is time-consuming, costly or impossible for nodes deployed in harsh environment. Hence, preserving the battery power for months or years is important. The extension of battery life and thus that of the network is the main objective of this work.

2.2.2 Software

The broad term software in electronics is generally aimed at user convenience. It is usually organized into layers, with the operating system (OS) lying at the bottom, interfacing with the hardware. Owing to fact that sensor networks are mostly used for specific applications, most tasks can be performed directly by the OS. The OS in WSN is thus tasked with facilitating communication between the programmers and the hardware, managing the resources of the system and providing high-level interfaces. These responsibilities also involve managing power consumption and every other resource. These call for an OS that is typically less complex. WSNs are application specific and mechanisms such as virtual memory, for example, are either unnecessary or too expensive to implement. These mean that, the complexities of general-purpose OS, like Windows and Linux must be reduced, more strongly resembling embedded system OSs (Levis et al. 2005). Typical WSN OSs include:

TinyOS: This is an event-based operating system specifically designed for sensor networks (Levis et al., 2005, Raman 2002). It requires minimal hardware and designed to support concurrency as required of sensor networks. Its uses Active Message Communication model which favors non-blocking applications and Multihop ad hoc routing. TinyOS is an open source OS, developed at the University of California, Berkeley. It has enjoyed a lot of support from the US's Defense Advanced Research Project Agency (DARPA), the National Science Foundation (NSF) and Intel Corporation.

Contiki: This is a lightweight operating system for wireless sensor network that provides a rich execution environment for tiny devices (Dunkels et al., 2004). It can dynamically load and unload individual programs and services. Though the kernel is event-driven, the system supports preemptive multi-threading that can be applied on a per-process basis. This is implemented as a library linked only to programs that explicitly require multi-threading. Contiki is implemented in C and has been ported to several microcontroller architectures.

Other OSs in common use for WSNs include SOS, free RTOS, Mantis, LiteOS, modified embedded system OSs like eCOS and uC/OS.

In addition to OSs, there are several middleware developed for WSN. A middleware is a software that provides services to software applications beyond those available from the operating system. A middleware makes it easier for software developers to implement communication and input/output functionalities on nodes, so the device can better perform its specific task(s). Some very common ones include agilla and sensorware (Lingaraj et al., 2018).

2.2.3 The Wireless Channel

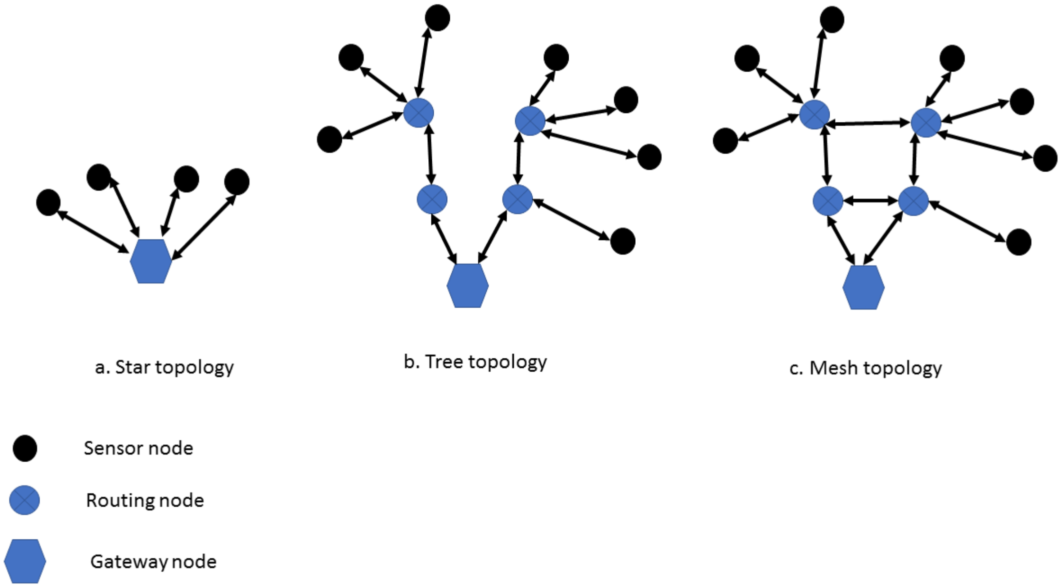
Being a wireless communication system, wireless sensor networks also experience the oddities associated with the wireless environment. These include:

- **Spectrum:** This limited resource can generally be assigned on an exclusive, or shared basis. Exclusively allocated frequencies are payable and can be quite expensive. Share spectrum on the other hand is free and thus subject to interferences. It must therefore be strictly managed in order to avoid interferences from multiple users. WSN uses the Industrial Scientific and Medical (ISM) band within the shared spectrum.

- **Multipath Propagation:** In the course of propagation from a transmitter to a receiver, a radio signal may undergo several different processes. This include line-of-sight LOS, reflection, diffraction, scattering, etc. These phenomena might cause its components to travel through different paths of different lengths, phases, amplitude, distortions, etc. on its way to the receiver. Thus, the intensity and path length of these different components may vary at different points. This is called multipath propagation and can cause the signal to be received differently at the receiver. At worst, there can be destructive interference, when the signals arrive symmetrically out of phase (180^o) and are subtracted from each other. Constructive interference occurs when they arrive in phase and add up. The end-result is fading of various degrees for analogue transmission and inter-symbol interference for digital transmission.
- **Noise and Interference:** Noise is the introduction of unwanted signal into a transmission. Transmission quality is usually measured using *Signal-to-Noise Ratio* (SNR) at the receiver. Interference occurs when more than one signal with the same frequency is present in the system. This makes it difficult for the receiver to isolate one signal from the other. This is what happens during a collision and may lead to retransmissions.

2.2.4 Deployment & Topology

Wireless sensor network deployment, the setting up of an operational wireless sensor network in a real-world environment is labor intensive and cumbersome. It can be further complicated by many unforeseen situations in the environment that include topography, the wireless channel and many others that cannot be modelled with exactitude in a laboratory (Ringwald and Romer, 2007). Many techniques can be applied for network deployment (Sharma et al., 2016). In terms of placement strategy nodes can be deployed either manually (orderly) or at random like dropping from an aircraft. These produce different topologies. Topology is the physical or logical layout of the network. An appropriate topology will go a long way to ensure adequate coverage of the targeted area and ease communication (Shi et al., 2018). There are three common topologies for WSN: star network, tree network, and mesh network (Figure 2).


a. Star topology

b. Tree topology

c. Mesh topology

Figure 2: Common Topologies for WSN

A star network consists of two kinds of nodes, sensor nodes and one gateway node. The sensor nodes transmit data only to the gateway node which is one-hop away. No communication is allowed between the sensor nodes themselves. The gateway node acts as the base station, receiving data from all sensor nodes and transmitting for processing. Its role is the same in all three topologies. The star topology is the simplest in terms of structure and is also energy efficient as it does not require a lot of route planning. Its main disadvantage is that of limited distance between the furthermost node and the gateway. Therefore, it is often applied in small-scale scenarios such as offices or courtyards and not suitable for large-scale scenarios like environmental surveillance.

A tree network consists of three kinds of nodes: one gateway node, routing nodes and sensor nodes. Routing nodes are structured in a tree format rooted at the gateway node. Here, like in a star network, communication between the sensor nodes is not permitted. Sensor nodes can only transmit data to their own parent nodes. The parent or routing nodes play the very important roles of route planners, running the routing algorithm to find the best way to a gateway node. A tree network has drawbacks of complexity of structure and low energy-efficiency due from route planning. This is related to prevention of loops and establishment of relationships between parent and child especially at tree setup. A tree network overcomes the disadvantages of limited distance of communications associated with a star network and therefore more suitable for application in large-scale scenarios.

Usually, the distance between a sensor and the gateway should not be more than 10 in order to balance complexity and cost.

A mesh network has the same three kinds of nodes as a tree network having the same functions. Nevertheless, routing nodes can communicate between themselves. This makes a mesh network quite robust in communications, but at a cost of higher energy consumption due to much more calculations for complex routes.

2.2.5 Synchronization

All computing devices are equipped with oscillators that guarantee their hardware clocks (Goyal et al., 2017). Different oscillator properties will lead to each clock keeping its own detailed timing. In a network, clock or time synchronization aims to provide a common timescale for each local clock (Skiadopoulos 2019). This helps the different network processes to correctly analyze data and even predict future system behavior.

2.2.6 Localization

Localization is determining the physical position of nodes after deployment as this is very important for some applications (Chelouah, 2018). It is not only important sometimes for a node to know its position but also that of its neighbors. Node's location helps in routing which can bring about savings in energy and thus prolonging lifetime. Many algorithms have been put in place for these. In (Amri et al., 2019), a proposed mechanism relies on a weighted centroid localization technique, where the positions of unknown nodes are calculated using fuzzy logic method. In many other algorithms, nodes are equipped with GPS (Global Positioning System) or any GNSS (Global Navigation Satellite System) modules in order for them to know their location. Since it might be too expensive to include these on all nodes, some nodes can use this resource on their neighbors, referred to as anchor nodes, to compute their own locations (Sharma and Kumar, 2018).

2.2.7 Data Aggregation and Dissemination

The data being transmitted from multiple nodes may often be quite redundant and/or too voluminous for relay nodes or even the base station to handle. There might therefore be the need for combining these into a high-quality information. This is accomplished through data aggregation. **Data aggregation** can be defined as the process of combining data from multiple sensors to eliminate redundancies, estimate the correct data and provide fused information to the base station. This also saves energy and prolongs network lifetime (Goyal et al.,2017). Data aggregation can be used as a supplementary technique to overcome some environmental issues that may abound in the area being monitored. These are issues linked to the fact that sensor networks are inherently unreliable and certain information may be unavailable or expensive to obtain, like the number of nodes present in the network and the number of nodes that are responding to an event. It might also be difficult to obtain complete and up-to-date information on neighboring nodes. Aggregation may be favored by clustering, sometimes required to conserve energy.

At the opposite end of data aggregation is **data redundancy**, the sensing and transmission of similar data in a network (Verma and Singh, 2018). Data redundancy ensures reliability which plays a very important role in the analysis, monitoring and forecasting of system behavior. Bad quality data may provide erroneous result in decision schemes. In WSNs, nodes are densely deployed in a region to collect information. Many sensor nodes sense the same data and forward to sink leading to redundancy especially at the sink. More useful information can then be picked out by sampling the data from these multiple and redundant sources. This results in more accuracy, reliability and security. Meanwhile, limiting the number of sources to receive data from helps in energy savings since most of the energy of the relay nodes or the sink that could be wasted in dealing with the redundant data, is saved (Verma and Singh, 2018). Nevertheless, data accuracy still needs to be preserved even if these increase network cost and/or time. Therefore, there is need for a mechanism in which we can extract extra information from the redundant data and be able to provide a more consistent, accurate and reliable data set in an energy efficient manner.

Data dissemination is when data and queries are broadcasted in the network. A node interested in an event can broadcast its interest to its neighbors who continue the broadcasting until every node in the network receives (Merzoug et al., 2019). A node that has the requested data responds to the request using the same route from which it received the message.

In data dissemination, all nodes including the base station can request data. It is therefore a bidirectional process. In data aggregation the aggregated data is periodically transmitted towards the base station. It is thus a unidirectional process. In addition, while data is transmitted periodically in data aggregation, it is always transmitted on-demand in data dissemination.

2.2.8 Wireless Sensor Network Reference Model

Being a communication system, WSN design also follows the network design abstraction of the layered model. A network model or architecture is a blueprint used to design a network. It provides a better understanding on how the network operates. The communication task of a communication system is broken down into several subtasks and related subtasks stacked into layers. At the top layer where users interact with the system, the tasks are more user-related and accomplished by application programs. These software packages use the services of other software packages and a lower layer, which also use the services of other packages until at the lowest layer, where an actual signal(s) is transmitted from one computer to another. These are the activities that are visualized in layers where, as can be seen, begins at the top with more software related activities, then move down to more hardware related activities.

In addition to the advantage of having the communication task as manageable subtasks, layering makes it easy for new services to be added, as they appear. This is because layering establishes clear interfaces between themselves. The different tasks of a layer are accomplished by **protocols**. The protocols provide communication services that higher-level objects such as application processes (accomplished by application protocols) or lower level ones use to exchange messages. Though the functions of each layer have generally been well specified, the lines between them may sometimes be quite hazy.

Several network models have been defined with the two most popular being the Open System Interconnection (OSI) model with 7 layers, and the Internet model with 4 layers. Nevertheless, because of the specific nature of WSN, its architecture deviates a bit from these (Alkhatib and Baicher 2012). Its layered design model consists of five layers as presented in the Figure 3.

The application layer combines the functions of the application, presentation and session layers of the OSI model. The functions of the transport, network, data link and physical layers are the same as for the OSI model. We provide a brief description of the functions of each of these layers.

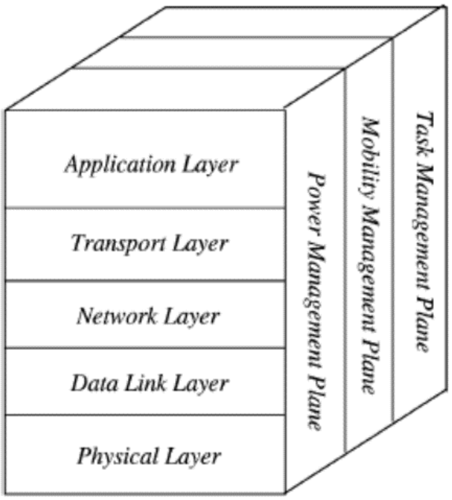


Figure 3: Wireless sensor network architecture (Alkhatib and Baicher 2012)

The Application layer: It directly interfaces with users. Its functions include application processing, data aggregation, query processing, external database access, security, etc.
The Transport layer: Its functions include data dissemination and accumulation, caching and storage
The Network layer: Its functions include adaptive topology management and routing.
The Data Link layer: Its functions include channel sharing (MAC), timing, and locality.
The Physical layer: Its functions include communication channel actuation and signal processing.

Added to these layers are the three cross layers planes shown above, constituting the management protocols. These include power, mobility and task management planes, all used to manage the network and make the sensors work together in order to increase the overall efficiency of the network.

- Power management: manages how a sensor node uses its power (Alkhatib and Baicher, 2012).

- Mobility management plane: detect sensor nodes movement. Nodes can keep track of neighbors' movements and power levels (for power balancing).
- Task management plane: schedule the sensing tasks to a given area. Determine which nodes are off and which ones are on.

2.3 Energy Management

Despite all the tremendous advances that has been made on wireless sensor network technology, energy has not followed suit. Energy efficiency which has to do with both efficient provisioning and expending of energy, has not been improving (Khan et al., 2015) at the same rate as other WSN subsystems. Though lots of work has been done recently on new methods for energy provisioning, battery remains the most viable source. This is mainly owed to its technological maturity and its size in relation to that of nodes. But a battery will eventually be drained of its power and the node will become useless if the battery cannot be either recharged or replaced. The deployment environment and the sheer number and size of nodes may make it difficult or even dangerous to access nodes after deployment for battery change or recharge.

Energy harvesting, a new technique in energy provisioning, entails scavenging energy from the environment with typical example sources being solar, wind and heat. By harvesting environmental energy, a sensor network can operate for almost forever (Valera et al., 2014). The main difficulties are the facts that nodes might be deployed in areas where these resources are either not available or reachable continually. Another issue is estimation of the right amounts to be harvested (Mothku and Rout, 2018). Radio frequency (RF) energy can also be harvested, the basic principle of which involves mutual inductance by magnetic coupling, where a voltage is generated when a time varying RF field passes through an antenna. Nevertheless, the supply of environmental energy is never continuous, thus there is need for a storage mechanism. But the size of the equipment required for scavenging and storage may be too cumbersome for nodes.

A quite recent development in energy provisioning is energy transference. This is the wireless supply of energy to deployed nodes. This most promising energy provisioning technique can be achieved using a mobile node (Tsoumanis et al., 2018) that moves around the network, transferring energy to others by electromagnetic (EM) waves. The technology includes EM radiation and magnetic resonant coupling. EM radiation is mostly provided through LASER reflection. The magnetic resonance-based technology is known as Witricity. These two technologies lack efficiency in terms of distance, since transference is limited by distance. Nodes therefore need to be deployed in proximity. The technologies require further improvements, specifically in sparse networks, where nodes may need to transmit energy to others farther than the maximum distance supported. There is the need to further explore alternative choices such as microwave power beaming and reflection-based technologies, which provide improvement in terms of distance. Like with energy harvesting, the issue of equipment size being a hindrance, also abounds here.

Energy efficiency in terms of expenditure in WSN is quite critical. It is clear from the limitations associated with energy sourcing, especially those linked with batteries, that much cannot be done in this direction by someone specialized in networking. Our only area of intervention is in efficient energy expenditure. These entails rigorous and efficient exploitation of the network if its lifetime must be maximized. We therefore concentrate on the communication subsystem which is one of the biggest consumers of energy (Rezaei and Mobininejad, 2012) of the WSN system. We are implicitly referring to communication protocols.

Like in all wireless networks, energy can be lost through interference, multipath propagation, transmission over a distance, etc. While we will not touch on these areas, it is enough to say that reducing transmission power or range is a major means of conserving energy in a wireless network (Díaz-Ibarra et al., 2019). This could imply multi-hop transmissions. Nevertheless, in WSN there must be a compromise between having one long hop and multiple short ones (Krishna et al., 2016). The later cannot only lead to delays as each hop does some processing but may also have issues with multiple access.

Many techniques exist to conserve energy in WSNs which can be associated to the different layers (Rezaei and Mobininejad, 2012) of the protocol suit. To mention just a few, transmission rates and ranges are physical layer responsibilities. Radio operations can be optimized (Rault et al., 2014) using schemes like modulation, error coding, antenna design, etc. in this layer. Multiple access control, control packets, retransmissions, idle listening, etc. are energy conserving issues that are handled at the data link layer. Proper design of routing protocols and topology management are the responsibilities of the network layer. Congestion control and load sharing are taken care of at the transport layer. Energy can be conserved by data aggregation (Rezaei and Mobininejad, 2012) at the application layer. Here information from multiple nodes are gathered to form a meaningful whole, with reduced size.

Energy conservation protocols or mechanisms in WSNs can generally be classified using the following three main approaches:

- Data driven approaches
- Mobility-based approach
- Duty cycling

Figure 4 presents a high-level energy management taxonomy.

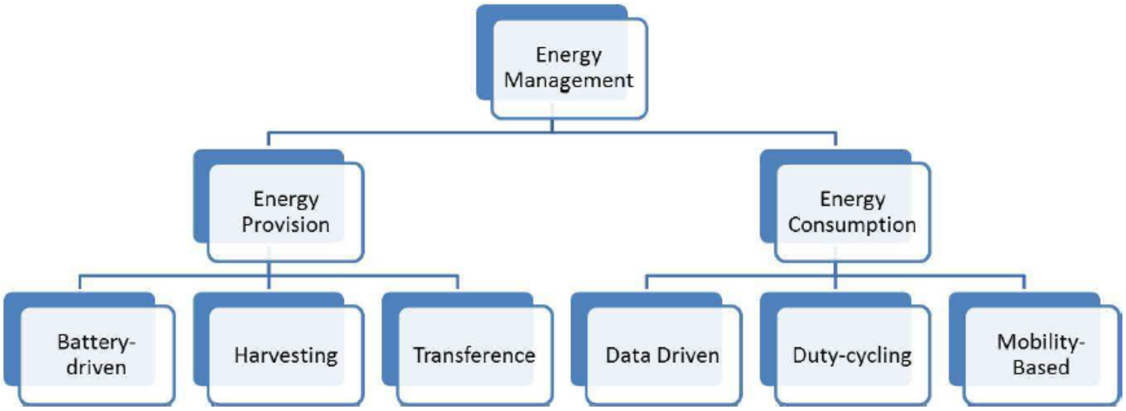


Figure 4: High-level energy management taxonomy (Khan et al., 2015)

2.3.1 Data Driven and Mobility Approaches

Data driven approaches do not only ensure that useful data is sensed but minimal data quantity is communicated to the sink (Krishna et al., 2016). A threshold quantity must be reached for sensing to be triggered. The point in the network where data is processed also has a significant impact on the energy consumption of the network. Data can be processed locally (at every node or cluster-head) (Da Silva et al., 2013) instead of only at the sink (Elshrkawey et al., 2018). That is, some pre-processing of data may be done at some intermediate nodes before sending to the sink. The sink too can do some data filtering. Other data-driven schemes include data coding, decoding, aggregation (fusion), compression, predictions, etc.

Energy savings by mobility is based on the fact that mobility brings communicating nodes closer to each other, thus saving energy on transmission distance. Mobility of nodes and especially those near the sink and or the sink itself can take care of a number of things. The first is funneling effect experienced by static nodes close to the sink. These nodes get their energies depleted quicker because they are more active in routing than those further from the sink. This can result in segmentation of the network or a sinkhole. Mobility can take care of this by reshuffling nodes near the sink. Secondly, since for some reasons energy depletion can be uneven amongst nodes causing some to die prematurely, mobility can prevent this by ensuring uniform energy depletion amongst nodes. Furthermore, the sink’s energy efficiency is improved whenever it moves closer to source nodes, as it moves. Nevertheless, node mobility in WSNs is quite costly in terms of energy. It is thus normal to limit it to just a few nodes.

2.3.2 Duty Cycling

At the heart of energy conservation mechanisms in WSN is duty cycling (Carrano et al., 2014), the panacea for energy efficiency in WSN. This is the periodic or aperiodic turning *on* (active) and *off* (inactive) of a system (Anandababu et al., 2016). Once *off*, a system consumes no energy. The duty cycle of a node is the percentage of time it is *on*. *Off* can sometimes be the low energy sleep mode. The longer a node sleeps, the lower is its duty cycle and the more energy it saves leading to a longer lifetime. In some systems the inactive period translates to completely putting it *off*, with this period being quite long and sandwiched between short active periods. This situation is referred to as disconnected modes and can bring the duty cycle to as low as 0.01% (Bober and Bleakley, 2014). While this is quite good, it is very difficult to achieve due to the fundamental problem of inter-node clock skew arising from nodes being *off* for long periods of time.

Duty cycling can be applied on any subsystem of a sensor node, but in this dissertation, we concentrate on the radio or communication subsystem because it is one of the biggest consumption centers of the system. Radios typically can operate in four power modes; transmit, receive, idle (idle listening), and sleep. These correspond to the activity of the radio at different states with idle or idle listening corresponding to a state where a node listens without receiving. Typically, the power consumed in the idle, receive and transmit mode are comparable in order of magnitude but the sleep mode consumption is several orders of magnitude lower. Mica2 Mote sensors for example, have power levels as shown in Table I.

Radio State	Power consumption (mW)
Transmit	81
Receive/Idle	30
Sleep	0.003

Table 1: Radio operation modes (Alfayez et al., 2015)

While any of the first three modes can be referred to as the *on* mode, our focus is on the idle mode as this is predominant in terms of time. The sleep mode is the *off* mode. To make substantial gain in energy, nodes exploit this mode property as much as possible. The radio is made to sleep (*off*) for as long as possible, waking up only when it expects to enter into communication. This is done in a cycle since it is impossible to predict when there will be need for communication. This *on* and *off* cycle of the radio is what we referred to as its duty cycle.

Duty-cycling is favored by two common phenomena in WSN operations. Firstly, it exploits the fact that nodes are typically deployed in a random and redundant manner. This can result in a single area or spot being covered by multiple sensors. The system can thus select only a minimal subset of nodes to remain active for maintaining connectivity and the rest put to sleep. This is topology management. Secondly, it exploits the fact that in most applications, event occurrences are rare. Nodes spend most of their time idle, doing nothing else but listening to the channel in a hope to receive, consuming energy in the process. This means that even the selected subset of active nodes, do not have to be always active. This is power management. Furthermore, duty cycling and energy efficiency are consistent with a battery property, where its lifetime doubles if used in short bursts separated by significant off time than in a continuous mode of operation (Panag and Dhillon, 2015).

Many protocols have been proposed on duty cycling. The scheduling of sleep or awake time amongst nodes is the main distinguishing factor between these protocols. That is, duty cycle protocols are classified according to the flexibility of sleep or awake time. Here, we can have the following two classes:

- Fixed (static) duty cycle
- Variable (dynamic) duty cycle

In fixed duty cycle protocols, the duty cycle of all nodes in the network is fixed at some value determined from some network parameters. Though the scheduling and duty cycle value within groups of nodes may be different, there is generally less overhead as the duty cycle(s) is/are programmed into nodes upon deployment. Nevertheless, there is need for time synchronization within and/or between groups of nodes, which consumes bandwidth, storage and even energy. An example here includes Sensor-MAC (SMAC) (Khatarkar and Kamble, 2013).

Still within the fixed class we can have staggered (various) duty cycle protocols. Staggered duty cycle involves cases where sets of nodes in the network have different duty cycle values according to their role or position in the network with the cycles overlapping to ensure communication between sets. We see this in many protocols, where, as the responsibilities of sets of nodes increase with their location approaching the sink, so does their duty cycles. This is because they need to route information from many more sources as their location approach the sink (Raman, 2002). Staggered Routing-Enhanced MAC (SRMAC) (Liu and Yao, 2011) and self-learning MAC (L-MAC) (Dinh et al., 2016) are examples in this class.

A number of protocols or applications exist for which the duty cycle of nodes is adjustable during execution in accordance with their residual energy or some other parameters. Protocols that implement these are referred to as dynamic duty cycle protocols. We see this in a protocol like Adaptive Staggered sLeep Protocol, ASLEEP (Anastasi et al., 2009a) and Adaptive Duty cycle MAC, ADMAC (Anandababu et al., 2016). Sometimes a network can start out with various duty cycles but end up with all converging to one.

Duty cycle protocols can also be classified into either synchronous or asynchronous classes. In a synchronous protocol, the sleep time of groups of nodes are synchronized (Chelouah et al., 2018). That is, they wake-up and sleep at the same time. This means that there must be a means of maintaining sleep time synchronization between nodes. This synchronization messaging adds overheads in bandwidth, processing time and memory to the system. Synchronization may not only be within groups but with the whole network. The later situation adds complexities to the network especially with increase in network size.

For asynchronous protocols, each node decides on its own sleep time. No synchronization of sleep time is necessary between nodes and thus no overheads in terms of synchronization messaging (bandwidth), processing and memory. Nevertheless, there must be a means of ensuring an overlap between awake times of communicating nodes in order to ensure that messages sent are received within the shortest possible time fame. A node wanting to send a message first has to listen to the channel to check if it is free. If the channel is free, it sends a preamble that contents the destination address. All nodes usually frequently sample the channel for preambles (short or long) (Sha et al., 2013). This procedure is sometime referred to as low power listening (LPL). If no preamble exists, a checking node returns to sleep immediately. Otherwise it waits to see if it is the intended recipient. If it is, it waits to receive the data. If not, it returns to sleep having waited for nothing and wasted energy.

There are also protocols where duty cycle is random. In (Aby et al., 2014), for example, the sleep is random and can cause changes in topology. The paper proposes a centralized and decentralized means for determining route availability.

We will classify duty cycling protocol a bit more detailly in Chapter 3. We now conclude our discussion on duty cycling in this chapter by strongly reiterating the following terms used in the duty cycle literature:

- **Duty cycle schedule**: Also referred to as wakeup or sleep schedule, this is the time (instance) a node either goes to sleep or wakes up. This usually happens in a cycle.

The sleep and wakeup durations depend on many factors like the number of packets to be accommodated during the wakeup period. Different nodes in the same network may have same or different schedules (Valera et al., 2014).

- **Synchronization**: While this generally means multiple nodes sleeping and waking up at the same time and made possible by synchronization messaging, the granularity of the process makes us try to distinguish between sleep and clock synchronization. Sleep synchronization refers to the broad timing when nodes sleep and wakeup at the same time. The nodes have the same duty cycle and schedule. The fine grain clock or time synchronization refers to matching the different clocks of the different devices, so they meet up with the sleep schedules. Sleep synchronization alone may lead to little mismatches in the sleep time of different nodes which can be corrected by regular synchronization messages.

2.4 Routing in Wireless Sensor Network

A strong relationship exists between routing and topology management, including its contributions to energy efficiency. Nevertheless, because we are not working on routing proper, we are only going to have a brief discussion on routing in this dissertation.

In a wireless network, it could be possible to send data from a source to a destination, irrespective of their distance apart if certain conditions are met. These may entail for example getting the transmission power to the right level and/or passing through relay stations. But getting transmission power to the right level may involve expending a lot of energy if the distance is long. This is so because energy expenditure grows exponentially with distance. For this same reason, direct communication between a source and a destination may not always be energetically feasible in WSN. It might be more energetically advantageous to transmit to short distances, even if it means multiple transmission hops despite the delays that this may introduce. Short hops entail dealing with routing issues especially where many paths exist.

As we have seen, nodes in WSN have limited energy storage capacities (Boukerche and Sun, 2018). This further militates towards the use of intermediate nodes to relay packets to the destination. It now becomes a matter of choosing the appropriate intermediate nodes to act as relay nodes. This is routing which generally means choosing a path to the destination (Chaudhary et al., 2014) and is accomplished by routing protocols. A routing protocol is thus responsible for discovering and maintaining energy efficient routes for reliable and efficient communication (Singh and Sharma, 2015). The following approaches may be applied in order to select a route:

- Nodes with maximum available energy
- Route requiring minimum energy
- Routes with minimum number of hops
- Route having minimum number of nodes on critical energy level
- Etc.

Generally, routing in WSN can be classified in three classes as being data-centric, hierarchical or location-based. Data-centric routing also referred to as flat routing routing or query-based routing is done according to the type or amount of data to be sent or received. Sensor Protocols for Information via Negotiation (SPIN) is an example protocol in this class (Kulik et al., 2002). Directed Diffusion also belongs to this class (Akkaya and Younis, 2005). Here, the sink sends request using naming schemes on the data it wishes to receive. The interest has gradient field which is a reply link to the neighbor it received from. The interest field also holds information like data rate, duration and expiration of the interest. With this information, paths are established, used by all and updated by the sink.

Rumor routing is variation of Directed Diffusion (Akkaya and Younis, 2005). Here instead of flooding the network, queries are routed to the nodes that have observed events and thus have the required data. Each node has an agent that goes about the network, informing of an event its parent has observed.

Hierarchical routing is one in which nodes arrange themselves hierarchically based on either their energy, position, number of neighbors, etc. (Parvin and Rahim, 2008). Nodes constitute themselves into clusters with one selected as a cluster head. Communication is not allowed between nodes. Nodes can only communicate with the cluster heads which then transmit the data to the sink. Popular protocols here include Low-energy adaptive clustering hierarchy, LEACH (Heinzelman et al., 2002) and Power-Efficient Gathering in Sensor Information Systems, PEGASIS (Lindsey and Raghavendra, 2002).

Location-based or geographic routing uses node location instead of its identification (Singh et al., 2010). Any node within a location can be selected to receive or process data. This means that nodes need to know their locations. This can be done either by having GPS or some distributed localization scheme based on received signal strength, arrival time or manual registration. Diffusing request only to where data is available takes care of multiple transmissions. This kind of routing is appropriate for systems with mobility. Geographic and Energy Aware Routing protocol (GEAR) is an example protocol in this class (Jiang et al., 2006).

2.5 Conclusion

In this chapter, we have reviewed some background information on the main subjects being investigated in this dissertation. This include energy efficiency and duty cycle. Owing to the fact that duty cycle is at the core of this dissertation, we will go further in Chapter 3, to describe the most common duty cycle protocols and assign yet some more into layers of the WSN protocol suit according the task they seek to accomplish.

Chapter 3

Duty Cycling Protocols in the WSN Model

We have discussed duty cycle in Chapter 2. Its protocols are applied in the different layers of the protocol suit to solve different network tasks, energy efficiently. In many protocols, the layer upon which a duty cycling protocol operates is obvious, sometimes even from the name of the protocol as in most MAC protocols. For others, this information can only be obtained by carefully examining the task(s) to be accomplished. In this chapter, we classify duty cycle protocols according to layers of the WSN model by carefully examining the task(s) to be carried out by the protocol. We take a layer, examine its responsibilities, and see how a duty cycle protocol is used to tackle them. We then associate the protocol to the corresponding layer. By these, it will become clear where our proposals, which also lean on duty cycle, are associated.

While some tasks cannot be easily tied to one layer, some protocols too are so designed that they take care of tasks that span multiple layers. Such are referred to as cross-layer protocols. In this chapter we will concentrate on protocols on the data link, network and transport layers upon which most duty cycle protocols have been proposed. A summary of this classification is presented in Table 2.

Data link layer	Network layer	Transport layer	Cross layer
SMAC, T-MAC, RMAC , B-MAC, X-MAC, L-MAC, BMA, WiseMAC, RI-MAC	GAF, GeRaF, Span, STEM, FPS	CCDC, ADCC, DCA	P-MAC, SOTP

Table 2: Example of duty cycle protocols according to layers

3.1 Data link Protocols

The responsibilities of the data link layer and especially the Media Access Control (MAC) sub-layer, are well known. But here, we are interested in protocols that use duty cycle to tackle energy wastage associated with accessing the shared medium and ensuring data delivery. Typical losses associated to this layer for which duty cycle is call upon to mitigate include (Samant and Datta, 2016):

- Idle listening: Listening to receive packets that are not sent
- Retransmission: When a sender needs to resend data that is lost during transmission.
- Control packet: Packets that are sent for control purposes, not being actual data.
- Over emitting: This is sending information to a destination that is not listening.
- Overhearing: This is receiving data not destined to the node in question.

Generally, MAC mechanisms can be classified as either contentious or non-contentious. The non-contentious Time Division Multiple Access (TDMA) (Anitha and Usha, 2013) schemes naturally favor duty cycle as channel access is done on a slot-by-slot basis. Nodes turn on their radios only during their slots. Energy consumption is thus reduced only to the minimum required for the transmission or reception of data during a slot. TDMA can be broken down into centralized and distributed. In the centralized TDMA, time slot scheduling is done by a central entity. The main disadvantages include control message overheads and the inflexibility associated with topology changes (adding new or removing dead nodes). Example protocols here include BMA (Jing and Lazarou, 2004), Mobility tolerant TDMA MAC (Thaskani et al., 2011), etc. For the distributed version, each node chooses its slot based on some criteria set in the network. Here control messages are few and the network is quite scalable. Example protocols here include DNIB (Slama et al., 2008) and DRAND (Rhee et al., 2006).

Contention-based MAC protocols are the most popular class of MAC protocols for wireless sensor networks. They too follow duty cycling by tightly integrating channel access functionalities with sleep/wakeup schemes coordinated by carrier sensing, where a node will determine whether to stay *on* or go *off* based on the outcome of carrier sensing.

Finally, hybrid protocols adapt to the level of contention in the network. Example protocols here include hybrid medium access control protocol (H-MAC) (Mehta and Kwak 2010) and PQueue-MAC (Wu et al., 2015).

Using duty cycling, media access control (contentious and non-contentious) and other data delivery techniques, we can group protocols in this layer under the following broad classes:

- **Synchronous**: In synchronous MAC protocols, sleep or wake up is synchronized for nodes while in asynchronous protocol, there is no synchronization of sleep or awake time. Example protocols in the synchronous class include Sensor-MAC (S-MAC) (Ye et al., 2002) and Time-out MAC (T-MAC) (van Dam and Langendoen, 2003). As has been seen in Chapter 2, protocols here can be staggered. An example protocol in this include is the staggered Routing-Enhanced MAC (SRMAC) (Liu and Yao, 2011).
- **Asynchronous**: Here there is no synchronization of sleep or awake time. Example protocols here include Berkeley MAC (B-MAC) (Polastre et al., 2004), X-MAC (Buettner et al., 2006), WiseMAC (El-Hoiydi et al., 2003) and the A receiver-initiated asynchronous duty cycle MAC protocol (RI-MAC) (Sun et al., 2008).

These two main classes have already been discussed under energy management. Here we present a brief review of some popular protocols in the synchronous class.

S-MAC is a canonical synchronous MAC protocol for WSN. It is one of the earliest and most popular protocol for WSN. The duty cycle is fixed though its scheduling amongst groups of nodes may be different. During the active (awake) period, there is schedule synchronization, channel contention and then reservation using RTS/CTS (request-to-send/clear-to-send) messages. If a node wakes up and hears no schedule, it determines its own to be followed by others a few hops down. But if it hears another, it follows it. A node may be on the border and hears a second schedule after haven already chosen a first. It chooses this as a second and follows the two.

T-MAC is an improvement of S-MAC in that it introduces early sleep. When a node wakes up and realizes it will not be involved in data transfer in a particular cycle, it goes back to sleep even before its awake time expires, to wake up again the next time it is supposed to wake up. Though this protocol is put here, it can be argued that since nodes can go to early sleep, its duty cycle is adjustable. Though nodes can exhibit early sleep, they still wake up when they are supposed to wake up following synchronization message. Thus, its placement here.

In asynchronous protocols, the main problem is how to ensure that communicating nodes meet each other while awake. One way to resolve this is by deployment in large quantity. By the way a node tries to ensure that its transmission reaches its partner, these protocols can be divided into preamble oriented and random. In preamble oriented, a node tries to ensure that its transmission is received by first sensing the channel before sending a preamble, if it finds it unused. The preamble can be long or short (intermittent) and contains some useful information. This carrier sensing is referred to as low power listening (LPL). All nodes do the same. A node returns to sleep if no preamble exists in the channel for it. If one does, it waits to see if it is the intended recipient. If so, it waits to receive. Otherwise it goes back to sleep, haven waited for nothing and wasted energy. An example in this class is WiseMAC (El-Hoiydi et al., 2003).

Depending on the initiator of the LPL procedure, asynchronous protocols can be transmitter- or receiver initiated. If a node has data to send, after sensing the channel and finding it idle, it transmits an intermittent request, a short preamble or the data packet itself, until this “hits” the listening time of the destination node to announce that it has data for it. On receiving this, the destination waits to receive the data. This is a transmitter-initiated asynchronous protocol with most of the protocols already presented like B-MAC and X-MAC, belonging to this class. The reverse is a more recent paradigm referred to as receiver-initiated asynchronous protocol. Here a receiver expresses interest using the same procedure, to receive from a particular source or locality. An example protocol here is RI-MAC.

In random asynchronous protocols, a node wakes up at random and sends a broadcast message to its neighbors informing of its availability to act as a relay node. The reception of transmission here is purely probabilistic. An example protocol here is Blind MAC (Aby et al., 2014).

We can go further to distinguish asynchronous duty cycling protocols into:

- **Uninterruptible**: A transmitting node stays on until it finishes the transmission of whatever it is transmitting irrespective of whether its wake time is over or not. An example here is uninterruptible asynchronous duty-cycling (U-ADC) (Cheng et al., 2012).
- **Interruptible**: Here a transmitting node stops transmitting once it is time for it to go to sleep irrespective of whether it still has data to transmit or not. An example protocol here is interruptible ADC (I-ADC) (Cheng et al., 2012).

Some popular protocols in this asynchronous class include:

Berkeley MAC (B-MAC): Here, a preamble is transmitted with the data and the transmission time is long enough to be received by the intended receiver. An awake node samples the channel only once a preamble has been detected. The overhearing that accompanies the long preamble is its main disadvantage.

X-MAC: This is the most studied asynchronous MAC protocol that improves on B-MAC. Here each station wakes up and listens to the channel at a different time from its neighbors. This offset time between stations' wakeup is random. Nodes send short preambles which contain the destination addresses. Between the emissions of two consecutive preambles, there is a pause in which the transmitter listens to the channel to know if a station has answered. This fact decreases the overhead because if a station, which is not the receiver, reads the packet, it will go to sleep until the next cycle (as it knows that in this cycle it can neither send nor receive). Any node that wakes up and finds out that it is the intended recipient, sends an early acknowledgement and waits to receive all the data. There is therefore an improvement in latency.

Protocols in the Data Link layer could also be classified as hybrid (Mehta and Kwak, 2010). Here the network is divided into clusters that are synchronized internally, while the communication between the clusters remains asynchronous.

Figure 5 presents a classification with an example each of duty cycling MAC protocols.

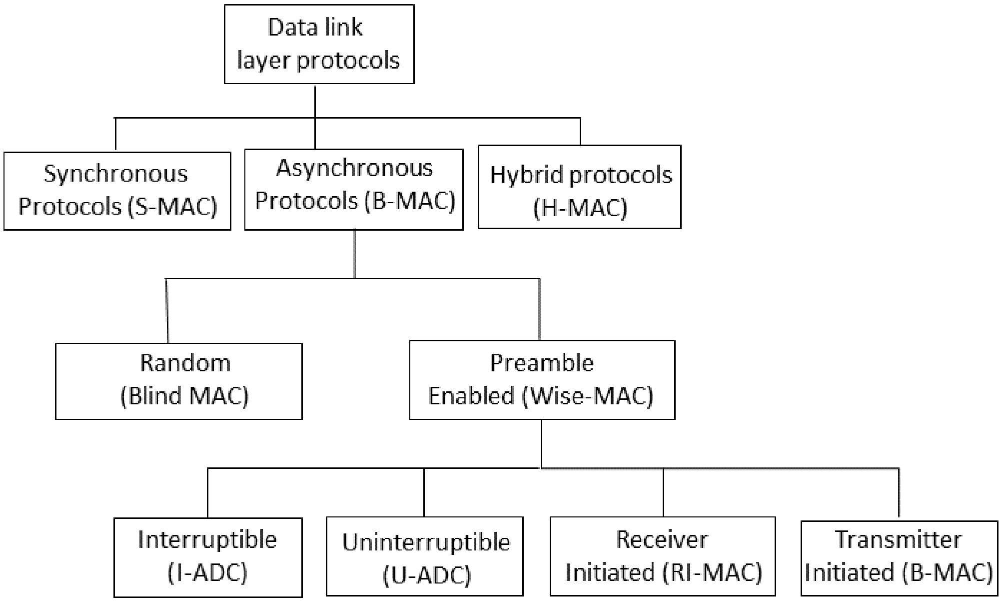


Figure 5: Taxonomy of duty cycling MAC protocols

3.2 Network Layer Protocols

For the network layer, we are looking at protocols that use duty cycles to ensure connectivity. Nodes that are not needed to ensure connectivity go to sleep to save energy. Finding the optimal subset of nodes that guarantee connectivity is referred to as topology management or control. Even then, the active nodes selected by the topology control protocol do not have to maintain their radios on continuously. The radios are switched off (or put in the low-power sleep mode) when there is no network activity, alternating between sleep and wakeup periods. This situation where the active nodes from topology management are duty cycled, is referred to as power management (Anastasi et al., 2009b). Topology control and power management are thus complementary network layer techniques that implement duty cycling with different level of granularity.

Several criteria can be used to decide which nodes to select for activation and when. In this regard, topology control and power management protocols can be broadly classified into the following two categories: (1) location driven and (2) connectivity driven protocols (see Figure 6). The former decides on nodes to keep awake as routing nodes, based on their location assumed to be known in relation to the location of the activity to be sensed. Location-driven protocols distinctly require that nodes recognize their positions. This is generally achieved by providing them with some location detection mechanism like GPS units. This is seen in geographic adaptive fidelity (GAF) (Roychowdhury and Patra, 2010) protocol where nodes within a given location form a virtual grid of equivalent nodes and then decides which wakes up and when. This role is rotated among nodes within the grid. Another protocol here is Geographic Random Forwarding protocol (GeRaF) (Zorzi and Rao, 2003).

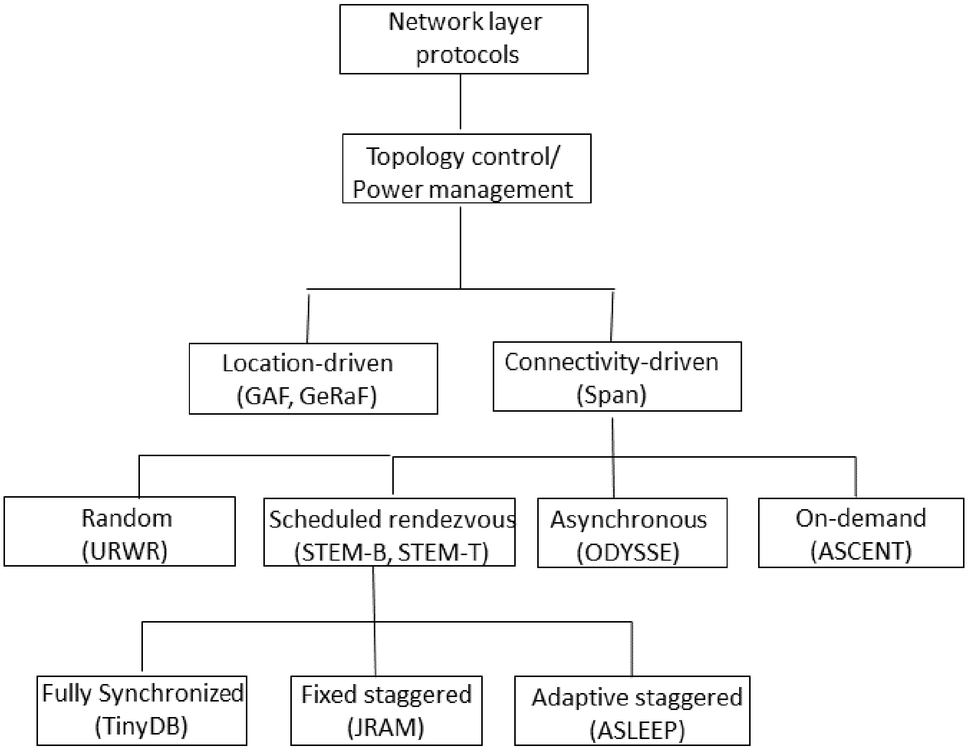


Figure 6: Network layer duty cycle protocols

Connectivity driven protocols dynamically activate/deactivate nodes so that connectivity is ensured. Span (Chen et al., 2002) is an example. It adaptively elects “coordinators” amongst nodes which should be *on* at all times for routing purposes. Other nodes periodically wake up to check if they can become coordinators while the coordinators also check for messages from potential coordinators, in order that they can step-down for replacements.

Establishing connectivity includes route selection, and this provides another sub-criterion for classifying duty cycle protocols in this layer. Thus, we can have (Anastasi et al. 2009b):

- On-demand
- Asynchronous
- Random
- Scheduled rendezvous

In an on-demand protocol, a node is awakened only whenever there is need to engage in communication. This approach is the most profitable when it comes to power management, with a node waking up only when another wants to communicate with it. The main problem with this scheme is how to inform a sleeping node that some other node wants to communicate with it. To this end, such schemes typically use multiple radios with different energy/performance tradeoffs, one for wakeup signaling and the other for data transmission. It is important to know which radio is suitable for what (Stathopoulos et al., 2007). The lower data rate, lower frequency and lower power radio is usually more suitable for signaling while the higher data rate, higher frequency and higher-power radio is suitable for data communication. While these protocols are very efficient in terms of energy, sensor nodes with multiple radios are not common.

Sparse Topology and Energy Management (STEM) protocol uses two different radios, one for wakeup signaling and the other for data transmissions (Schurgers et al., 2002). There are various variants of these, mainly based on how nodes are awakened. The original version uses a beacon within the signaling channel and is termed STEM-B. In another variant referred to as STEM-T, a wakeup tone is used instead. The main difference is that in STEM-T all nodes in the neighborhood of the initiator are awakened by the tone.

A somewhat different approach for on-demand scheme is presented in Flexible Power Scheduling (FPS) protocol (Hohlt et al., 2005). It uses TDMA scheme that involves an on-demand slot reservation mechanism, allowing nodes to reserve slots in advance.

In asynchronous routing, nodes wake up at their own times to take part in communication. This means that a node may have to wait for a sleeping node to wake up in order for it to route data to/through it. This can lead to high levels of latency in the system. A number of mechanisms exist to take care of this in different protocols. For example, a sender may have to predict when a relay node will come awake. An example of a protocol here is

Opportunistic Duty cYcle based routing protocol for wirelesS Sensor nEtworks (ODYSSE) (Amdouni et al., 2016).

There are routing protocols built on the premise that duty cycle is random. Here, senders bank on purely on probability in order to catch the relay nodes or destinations. An example protocol here is Unicast Random Walk based Routing (URWR) (Mabrouki et al., 2007). There is the also the scheduled rendezvous approach. Here, nodes in a neighborhood wakeup and sleep at the same time. This guaranties that once a node is awake following a predefined schedule, some of its neighbors are also awake, and they can communicate. The awake time can be kept quite short, enough to permit data transfer while still meeting its neighbors awake. The main disadvantage is the requirement for synchronization of their awake time. There might also be need for strict clock synchronization among node in this network. A protocol is this class is delay-aware energy-optimized flooding algorithm (DEF) (Wu et al. 2016).

These protocols (scheduled rendezvous) are implemented in different ways in accordance with the time groups of nodes exercise their duty cycles. The fully synchronized approach is the simplest and easiest implementation (Mihaylov et al., 2011). Here, all nodes in the network follow the same schedule. Because of these, it is the most popular implementation, practiced in TinyDB (Anastasi et al., 2009b) for example. It enables low duty cycles, where the active time is significantly smaller than the sleep time. When the active time is large, strict clock synchronization might not be necessary. Because all nodes become active at the same time, multiple access issues may occur.

Furthermore, a network can be structured into a tree topology with a base station typically rooted at the sink (Anastasi et al., 2009b). Duty cycling schemes take advantage of this to size the active time of different nodes according to their position in the tree. This is the staggered wakeup scheme and the duty cycle of different nodes are fixed, if nodes are static. The active time of nodes belonging to adjacent levels must be partially overlapped in order for them to be able to communicate. This scheme has several advantages. Because nodes at different levels wake up at different times, only a subset of nodes in the network will be active at the same time, lowering the number of collisions. It is also suitable for data aggregation or filter. The scheme also has some drawbacks related to limited flexibility due to the fixed duration of the active periods. Example protocol here include Pipelined-forwarding, Routing-integrated, and effectively-Identifying MAC (PRI-MAC) (Tong et al., 2013) and joint routing and medium access control (MAC) protocol (JRAM) (Singh et al., 2017).

It is possible in these networks to have the active periods quite low. This does not only help is saving energy but also in minimizing the latency of packet transmission. More so, it is possible for nodes at different branches of the tree to manage different amounts of data depending on their location, network traffic, node mobility, etc. Active periods of the different groups can be optimized accordingly. This point is accentuated where there are frequent topology changes and variations in the traffic patterns that are common in WSNs. Several protocols take advantage of these and permits various groups of nodes to dynamically adapt their duty cycles like in the one proposed by. By being able to set the length of the active period to the minimum value in accordance with current network activities, this adaptive scheme performs better in terms of energy efficiency and average packet latency than its fixed staggered counterparts. In addition, this whole scheme reduces the number of collisions. An example protocol here in An adaptive sleep protocol (ASLEEP) (Anastasi et al., 2009a).

3.3 Transport Layer Protocols

One of the responsibilities of the transport layer is flow control. Many issues can lead to congestion in WSNs such as buffer overflow, varying transmission rates, convergecasting (many-to-one flow), channel contention and interference. Congestion can lead to node energy depletion, deterioration of various network performance parameters and packet losses. Duty cycle protocols that take care of this and other transport layer responsibilities are less numerous than for other layers. However, here we present a protocol that aims at mitigating congestion, while being energy efficiency.

Congestion Control Technique for Duty Cycling WSN (CCDC) (Jang and Yoo, 2017): For a high traffic network, duty cycling can increase congestion which increases packet loss and delay. In this protocol, each node checks its queue size against a threshold for congestion control. Congested nodes inform their parent (upstream) nodes to adjust their duty cycle values accordingly.

Other protocols in this area include Adaptive Duty-cycle Based Congestion Control (ADCC) (Lee and Chung, 2010a) and Duty-cycle Based Congestion Avoidance (DCA) (Lee and Chung, 2010b).

3.4 Cross-Layer Protocols

Looking closely at the responsibilities at the MAC and Network layer, it becomes very difficult in terms of duty cycles, to draw a clear distinction between these two layers. Apart from its main responsibilities of energy efficient channel access, MAC duty cycle protocols inherently lead to reduction of end-to-end delivery latency, a routing or network layer responsibility. This makes it quite difficult to separate protocols between these two layers. There are two possible ways to develop a protocol. A protocol can be built independently of any routing protocol. That is, no particular routing protocol is specified, in which case packets are simply flooded or gossiped into the network until they reach their destinations (Akkaya and Younis, 2005). Directed Diffusion can also be used. A protocol that takes care of multiple access without incorporating any routing protocol (implying they use flooding, gossiping or directed diffusion) is considered a MAC protocol. That is, they are built without any consideration of routing.

Protocols can also be designed to operate with a network or routing protocol. A protocol that takes care of both MAC and end-to-end delivery, incorporating a routing protocol, is cross-layer. Cross-layer protocols like Pipeline MAC (P-MAC) (Tong et al., 2011), and self-organized TDMA protocol (SOTP) (Wang et al., 2006), use information emanating from the network layer. Thus, it is truly a cross-layer protocol.

P-MAC uses the concept of staggering duty cycle described in Section 3.2 to arrange nodes into grades according to their distances (hop) from the sink, with nodes in the same grade having the same sleep/wakeup schedule. The sleep/wakeup schedule of nodes in adjacent grades are overlapping. Thus, a pipeline is formed for the transmission of data to the sink.

On its part, SOTP used the MAC scheme of TDMA in combination with routing layer requirements to allocate time slots to nodes in a centralized manner.

3.5 Conclusion

In this chapter, we have made a link between the task(s) to be accomplished by duty cycling protocols and the WSN protocol suit. This is because the relationship is not obvious at just a glance and can only be done by a careful review of the task they seek to accomplish while being energy efficient. It can clearly be seen that more work in this area has been done for the MAC sublayer and to a lesser extent the network layer than the others. For this reason, our classification has focused more on these layers.

Having looked at duty cycling, it is now time for us to see how to tap on its advantages in the exploitation of the WSN. We start by building, in Chapter 4, the topologies upon which this will be applied.

Chapter 4

Set Formation Algorithms

In this chapter we propose several topology management algorithms for wireless sensor networks that arrange nodes in the network into different logical set formations. The goal is so that the resulting sets be used for the implementation of other protocols/algorithms being designed in Chapters 5 and 6 of this work. We will simulate some of the set formations to demonstrate the effectiveness of the algorithms.

4.1 Introduction

In a wireless sensor network, topology management can involve arranging nodes or nodes arranging themselves physically or logically into particular formations. In our case, we are concerned with logical formation. Topology management (logical) taps on some of the many advantages of redundant deployment of nodes, a common practice in WSN (Slijepcevic and Potkonjak, 2001). The main goal of most topology management schemes is to save or further save energy by the application of duty cycling. Energy consumption of the network can be saved through optimization of the network topology. Nodes logically arrange themselves in such a way that while duty cycling for energy efficiency, total network connectivity is maintained. That is, while sets of nodes are sleeping, at least one set stays awake to ensure connectivity (and/or coverage). The emphasis here is on auto-configuration, i.e. nodes arranging themselves by themselves (Slijepcevic and Potkonjak, 2001). Duty cycling has been presented in Chapters 2 and 3.

Topology management is very closely related to network deployment. The issue of network deployment (Sharma et al., 2016) is quite critical for wireless sensor networks as it has direct impact on energy efficiency, routing, coverage optimization, transmission coverage, etc. Wireless sensor network deployment is the establishment of an operational wireless sensor network in real life (Ringwald and Romer 2007). For mobile wireless sensor network, deployment is not very critical as the mobility of nodes can solve many of the problems that deployment brings. A problem that can result from deployment is uneven distribution of nodes in the field. This can be solved by nodes automatically moving from congested area to less congested ones. This technique can also be applied in cases where unevenness is caused by death of nodes or addition of new ones. But, as has been said in Chapter 2, mobility is quite expensive in terms of energy.

Deployment is usually done either according to the activity, the surface area or even the topographical constraints in the area to be monitored. These can result in the network having different forms. Thus, we can have wireless sensor network topologies being linear, in grids, random, etc. In this chapter we consider a network where static nodes are deployed in a random and redundant manner between an epicenter and a sink. An epicenter is a location where the probability of event occurrence is highest while a sink or base station is where information is gathered or processed. By redundancy, we are talking about more than one node covering/sensing the same area/event. In this kind of random deployment, nodes are left to organize themselves into a functional network. Although (as has earlier been seen in Chapter 2) redundancy improves reliability and robustness to failures, the cost of nodes can make it expensive. Nodes can also be deployed deterministically at pre-defined locations, but we will not look in this direction in this dissertation.

Duty cycling has been shown to be applied on a broad scale of topology control or management, where some nodes (or their radios) in the network are put to sleep, leaving only a selected few to be active to take care of connectivity (and/or coverage). This active state can be rotatory amongst nodes, leading to the term **activity scheduling**. This is the topology management option of rotating the active state of sets of nodes amongst themselves. Topology control can also take care of the data link issue of media access control (MAC), where the number of nodes trying to access the media is reduced, leading to further savings in energy due from retransmission that results from collisions. We can therefore describe topology management or control to simply mean the logical organization of deployed nodes in a particular manner, such that certain network performance metrics are met or improved. In most cases and in our study, the metric is the maintenance of total connectivity and coverage. By these terms (coverage and connectivity), we are referring to the ability of the network to maintain strict surveillance of the field of interest and gather the desired information and to forward the sensed data to where it is required (Boukerche and Sun,2018), respectively. This brings about huge savings in energy as nodes not required at the moment to ensure either connectivity or coverage are put to sleep. A finer grain power management scheme occurs when even the few active nodes resulting from topology control are duty cycled.

In this chapter, we propose topology management algorithms where nodes arrange themselves into concentric sets around one node (most probably, the sink). These algorithms are grouped into two broad groups. In the first group, sets are formed in such a way that members of the same set, though extended throughout the network, ensure total connectivity in the network. Members of a set form a path between any source and the sink. They are interleaved with members of other sets. We call this *repetitive* set formation algorithms. In the second group, sets are formed such that successive sets are within transmission range of each other. Here too, there exists a path between any source and the sink node but passing through different sets. We refer to this group as *continuous* set formation algorithms. In both cases, total connectivity (and coverage) of the network is ensured.

We intend to use these proposals to show how topology management can further improve network performance in terms of energy, latency and packet delivery ratio. We will also use some of our algorithms to construct a set of disjoint virtual backbone networks in Chapter 5.

4.2 Review of Topology Management Schemes

The objective of these protocols may not directly suggest topology management, but topology management is obvious in their implementations. Examples groups of protocols here include mostly clustering and staggering sleep protocols.

In this section we look at some protocols that apply topology management even if this feature is not their main goal or is not explicit. That is, although they may be propagating some other phenomenon, topology management is applied in the implementation of the protocols. In as much as deployed nodes arrange themselves in a particular order to meet a certain goal, it is topology management. In clustering for example, the auto-formation of clusters and the selection of cluster heads is topology management. The same can be said of staggered sleep management schemes where nodes arrange themselves into a tree structure, with a root on one node (Tong et al., 2013, Singh et al., 2017). As has been seen in Chapter 3, the staggered sleep might be fixed or dynamic. We have also seen in Section 3.2, the geographic adaptive fidelity (GAF) (Roychowdhury and Patra, 2010) protocol where nodes within a given location arrange themselves into grids of equivalent nodes and then decides which wakes up when, with this role rotating among nodes within grids.

Bober and Bleakley (2014) propose a staggered wake-up scheme at the start of duty cycle wake-up periods to take care of clock skews and as a means of synchronizing nodes' sleep cycle. Liu and Yao (2011) present a staggered routing enhanced MAC protocol for wireless sensor networks, where the duty cycles of groups of nodes are staggered in accordance with their location in the network. Nodes arrange themselves in terms of routing, in a tree manner, with their root at the sink. This takes care of many network performance issues like delay and packet delivery ratio. Anastasi et al. (2009) propose ASLEEP protocol. This is an adaptive staggered sleep protocol where the sleep cycles of nodes are firstly staggered, that is, different nodes in adjacent tree levels having their different sleep times overlapping. Secondly, the duty cycles adapt to different network conditions like nodes' positions, network congestion, etc.

In clustering, neighboring nodes group themselves into clusters, with one chosen as a cluster head (CH). Nodes within clusters communicate with each other following some agreed rules. CH are then responsible for transmitting these data to the sink or between clusters. Thus, CHs have more responsibilities as they collect data from nodes in their clusters, proceed them before forwarding to the sink or other CHs. Xu et al. (2012) present clustering as playing a very important role is saving energy and favoring scalability in WSN. They go further to classify clustering algorithms according to how cluster heads are chosen into the following 3 classes:

1. Heuristic Algorithms: The choice of CH here include node with highest ID or first claim in the absence of any CH. They can also use number of neighbors as a base

for choosing a CH. Example protocol here include Linked Cluster Algorithm (LCA) (Carlos-Mancilla et al., 2016).

2. Hierarchical: This class of algorithms use either the remaining energy or proximity to the sink to choose a node as CH. Example protocols here include LEACH (Verma and Singh, 2018) and HEED (Chand et al., 2014).
3. Overlapping Clustering: Here clusters overlap, with boundary nodes functioning as gateway between CHs. These protocols are robust in terms of energy efficiency and scalability.

Topology changes can be caused by dead nodes, addition of new ones, channel fading or network partitioning (Renold and Chandrakal, 2016). Topology control also involves varying transmission range or duty cycle. The goals of topology control include:

- Prolonging network lifetime
- Limiting congestion
- Improving MAC and routing efficiencies
- Ensuring better coverage and connectivity
- Improving network performance (delay, packet delivery ratio, etc.)

Clustering is usually accomplished in two phases; setup and steady state (Elshrkawey et al., 2018). In the setup phase, there is cluster formation and selection of cluster heads. In the steady state there is transfer of data within and between clusters usually by time division multiple access (TDMA). For LEACH, setup involves random choosing of CH at the beginning of each round. Nodes select a random number between 0 and 1. The node with the least number below a threshold is chosen as the CH. Nevertheless, LEACH has the following disadvantages:

- No consideration for remaining energy of nodes before choosing them as CH
- The disadvantage of synchronous TDMA where a node without data still keeps its time slot making it go wasted, even if others have data. This waste bandwidth.
- Unequal distribution of nodes in clusters
- Network partitioning caused by choosing node with small remaining energy as CH

- Assumes equal energy for all nodes
- Assumes homogeneity of nodes
- Offer of no security

Though LEACH is a reference protocol for clustering, many protocols strive to address its shortcomings, especially the wastage associated with synchronous TDMA, remaining energy, randomness of CH selection, etc.

4.3 Set Formation Algorithms

Since WSN deployment and/or application can cause it to take various physical forms, we assume a dense and randomly deployed network between an epicenter and a sink. We apply our proposed set formation algorithms on these networks. Our algorithms permit nodes to arrange themselves logically into concentric sets such that total connectivity between the epicenter and the sink is ensured.

Before presenting our proposals, we make the following general assumptions:

- Nodes are immobile
- The sink is connected to power source
- Duty cycle is the same for all nodes and is fixed

Our different algorithms will be discussed according to how sets are formed and according to nodes' knowledge of their locations and their ability to vary their transmission ranges or powers. According to the manner in which sets are formed, we have:

1. Repetitive set formation
2. Continuous set formation

According to nodes' knowledge of their location and their ability to vary their transmission ranges or powers, we have:

Case I: Nodes know their locations and that of the sink and cannot vary their transmission ranges or powers

Case II: Nodes do not know their locations and that of the sink but can vary their transmission ranges or powers

Case III: Nodes do not know their locations and only the sink can vary its transmission range.

The above classification is presented in Table 3.

	Repetitive set formation	**Continuous set formation**
Case I	Repetitive Geographic	Continuous Geographic
Case II	Repetitive Broadcast	Continuous Broadcast
Case III	Multi-signal Broadcast	

Table 3: Set formation algorithms

4.3.1 Repetitive Set Formation Algorithms

In these algorithms, sets are formed in a repetitive and interleaving manner. That is, within the transmission range of any node, members of all other sets are found. Thus, there is interleaving of sets within the coverage area such that members of each set form a path between the source and the sink, ensuring total connectivity (and coverage) of the network. This is illustrated in Figure 7 and 8 in different views.

Figure 7: Top view of repetitive set formation for 3 sets

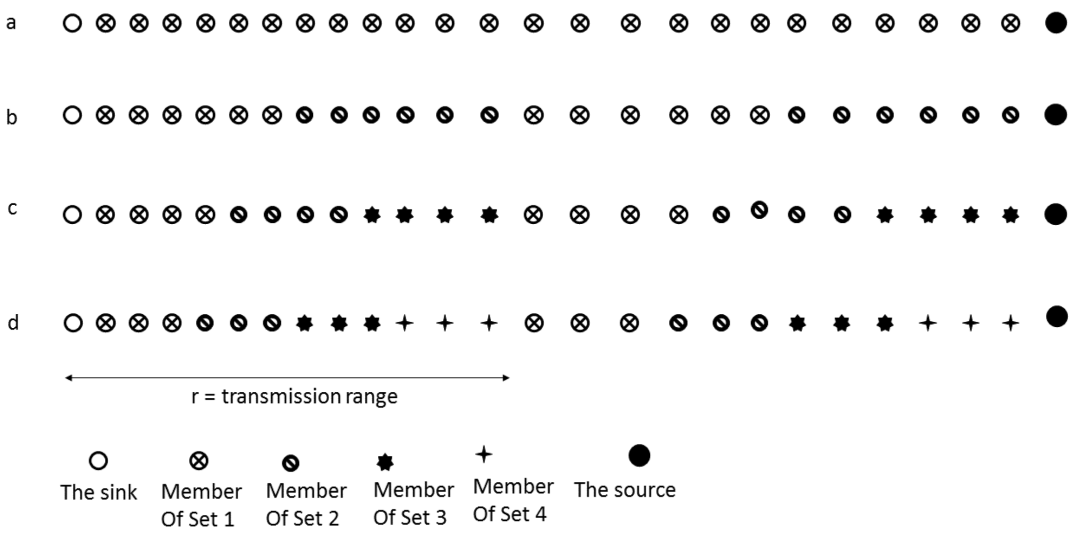


Figure 8: Side view of repetitive set formation

Figure 7 and 8 show different views of set formations. For Figure 8, apart from the source and the sink:

- In (a), all nodes belong to the same set. It is a 1-set configuration
- In (b), the network is divided into 2 sets of nodes
- In (c), the network is divided into 3 sets of nodes
- In (d), the network is divided into 4 sets of nodes

In this Figure, the positions of nodes are the same, irrespective of the number of sets formed. This is to stress the fact that topology management is done logically and not physically.

4.3.1.1 Repetitive Geographic Set Formation Algorithm (Algorithm 1)

In this algorithm, sets are formed in an interleaving manner. It is distributed in nature as each node runs a copy of the same algorithm. However, each node's state depends on its local context. Its name is gotten from two of its properties; nodes' knowledge of their geographical location which they use to calculate the set to which they should belong and the fact that the sets are repetitive.

Using its location and that of the sink, each node independently calculates their distance apart. The node now uses this distance, *D*, the number of sets to be formed, *n*, and its radio transmission range, *r*, to determine the set, *s*, to which it should belong (see Equation 1). We use Euclidean distance to determine the distance, *D (P1, P2)* between a node *P1= (X1, Y1)* and another (the sink) *P2 = (X2, Y2)* as presented in Equation 2:

$$s(Pi) \ = \ \left\lfloor \frac{D(Pi, sink)}{r} \right\rfloor \% \, n \qquad (1)$$

where

$$D \ (P1, P2) = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2} \qquad (2)$$

For example, nodes with a transmission range of 5 units at a distance *D* units from the sink, for 2 and 3 repetitive sets, will belong to sets (for set counting beginning at 0) as presented in Table 4.

D	*s*	
	n=2	*n=3*
3	0	0
12	0	2
17	1	0
22	0	1
26	1	2

Table 4: Membership of nodes according to algorithm 1 (D is distance between a node and the sink, n is the number of sets for range, r = 5)

4.3.1.2 Repetitive Broadcast Set Formation Algorithm (Algorithm 2)

In this algorithm, like with the previous, we aim to form sets in a repetitive manner. Because the set formation is initiated by a broadcast message, we referred to it as Repetitive Broadcast Set Formation Algorithm.

At initiation, the sink, acting as set initiator since it has no constraint in energy, broadcasts a setup message to its neighbors with transmission range, *r/n* (*r* is the maximum radio range while *n* is an integer that represents the number of sets to be formed). Included in this message is the number of sets to be formed *(n),* the set to which it belongs (s = -1) *and* hop count (h= 0). Every other node initially has its variable *s* and *h* set to infinity. Once a node receives a setup message, it compares the *s* received (*s'*) and the *h* received (*h'*) with its own. Once these received values are less that its own (less than or equal for the hop count), it increments these values. For h this becomes its hop count h. For s, it takes the modulus of the increment and the number of sets to be formed n, as its set number, s. The node now updates the setup message with its own values and rebroadcasts it to its neighbors. This procedure continues until all nodes have chosen their sets.

The format of the setup message for this algorithm is presented in Figure 9 while a flowchart is presented in Figure 10.

n	s	h

n = number of sets
s = set number
h = hop count

Figure 9: Format of setup message for algorithm 2.

The fields in this message header are explained thus:

- *n* (number of sets): This is the total number of sets to be formed.
- *s* (set number): This is the number that a node sets as its set number
- *h* (hop count): The number of hops from the sink

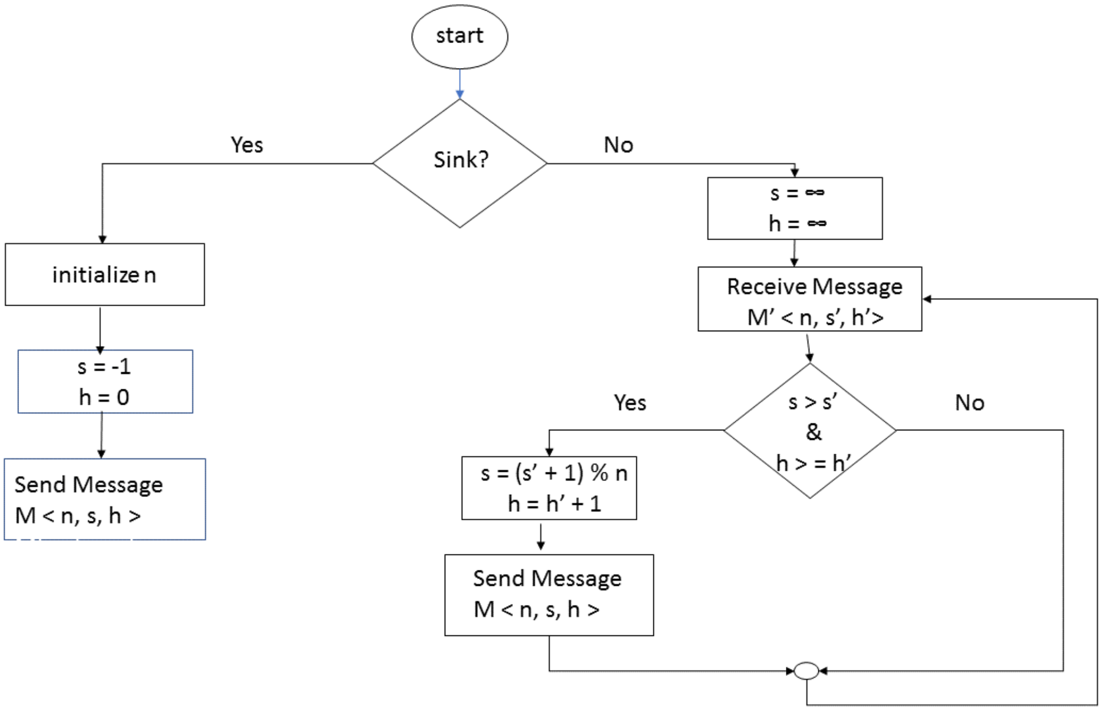


Figure 10: Flowchart for algorithm 2

4.3.1.3 Multi-signal Broadcast Set Formation Algorithm (Algorithm 3)

This is the case where only the sink can vary its transmission range. Its name is coined from the fact that the sink broadcasts *n* messages, where *n* is the number of sets to be formed. Like with the two previous algorithms proposed, it produces repetitive sets.

The procedure starts with the sink broadcasting a number of messages according to the number of sets to be formed. The transmission ranges of these messages are a factor of the number of sets, *n* to be formed. For the formation of two sets ($n = 2$) for example, two messages with transmission ranges $r/2$ and r, respectively are broadcasted. The set number of these messages are also set accordingly (0 and 1 respectively) and hop count fields are set to 0. All other nodes have the values of *h* and *s* set to infinity. Once a node receives a setup message, it compares the received hop count with its own. Once this is lower than its own, it also compares the received set number with its own. Once this too is lower, it adopts both these numbers, updates the message and rebroadcasts it and the process continue.

The setup message format and a flowchart for this algorithm are shown in Figure 11 and 12, respectively.

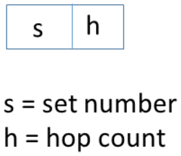


Figure 11: Message format for algorithm 3

The fields of this header are explained as follows:

- *h* (hop count): This is to determine that a progression in distance is made in the network
- *s* (set number): This is the set number to be copied when appropriate

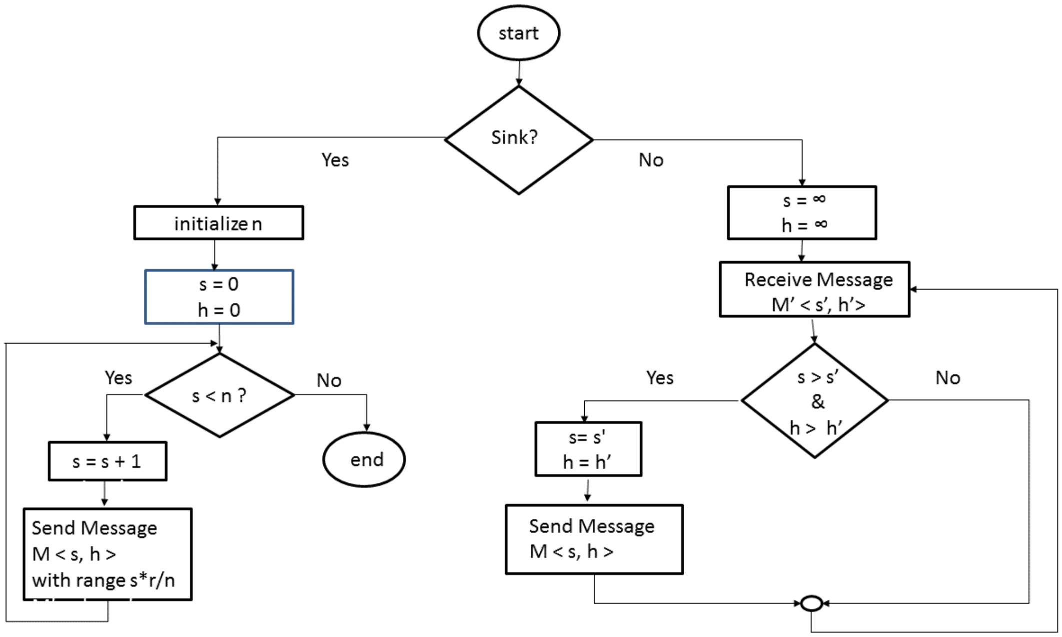


Figure 12: Flowchart for algorithm 3

4.3.2 Continuous Set Formation Algorithms

In these group of algorithms, like with the repetitive ones, nodes arrange themselves into concentric sets around the sink. The main difference is the fact that sets are formed in an ascending order, depending on the coverage area. That is, each set occupies a communication range. Thus, the number of sets is proportional to the coverage area. This formation is presented in different views in Figures 13 and 14.

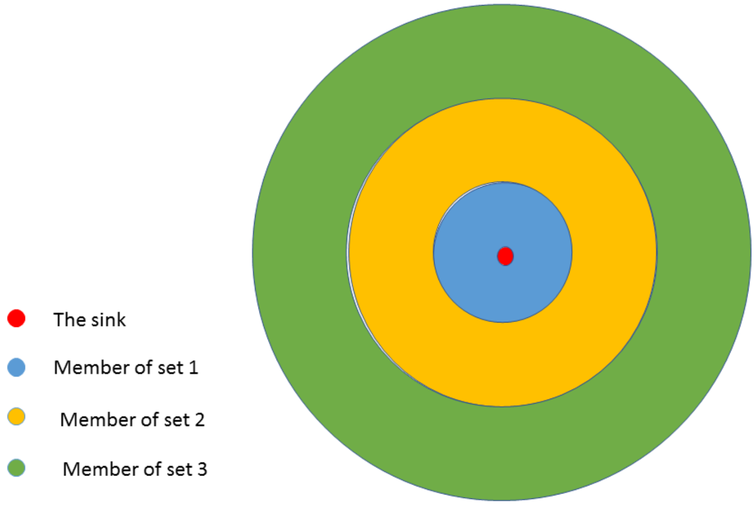


Figure 13: Top view of continuous set formation

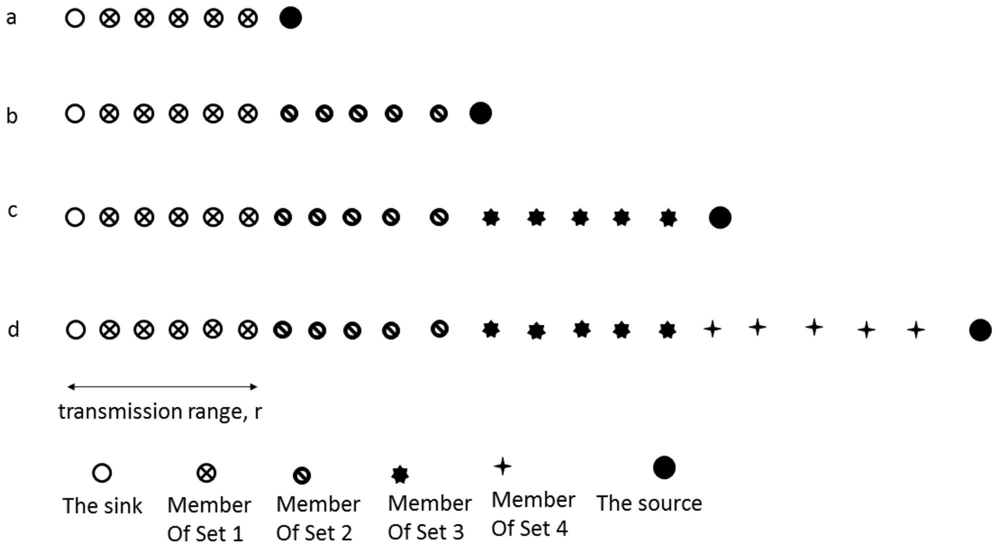


Figure 14: Side view of the continuous set formation

4.3.2.1 Continuous Geographic Set Formation Algorithm (Algorithm 4)

This algorithm is a derivative of the repetitive geographic broadcast algorithm, presented in Section 4.2.1.1, where a node is able to independently and mathematically compute the set to which it should belong using the distance *D*, between itself and the sink, and its transmission range. In this case, it produces sets of nodes in ascending order as opposed to repetitive sets.

To compute the distance *D (Pi, Sink)* between a node *Pi* and the sink, we use Equation (2). To compute the set to which a node belongs, each node *Pi* divides its distance *Di*, from the sink by the transmission range *r* and take the integer part of the answer.

$$s(Pi) = \left\lfloor \frac{D(Pi, sink)}{r} \right\rfloor \qquad (3)$$

This procedure produces concentric circles around the sink. For example, using the same example as is shown for the other algorithms, if we take a node at distance *D* units, and a transmission range of 5 units, it will have *s* as presented in Table 5.

D	*s*
3	0
12	2
17	3
22	4
26	5

Table 5: Membership of nodes according to algorithm 4

It can clearly be seen that this algorithm is distributed in nature, where each node independently determines the set to which it belongs, by running the same algorithm.

4.3.2.2 Continuous Broadcast Set Formation Algorithm (Algorithm 5)

This algorithm starts with a broadcast message to produce non-repetitive sets, thus its name. Sets are formed in ascending order, where a set spans a transmission range. It is a derivative of the repetitive broadcast set formation algorithm, presented in Section 4.31.2.

This algorithm proceeds as follow. After deployment, the sink acting as set initiator broadcasts a setup message to all its neighbors. In this message the sink initializes its set (s) and hop count (h) as -1 and 0 respectively. All other nodes at initialization have both values set to infinity. Any node that receives a setup message compares both its set number and hop count to the ones received. Once these are both larger than the received, it increments both counters, updates the setup message and rebroadcast it and the process continues.

The message format of this algorithm is the same as that of algorithm 3. The flowchart of this algorithm is presented in Figure 15.

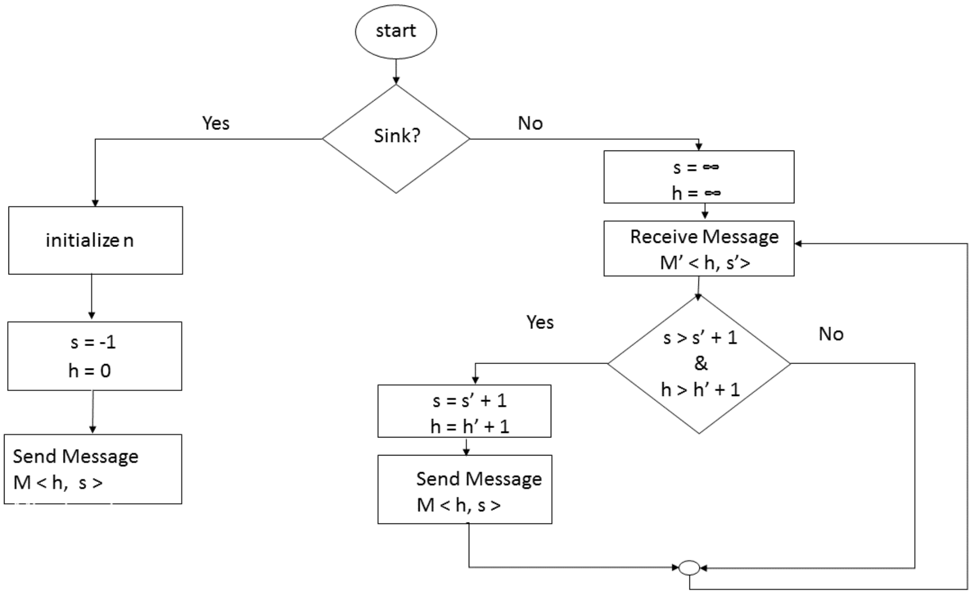


Figure 15: Flowchart for algorithm 5

4.4 Experimental Demonstration

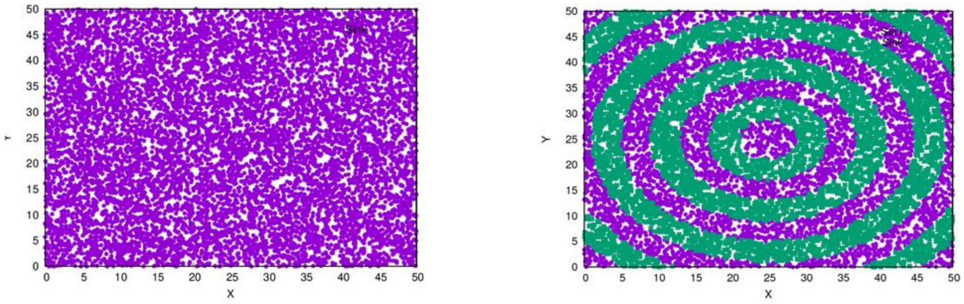
At this point we choose two algorithms, one repetitive and one continuous to run. Firstly, we choose the repetitive geographic set formation algorithm (algorithm 1) to see how the sets will be formed. Then we choose the continuous broadcast set formation algorithm (algorithm 5) to demonstrate the set formation.

4.4.1 Repetitive Geographic Set Formation

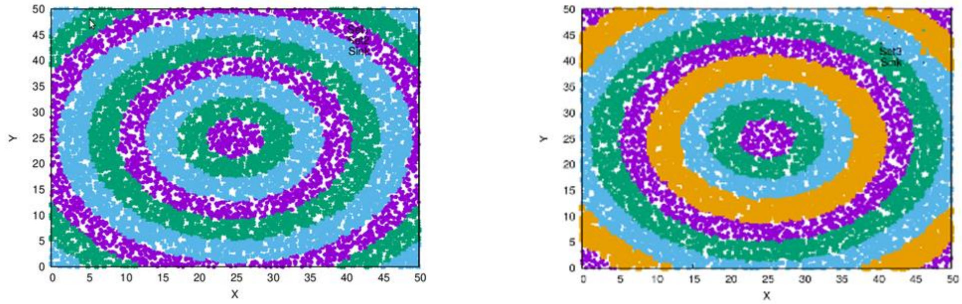
We carried out some experimentations to construct the repetitive geographic sets. The following parameters are used to build our sets that produce the results presented in Figure 16.

- Coverage area: 50 x 50

- Sensor transmission range: 9
- Sink location: (25, 25)
- Deployment: Random



(a) One set (b) Two sets



(c) Three sets (d) Five sets

Figure 16: Top view of repetitive set formation

In this figure, we applied the same color to nodes of the same set. We observe how nodes arrange themselves into repetitive sets around the sinks. In (a) all nodes have the same color. In (b), we see am alternation in just two colors. This same trend is seen with 3 and 4 colors for 3 and 4 sets respectively.

4.4.2 Continuous Broadcast Set Formation

We use ViSiDiA to simulate the continuous broadcast set formation algorithm. This simulator is a framework which enables the implementation of local computations rules (Abdou et al. 2014). The local computation model, which is described by graph relabeling models, defines how nodes change their labels (states) according to their actual states, those of their neighbors and the states of adjacent links.

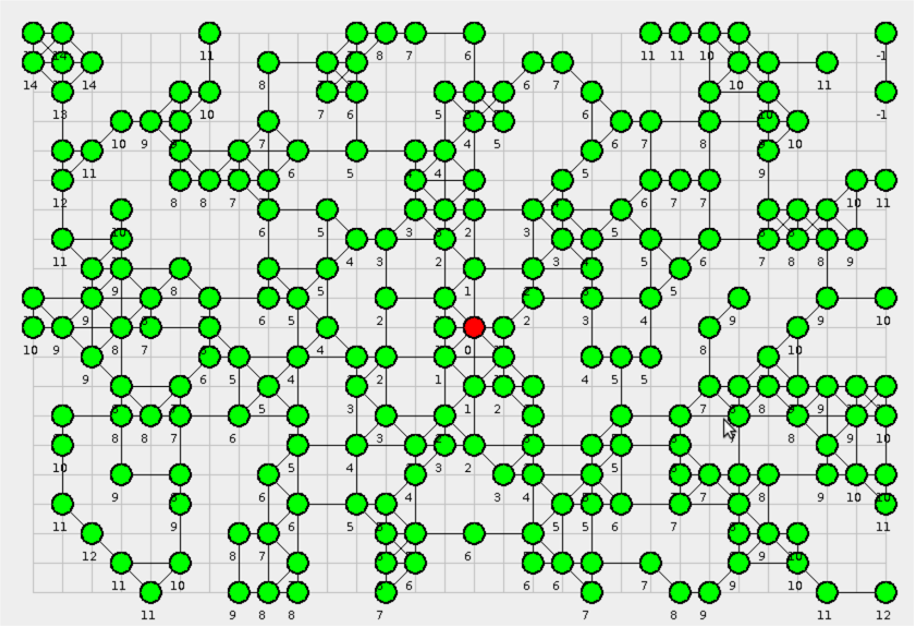
To make easier the visualization of our implementation, we used a synchronous version of distributed algorithms. We assume that at the beginning, every node has its set number initialized to -1, except the sink whose set number is initialized to 0. The sink first sends the setup message which contains its set number. Every node which receive a setup message relays it to all it neighbors at the next round. At each round, every node is eable to know its neighbors' set number.

Let minSet be the minimum set number received from the neighbors (excluding values equal to -1) and s be the current node's set number. We define the following rules:

R1: if s = 0 (the current node is the sink), then initialize the setup procedure (send a message to all the neighbors);
R2: if s = -1 and the current node received a setup message during the previous round, s = minSet + 1
R3: if s != -1 and s > minSet + 1, then s = minSet + 1.

Figure 17 illustrates the results obtained using the following parameters.

Coverage area	1000x1000
Numbers of nodes	236
Sink location	(500, 500)
Deployment	Random
Transmission range	10



3-units range

Figure 17: Screen shot after running continuous set formation algorithm

We can see that nodes in proximity belong to the same set. This changes as we move outwards across sets, as shown by the numbers beside each dot.

4.5 Conclusions

In this chapter we have proposed five algorithms that group nodes into concentric sets in such a manner that total connectivity (and coverage) is ensured. Our goal has been to demonstrate the possibility of set formation using an algorithm from each of the two classes of algorithms. We will move further to build on some of these algorithms to construct disjoint virtual backbone networks with activity scheduling, in Chapter 5. Our ultimate objective is to demonstrate the benefits that set formation can bring to a network.

Chapter 5

Disjoint Virtual Backbone Networks Construction

In this chapter we build on the continuous set formation algorithms developed in Chapter 4 to propose two algorithms for the construction of disjoint virtual backbone networks composed of distinct connected dominating sets. We implement our proposals and evaluate their performances from where we obtain approximation factors. We then compare the two algorithms.

5.1 Introduction

Topology management or control in a WSN can go a long way to enhance network performance. Apart from energy efficiency, topology control can greatly enhance routing, packet delivery and many other network performance issues. Data dissemination, an element of routing tends to be quite complicated and difficult in an infrastructure-less network as WSN (Fu et al., 2015), where the network can be dense and extensive in coverage with multiple routes existing between a source and a destination. Routing is the discovery and maintenance of paths between communicating entities. Since these entities cannot always be within transmission range to each other, some nodes should be able to act as forwarders (Penumalli and Palanichamy, 2015).

Data dissemination in these networks is usually done through flooding which is very burdensome in terms of bandwidth, energy and processing. It brings about what is described as broadcast storm problem (BSP) (Tseng et al., 2002). BSP is associated with the overlapping and implosion that occur in the network as a result of either several sources sending the same message or a node receiving the same message from several sources. A better routing strategy is thus necessary that concurrently ensures energy efficiency.

Topology control can result in the formation of a virtual backbone or spine network where routing is confined to. This reduces the BSP issue and saves on bandwidth, energy and processing, since all broadcast is restricted to the backbone network. Communication between any pair of nodes especially those out of transmission range is done through the backbone irrespective of whether one or both are directly connected to the backbone or not. Nodes within the backbone communicate directly with each other while those out of it pass through it to communicate. Many more advantages accrue to this arrangement, which include:

- Smaller routing tables
- Reduced messaging burden
- Quick adaptation to changes in topology
- Improved network coverage
- Improved data collection
- Improved data aggregation

One way to construct a virtual backbone network (VBN) is by using the graph-theoretical notion of a **connected dominating set** (CDS). A CDS is a **dominating set** (DS) which is connected. A DS is a subgraph of a graph where any node of the graph is either a member or a neighbor to a member of the subgraph. A node within the CDS is referred to as **dominator** or **backbone node** while the others are **dominated** or **non-backbone** nodes. The smaller the size of the CDS, the better it achieves its purpose. A smaller CDS size leads to a lower computation and communication cost. It also leads to less contention for channel access and interference. There may exist multiple CDSs in a graph but the one with the fewest number of nodes is termed the **minimum connected dominating set** (MCDS). The construction of a MCDS is NP-hard (Penumalli and Palanichamy, 2015) and can mostly be approximated using polynomial time approximation algorithms. The quality of a VBN formed can be measured by its **approximation ratio** (or factor) which is the ratio of its size to that of an optimal VBN size.

Sometimes the construction of just one VBN is not robust enough. To make the network more reliable, the number of VBNs should be increased as much as it is possible. If the various VBNs are disjoint and distinct, we then have a disjoint virtual backbone network

(DVBN). That way, **activity scheduling** can be applied to the network. Activity scheduling is the alternation of activeness between the different VBNs. This further increases the energy efficiency of the network.

In this chapter, we build on either of the two continuous set formation algorithms proposed in Chapter 4: the continuous geographic or continuous broadcast algorithms, to propose two algorithms for the construction of disjoint virtual backbone networks. It should be remembered that the continuous set formation algorithms end with nodes arranging themselves into consecutive concentric circles around the sink. This is illustrated in Figure 18, where the labels (C1, C2 and C3) inside the circles represent the circle numbers inscribed on the circle counter of nodes (these circle numbers correspond to set number presented in Chapter 4). The orange dot at the center represents the sink.

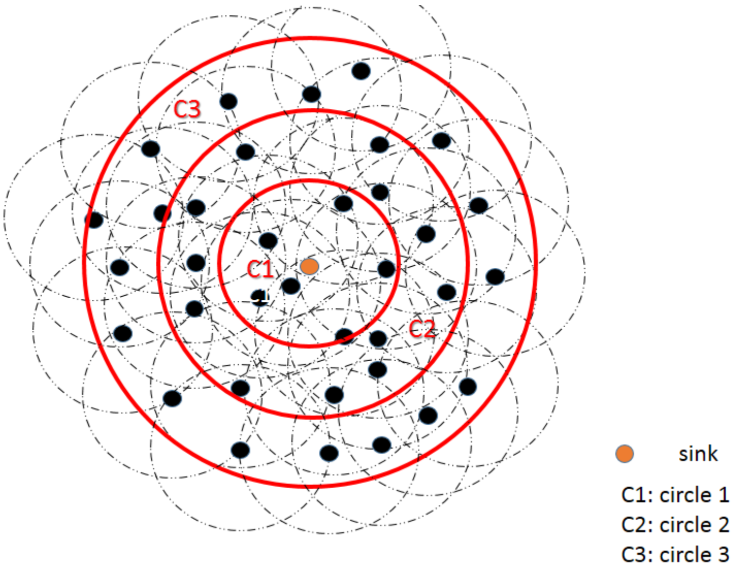


Figure 18: Outcome from continuous set formation

The proposed DVBN formation is accomplished using only nodes' locations or distances especially from the sink and is done in rounds. Each round starts with sink taking action that produces a distinct VBN. Then the sink restarts the procedure for another VBN.

The ultimate goal of forming disjoint VBN is for activity scheduling. While this will ensure added savings in energy, it will also ensure uniformity in the depletion of energy amongst nodes, especially in a regular deterministically deployed network. In a random and redundantly deployed network, it is less evident that energy depletion will be uniform in all nodes because of possible different spatial concentrations of nodes.

5.2 Some Definitions in Graph Theory

In this section we give some definitions on the concept of a virtual backbone as seen from a graph theorist's view. We are simply expressing some terms used in the literature of the concept in plain English while referring interested readers to Diestel, (2010) for more on graph theory.

A dominating set is a graph theory combinatorial concept used in computer networking to solve message broadcasting, network clustering, multicast routing and network backbone formation. In line with the literature of virtual backbone networks, the following terms encountered in this dissertation are defined:

- A dominating set (DS) of a graph *G*, is a subset *X* of *G* such that any node in *G* is either a member of *X* or a neighbor of a node in *X*.
- A connected dominating set (CDS): This is a dominating set that is connected.
- A minimum connected dominating set (MCDS): There may exist multiple CDSs in a graph but the one with the fewest number of nodes is termed a MCDS.
- **Independent Set** (IS): A set of vertices in a graph with no two adjacent to each other.
- **Maximum Independent Set** (MIS): An IS in a graph in which the addition of any vertex from outside annuls the independence of the IS. It is well known that any MIS is also a DS.
- **A Unit Disk Graph** (UDG): A graph made up of intersections of unit disks in the plane. A Unit disk graph allows us to model wireless networks in which all nodes have the same transmission range.
- **A Steiner Tree**: A tree of minimum total length connecting nodes of a subset of vertices in a graph.

5.3 Review of VBN Construction Algorithms

A lot of work has been done on the construction of virtual backbone network, a topology control mechanism in WSN. Historically, topology control has been accomplished through either adjusting transmission power (range) or getting location information. Because these techniques are expensive, the subject has evolved to graph theory, i.e. connected dominating sets. Many algorithms have been proposed for the formation of CDS, which can be classified according to some criteria (Jiguo et al., 2013), the most important being how the topology is arrived at. Following this criterion, we can have them as either centralized or decentralized. In a centralized algorithm, the network is built based on a global knowledge of the network topology. Thus, the CDS building is quick and the size is small. The problems with this class include the fact that the network cannot grow (lacks scalability) and that network information is stored in a single location. A decentralized algorithm builds a VBN based on local information. Here again the algorithms can be either distributed or localized. In a distributed algorithm, each node can decide its role following its state and eventually those of its neighbors. In localized algorithm, a few nodes decide roles for everyone. They can be addition-based, or subtraction based. Addition-based algorithms can either be MIS-based or tree-based. In the former, an MIS is formed then connectors are added to form the CDS. In the later a single or few nodes initiate the formation and others are added. In subtraction-based algorithm, there is pruning of the initial CDS formed by a few or all nodes in the network. Figure 19 presents this taxonomy.

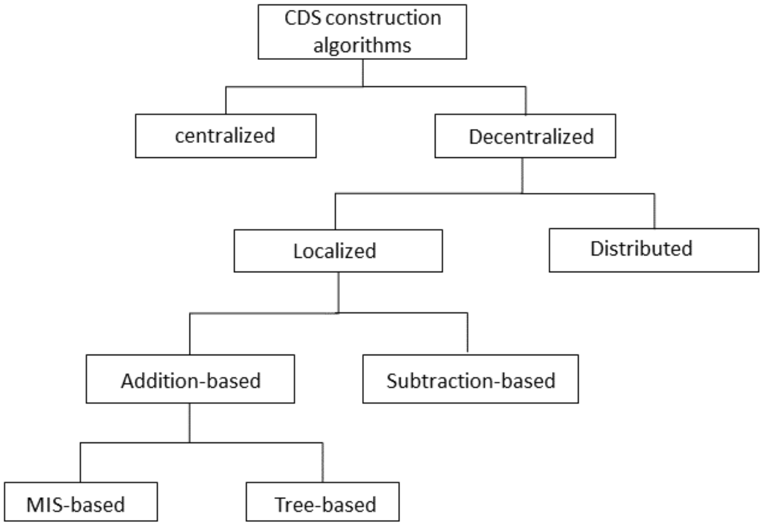


Figure 19: Taxonomy of CDS construction algorithm

According to the classification in this paper, our algorithms on virtual backbone formation fall under a localized class. This placement of our VBN algorithms might be different from that of the continuous set formation algorithm (one of which is geographic) from which it is built.

In Fu et al. (2015), the CDS formation is done in two stages. Firstly, a Maximum Independent Set is formed. A node is chosen to belong to the MIS by virtue of the number of its neighbors. Once a node is chosen, all its neighbors are excluded from belonging to the MIS. Then the MIS nodes are interconnected by adding intermediate nodes to construct a redundancy connected dominating set (RCDS). Choose of an intermediate node for interconnection is also based on node degree amongst neighbors of the MIS nodes. According to the paper, the RCDS is a relatively optimal CDS with some little redundancy. The formation of the CDS as proposed by this algorithm is quite different from ours. One major difference is the fact that our CDS formation does not depend solely on the density of the network.

In Mohanty et al. (2016), a CDS algorithm that works in three phases is proposed. Firstly, there is the construction of a MIS called pseudo dominating set (PDS) because not all nodes in it will eventually belong to the CDS. This construction is based on the fact that nodes within it should be a least number of hops apart. This phase is followed by the construction of a Steiner tree to interconnect nodes within the PDS. The third phase involves elimination of redundant nodes within the PDS, done so as not to loss connectivity and/or coverage. The paper proposes activity scheduling as an improvement for energy efficiency but the procedure to build it is not included in the work. The formation of the CDS proposed in this paper is quite different from our proposal. A main difference is the fact that in our CDS formation there is not initial formation of a MIS.

In Penumalli and Palanichamy (2015) after exchanging messages, energy and mobility of nodes are used to construct MCDS with activity scheduling, in 5 phases. In the first phase, there is dominator selection, where nodes with highest number of neighbors and lowest identities are chosen as dominators. In the second phase, these nodes are interconnected. In the third phase, redundant nodes are pruned. In the fourth phase, using remaining energy of nodes, activity scheduling is accomplished by the dominators selecting equivalent adjacent dominators to replace them. That is, dominators select their replacements in this phase. The last phase of the procedure involves maintenance at intervals. Things that are considered in this phase involve taking care of dominators that have no counterparts to replace them, introduction of new nodes, mobility and remaining energy of nodes. This procedure is quite different from ours as our choice for dominator is not based on remaining energy or number of neighbors.

Another approach to the building of a CDS is clustering. Clustering is the grouping of similar objects from a set of objects. In networking, clustering is the partitioning of a network to ease communications. Nodes groups themselves into clusters, with a cluster head (CH) chosen as the coordinator for that cluster. Nodes within clusters can communicate between themselves and pass through the CH for communication with nodes outside the cluster. The CH is responsible for communication between clusters and with the sink. A backbone network is formed between CHs to create a virtual path. In Dagdeviren et al. (2015), nodes are given weights according to their energy, which is the basis on which the sink choses CHs in rounds. Again, this approach is quite different from ours especially as their dominators are chosen based on remaining energy of nodes.

In Surendran and Vijayan (2015), a CDS is approximated using graph theory principles. It computes ego centrality and uses partial adjacency matrix for computing the importance factor. Importance factor is computed using ego network bridge centrality and a distributed algorithm to form connections between nodes. From the brief description of the procedure for the formation of a CDS given by this publication, it is clear that our procedure is quite different.

In Mohanty et al. (2017), the authors propose a distributed degree-based greedy approximation algorithm which is named Distributed Construction of Minimum Connected Dominating Set (DCMCDS) which constructs a CDS in 3 phases. The first phase of the algorithm is the construction of n MIS in a greedy manner. In the MIS, dominators are 3-hop apart, selected with highest degree amongst neighbors. All in-between nodes are dominated. In the second phase Steiner nodes are selected from the dominated nodes in a greedy manner. The last phase is the removal of redundant nodes (pruning).

In Qureshi et al. (2013), the authors first compared a number of CDS formation proposals that got them inspired to propose a new protocol that they call clique-based CDS discovery (CCDS). This protocol forms a CDS in one phase. Nodes are selected into the CDS on a first-come-first-serve basis. The CDS size is proportional to the network size. An initiator is selected and call clique size 1. It broadcast a message to all its neighbors. Once a neighbor receives, it increments its clique size and rebroadcast the message, adding its identity and remaining energy. This means that a node can receive from nodes with larger clique sizes, but in this case it does nothing. Nodes use timers to know if they are leaf nodes when after an interval, they receive no broadcast from a superior clique node. Backbones are formed progressively as a clique hands-over to a subsequent clique and goes to sleep. In this protocol each clique is a backbone network as it can receive from node just below or above its.

This protocol has a lot of similarities with ours. Firstly, it is initiated in the same way as ours. Nevertheless, there are some significant differences.

1. The number of nodes constituting the backbone network is not optimized in any way
2. It forms a CDS at a stage where we are just forming circles
3. Their CDS is constituted by all the members of a circle

Table 6 summarizes the CDS formation methods applied by the protocols presented.

Reference	**CDS Formation method**
Fu et al. (2015)	MIS-based
Mohanty et al. (2016)	MIS-based
Penumalli and Palanichamy (2015)	Remaining energy and mobility
Dagdeviren et al. (2015)	Clustering
Surendran and Vijayan (2015)	Graph theory
Mohanty et al. (2017)	MIS based
Qureshi et al. (2013)	Clique-based

Table 6: CDS formation protocols

From the foregoing, it is clear that our proposal for the formation of a disjoint virtual backbone is unique. Many previous protocols firstly form an MIS or a PCDS, then proceed with interconnection of nodes in the MIS before pruning. In our case, there are no MIS, no interconnection or pruning process. We form the CDS in a way that it becomes obvious that it is close to minimality.

5.4 DVBN Algorithm

We assume that nodes have implemented either of the two continuous set formation algorithms presented in Chapter 4 and have arranged themselves into consecutive concentric circles. The Disjoint Virtual Backbone Network (DVBN) formation procedure, which is a single phased, proceeds in rounds with each round producing a virtual backbone network disjoint from the previous. Each node has three counters: circle, C, dominator, D and round, R counters. Knowing the circle to which it belongs, a node enters its circle value into its circle counter (circle is the same as set as used in Chapter 4). No further change is made to this counter. The dominator and round counter are both initiated to 0.

A round proceeds as follows. The sink acting as dominator initiator chooses (using dominator setup message) its furthest neighbor as the first dominator for the first circle and for the first round. This node increments its dominator and round counters to 1. Thus, it will have all its counters now having the values 1 (C=1, D=1 R=1). The sink broadcast this Dominator 1 to all its neighbors. Dominator 1 now chooses the furthest neighbor from itself and the sink (sum of the two distances which is tantamount to taking a direction of either clockwise or counterclockwise) in the same circle as the next dominator (Dominator 2) for its circle. This dominator adopts the round and dominator counter values of the sender but only increments its dominator counter. Thus, this Dominator 2 will have its circle counter value as 1, its dominator counter as 2 and its round counter as 1 (C=1, D=2 R=1). This new dominator now selects the neighbor furthest to itself and Dominator 1 (sum of the two distances) in the same circle as Dominator 3. Like with Dominator 2, it now adopts the round and dominator counter values of its sender but only increments its dominator counter value. Other dominators are selected in the same manner until when there are either no further neighbors in that direction to a select a dominator from or a selected dominator realizes that it is a neighbor to Dominator 1 (except the last node in this forward direction is Dominator 2. That is, Dominator 2 has not neighbor in the forward direction). In the first case, the dominator reports back to Dominator 1 (using backwards messaging), who then follows the same procedure to select other dominators in the reverse direction. That is, it chooses the furthest neighbor from itself and Dominator 2. For the second case (the case where the last selected node in the forward direction notices it is neighbor to Dominator 1 and it is not Dominator 2), the dominator becomes the last dominator of the circle.

While the Dominator 1 of circle 1 is choosing Dominator 2 for its circle, it is concurrently choosing its furthest neighbor from itself and the sink in the subsequent circle (2) as Dominator 1 for that circle, broadcasting this choice to all its neighbors. This new Dominator 1 for circle 2 will adopt both the dominator and round counter values of the sender, keeping its own circle counter value. Thus, its counter values will be circle 2, dominator 1 and round 1, (C=2, D=1, R=1). It now follows the same procedure as Dominator 1 for circle 1 to determine all other dominators for its own circle, and subsequent circles until there are no more circles. This is the end of first round (round 1, R1) of the process. This procedure is summarized diagrammatically in Figure 20. In this figure the number of nodes has been greatly reduced as against what a redundant node deployment looks like. As can be seen, we have also limited the number of circles to just 2. Dominators selected are labelled D*i*. The arrows labelled *i*, represent moves in the algorithm. It begins with the sink selecting the furthest node D1 with move 1. There are two moves 2, where D1 concurrently selects D2 in its circles and D1 for the subsequent circle.

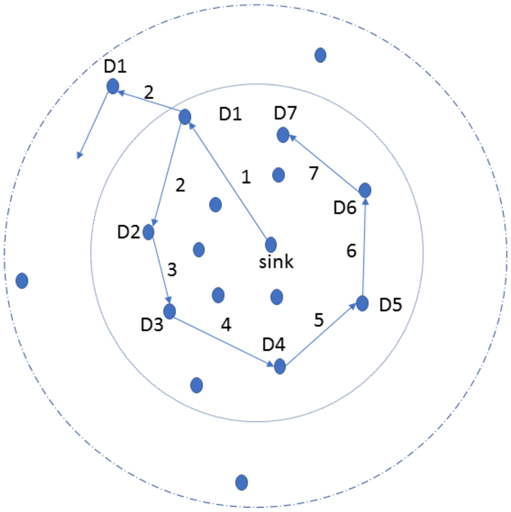


Figure 20: Illustration of our DVBN formation

For the subsequent round, the sink choses the next furthest non-dominator neighbor in the same manner as in round 1, as Dominator 1 of circle 1 of round 2 (C=1, D=1, R=2). This dominator now follows the round 1 procedure to choose other dominators for that round. All selected dominators in the round and all other must not have been previously selected as dominators. The system follows this same procedure to construct of VBNs for subsequent until the number of rounds is attained (considering the density can support the number of rounds).

In Figure 20, returning to circle 1, D2 continues the selection of furthest neighbor which also continues the procedure until we get to move 7 where D6 is selecting D7. Knowing that it is a neighbor of D1, D7 becomes the last dominator, taking no further action and the process ends for that circle. D1 for the subsequent circle follows the same process of D1 for Circle 1, selecting other dominators in its circle and D1 for the subsequent circle. This round ends when there are no further circles to select from.

Round 2 starts with the sink choosing its furthest neighbor from the unselected nodes in circle 1, and the process continues as in round 1 until the number of rounds initiated in the algorithm is reached. A combination of this round with the first is sketched in Figure 21. In this Figure, the sink again chooses its furthest neighbors amongst those not already selected as Dominator 1 (move 8), and the cycle continues as in round 1. This produces another disjoint VBN. In this figure, we show only the beginning of the second round that starts to produce the second VBN in red.

The flowchart for this algorithm is presented in Figure 22. As we will see, if nodes in the network are evenly distributed, the result will be disjoint VBNs with evenly distributed number of nodes per VBN.

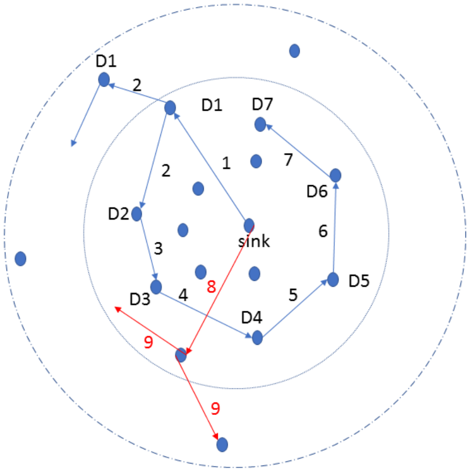


Figure 21: Illustration of DVBN formation

5.4.1 Results and Evaluation of DVBN

In this section we present our simulation results and evaluate the performance of DVBN in order to obtain its approximation ratio. We start with just one VBN, and then proceed for multiple DVBNs.

The following general parameters and variables presented in Table 7, are used on all simulations to obtain the results that are presented.

Coverage area	1000 x 1000
Transmission range:	25 – 100
Number of nodes	500 - 100,000
Number of rounds	1 – 12

Table 7: Simulation parameters for both DVBN and DVBN+

In all simulations, we conduct each experiment ten times and take average values. While watching our results, we start with low density and progress to the point when the percentage disconnected nodes (nodes not connected to a dominator) fail to almost zero. Then we are sure that our VBN cover the area under consideration (CDS is formed). We present some screen shots obtained after some simulation runs in Figure 23 and 24, for different node densities and transmission ranges, respectively.

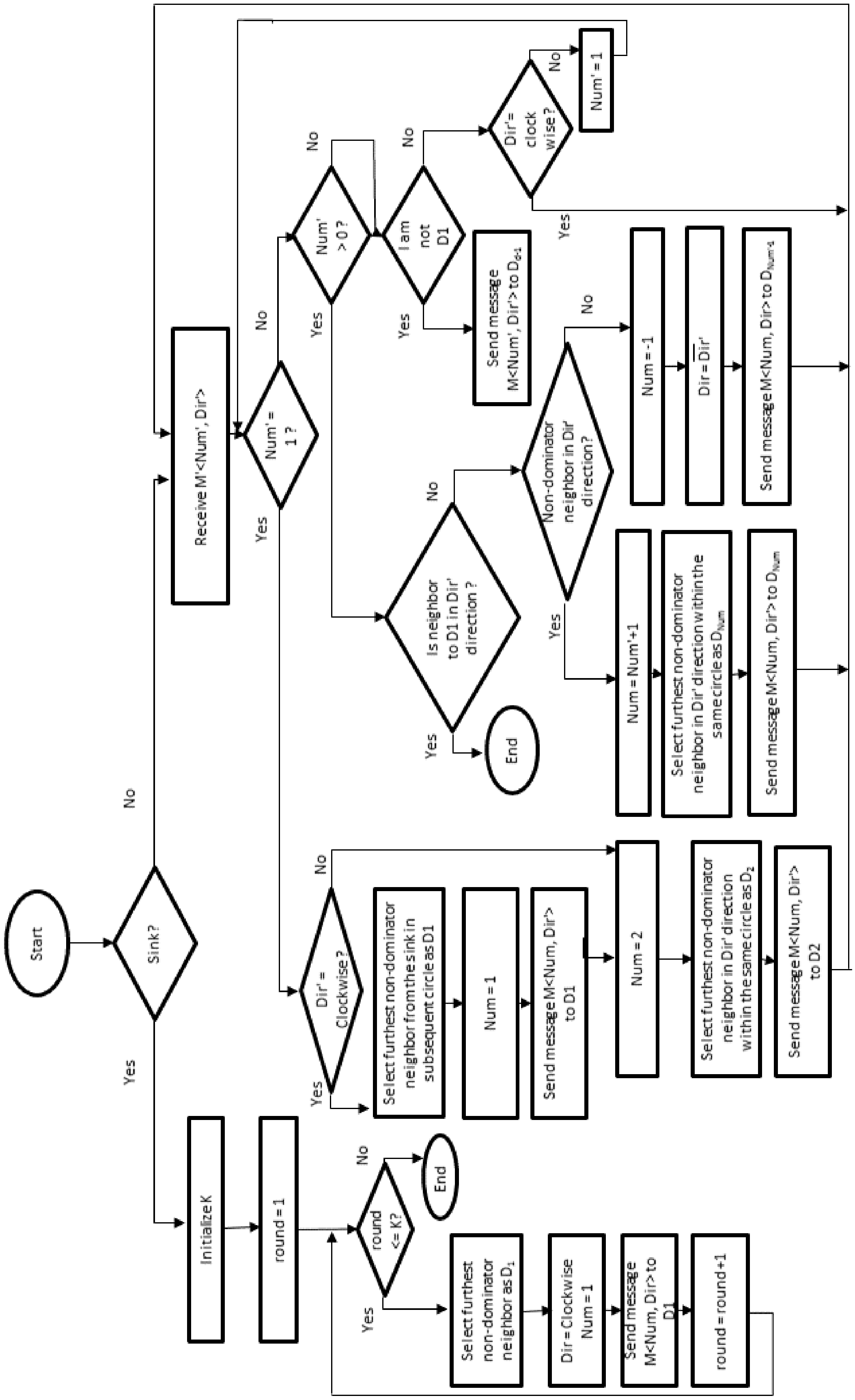
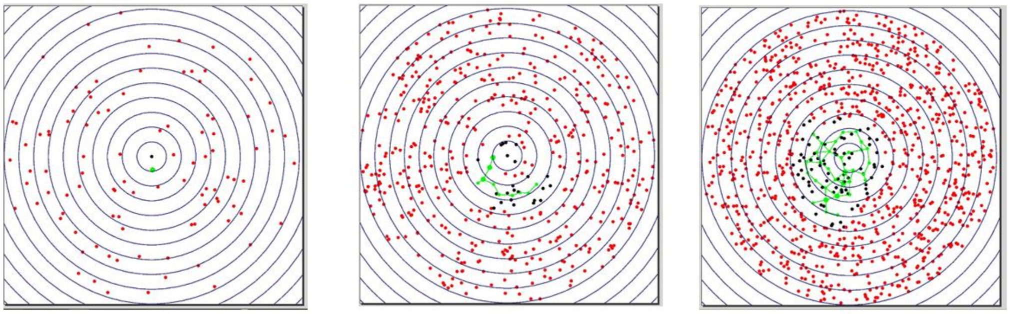
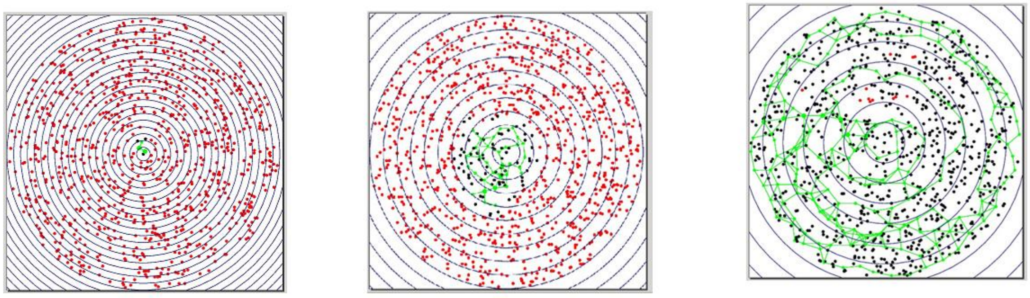


Figure 22:DVBN flowchart



a. 100 nodes with range of 50 units b. 500 nodes with range of 50 units a. 1000 nodes with range of 50 units

Figure 23:Screenshots from DVBN simulation runs, varying node density (reds are disconnected nodes, greens are dominators and blacks are nodes connected to dominators)



a. 1000 nodes with range of 25 units b. 1000 nodes with range of 50 units c. 1000 nodes with range of 75 units

Figure 24: Screenshot from DVBN simulation runs, varying node ranges (reds are disconnected nodes, greens are dominators and blacks are nodes connected to dominators)

Figures 25 to 28 show results obtained from simulation of DVBN algorithm for one round. Figure 25 presents the percentage disconnected nodes versus the number of deployed nodes. In this figure, we start with a situation where the percentage is quite high and stop when the percentages reach zero for all transmission ranges. These are points where all nodes are connected to the VBN. Thus, at these points we have VBNs with as minimal nodes as possible. At the low transmission ranges of 25 and 50 nodes, VBN formation occur at quite a huge number of node deployment (above 100,000 nodes for a range of 25 and above 4,000 for a range of 50 units). DVN formation gets better as transmission range is increased.

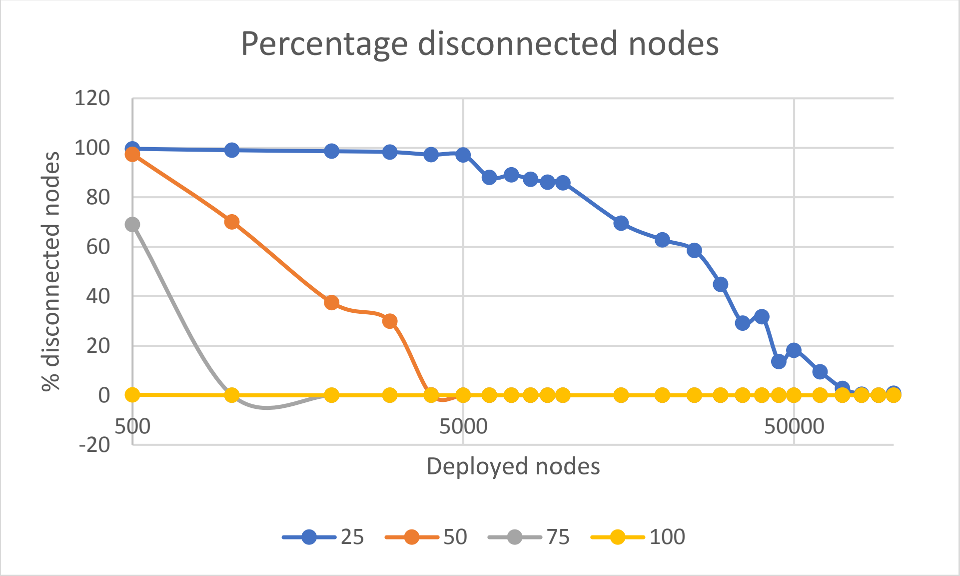


Figure 25: Percentage disconnected nodes for DVBN (range of 25, 50, 75 and 100 units)

We corroborate our stance on Figure 25 with Figure 26, which presents the number of dominators obtained at the point where the percentage disconnected nodes reaches zero versus transmission range. We can see from this figure that as the transmission ranges increase, the number of nodes that constitute the VBN decreases. Though this decrease is not linear, it is expected that the curve will continue to behave in the same way till it becomes 1 (the sink only) as we reach a transmission range of 1000 units. At this point, all nodes are neighbors to the sink which is then the only dominator.

Figure 27 shows the number of dominators versus the number of deployed nodes for the transmission ranges of 25, 50, 75 and 100 units.

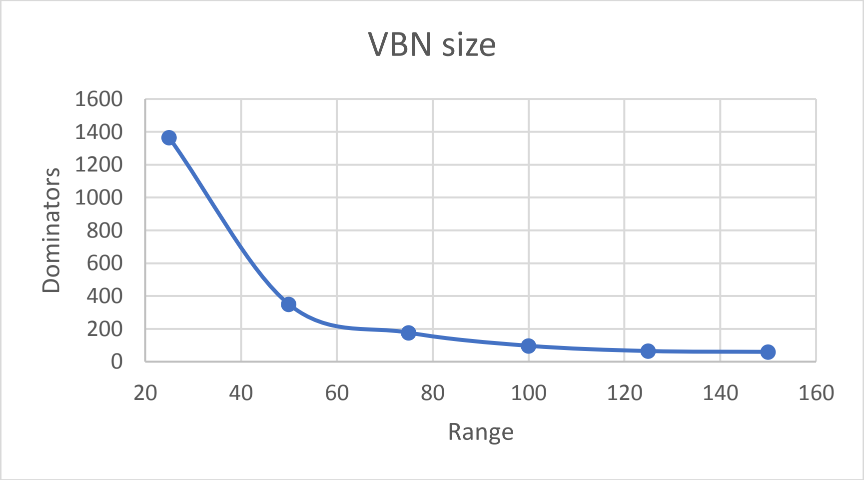


Figure 26: No of dominators vs range for DVBN (for enough deployed nodes)

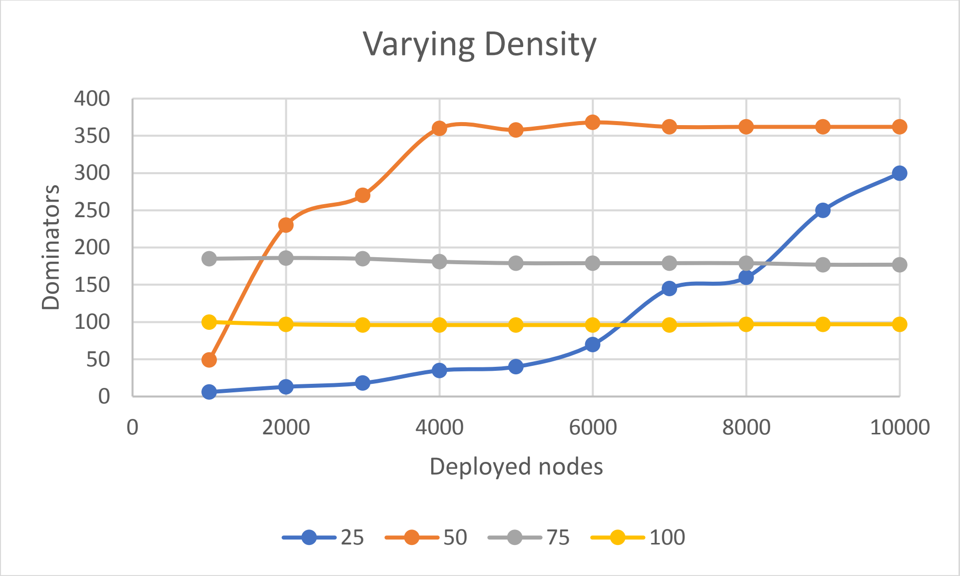


Figure 27: No of dominators vs deployed nodes of DVBN (ranges of 25, 50, 75 and 100 units)

It can be seen from this Figure that as the density is increased so is the number of dominators, especially at low transmission ranges. As expected, the curve flattens out (the increase in number of dominator stops) once all nodes get connected to the VBN. This is so because at low node density, the network is scanty, making it less probable to have nodes within transmission range. Thus, CDS are not formed. At low transmission range, circle radii are small, making it less likely for nodes to be within transmission range. This point is corroborated at high transmission ranges where even at low densities, many more nodes are connected to the VBN. Once all nodes get connected to the VBN, density does not affect the number of dominators. We can deduce that there is a minimal density upon which our algorithm works well.

We now present Figure 28 which shows the number of VBNs formed in relation to the number of deployed nodes. We can see that as the number of deployed nodes increase, the number of VBNs also increase. This is quite consistent with our design as the number of VBN formed depends on the number of deployed nodes.

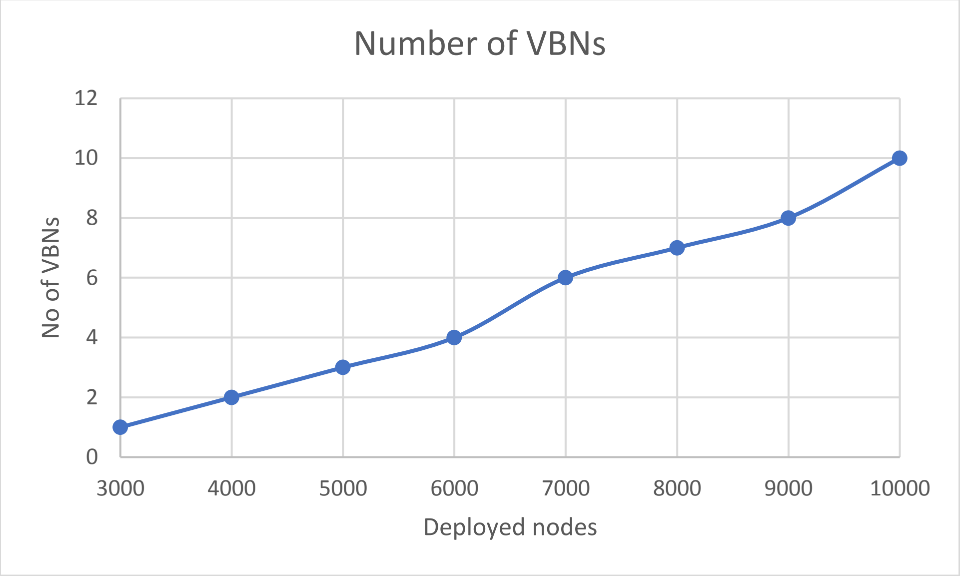


Figure 28: No of VBNs vs deployed nodes (range of 50 units)

We now evaluate the approximation ratio of our DVBN algorithm. In order to be able to have an estimate of it, we compute the ratio of the number of dominators selected by our algorithm versus the minimal number of nodes needed to cover the area (in our case: the disk of diameter 1000 units). This ratio is greater than or equal to the approximation ratio of the algorithm since covering the area is not sufficient, more nodes are possibly necessary to ensure connectivity between dominators in order to have a CDS.

Figure 29 illustrates this scenario. Assuming the node density is sufficient and node deployment is uniform, a minimum VBN must cover all parts of the area being considered. This is a sort of circle packing (circles into circle) problem (Hifi and M'Hallah, 2009). To compute the minimum number of nodes *Nmin* to cover the disk area, we divide the area of the disk by the area covered by a node. If all nodes have communication range *r*, then

$$
\begin{aligned}
Nmin\quad &= Disk\ area/node's\ coverage\ area \\
&= big\ circle\ area/small\ circle\ area \\
&= 500^2/r^2
\end{aligned}
$$

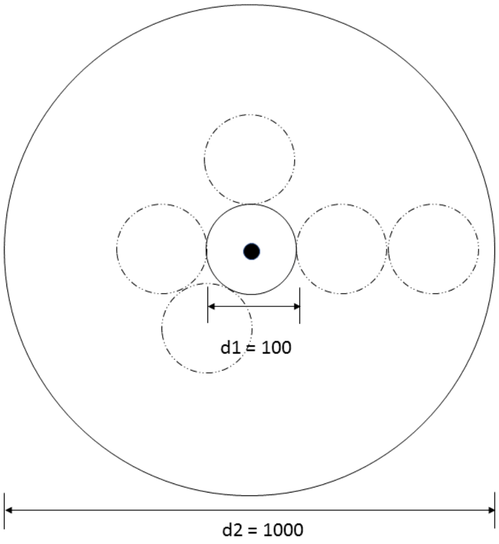


Figure 29: Circle packing

We can see from Figure 27 that VBNs of different sizes are obtained for different transmission ranges at different sufficient densities. That is, if the density is right, we will obtain a minimal VBN of specific size at a specific transmission range. We understand the relationship between transmission range and energy efficiency. At this level, we are not proposing a right transmission range but are instead proposing a table for a relationship between transmission range (as experimented) and approximation ratio (considering the right node density is kept) using this algorithm. Our table is based on disc diameter of 1000 units used. Table 8 presents this relationship.

Transmission range r	Approx. ratio α
25	3.42
50	3.52
75	3.98
100	3.84
125	4.06
150	5.40

Table 8: Approximation ratio per transmission range for DVBN

5.5 DVBN+ Algorithm

We follow-on on DVBN to propose an improved version, that we name DVBN+. Here, the selection rights exclusively held by Dominator 1s in DVBN, to select their counterparts in subsequent circles are transferred to other dominators of the same circles, if a Dominator 1 cannot find neighbors in its subsequent circle. This is a very likely occurrence when the network is scanty. Once a Dominator 1 cannot find a neighbor in a subsequent circle, it transfers the search to the next dominators who also searches for a neighbor in the subsequent circle. This is repeated until either a neighbor is found by a dominator, or the relay comes back to Dominator 1 who then stops the search.

If the dominator finds a neighbor, the neighbor is chosen as Dominator 1 for its circle. Otherwise, the Dominator 1 (in preceding circle) takes no further action. A diagram of this scenario and the flowchart are presented in Figures 30 and 31, respectively. To handle these changes, we have added a third parameter to the messages transmitted which is a one bit field *f* indicating whether the destination of the message has to choose a dominator for the next circle ($f = 1$) or the dominator for next circle has already been chosen ($f = 0$).

In Figure 30, D1 is in circle C1 and cannot find a neighbor in circle C2, hence, it transfers the search responsibility to D3 through D2, who is able to select a Dominator 1 in C2.

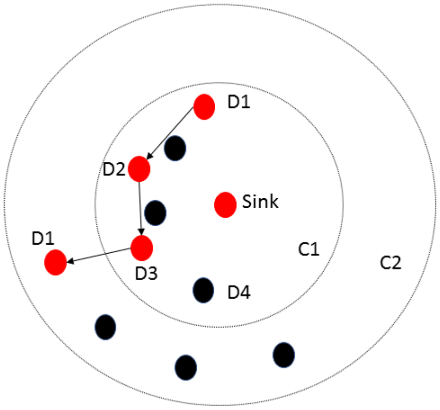


Figure 30: Modifications introduced by DVBN+

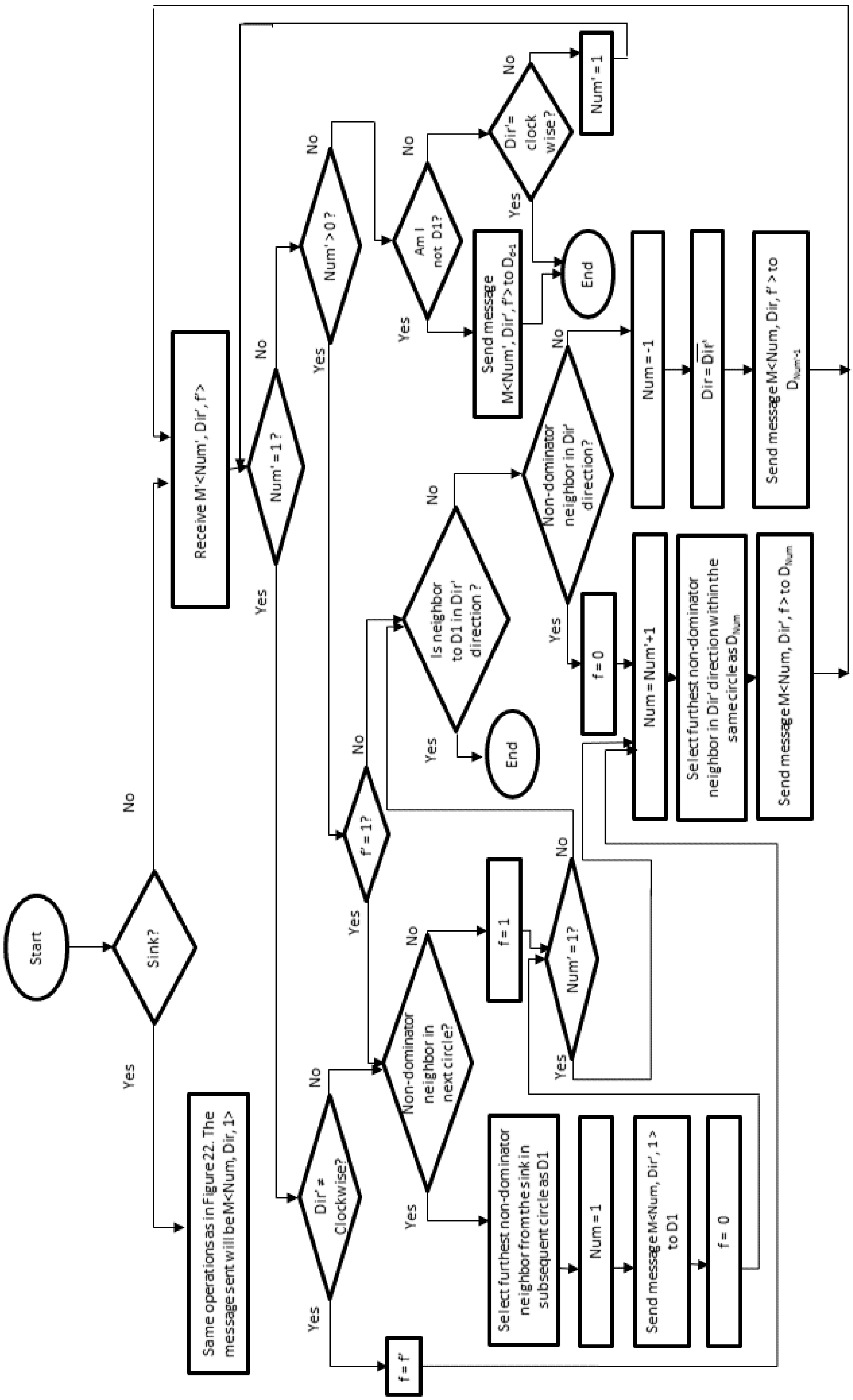
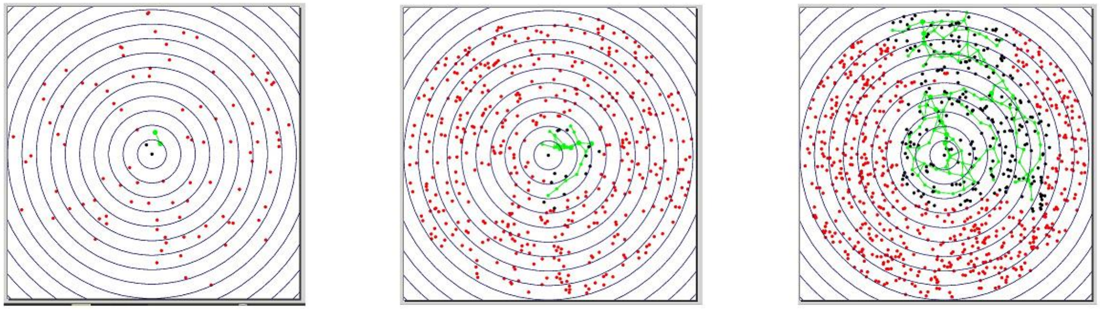


Figure 31: Flowchart for DVBN+

5.5.1 Results and Evaluation of DVBN+

We present screenshots obtained after running this algorithm on a simulation program in Figures 32 and 33, varying node density and transmission range respectively. Circles represent the transmission ranges and the dots represent nodes. As can be seen, compared with DVBN (Figures 23 and 24) with the same values of the parameters, we have less disconnected nodes. We are going to measure this in the next section.

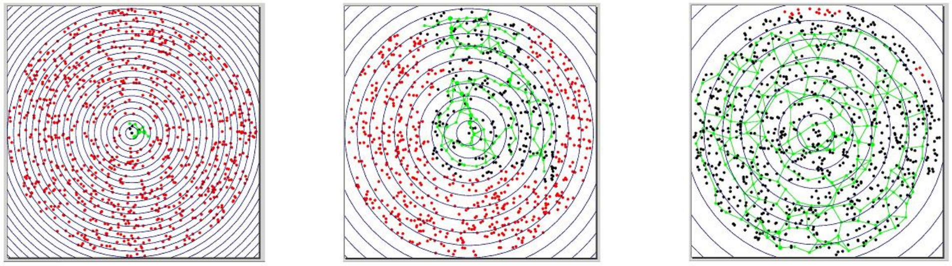


a. 100 nodes with ranges of 50

b. 500 nodes with ranges of 50

c. 1000 nodes with ranges of 50

Figure 32: Screenshots from DVBN+ simulation runs, varying node density (reds are disconnected nodes, greens are dominators and blacks are nodes connected to dominators)



a. 1000 nodes with ranges of 25

b. 1000 nodes with ranges of 50

c. 1000 nodes with ranges of 75

Figure 33: Screenshots from DVBN+ simulation runs, varying ranges (reds are disconnected nodes, greens are dominators and blacks are nodes connected to dominators)

Figures 34 to 37 show the results obtained for DVBN+ for the same scenarios as in Figure 25 to 28 for the DVBN algorithm. While we see the same trend globally for both scenarios, meaning the same justifications, we observe better results particularly in the percentage of disconnected nodes (Figure 34). We node here that DVBN+ forms a CDS with far fewer deployed nodes than DVBN+.

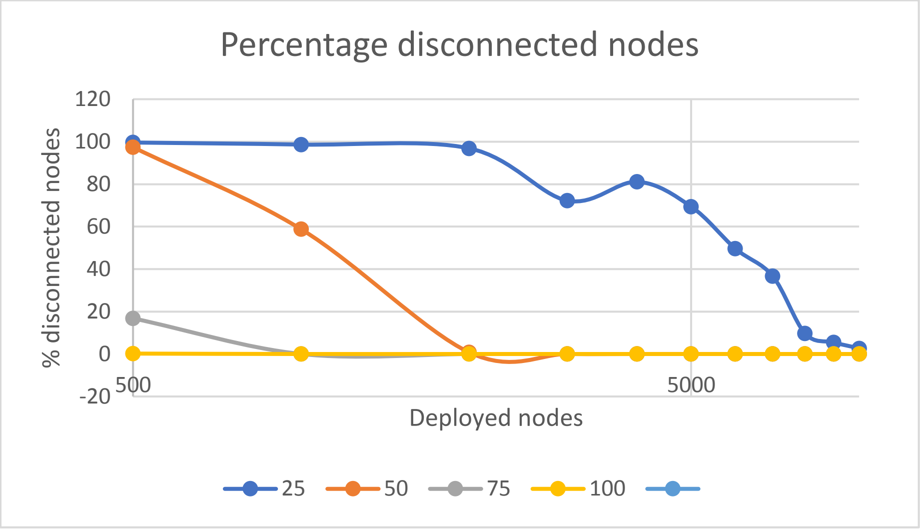


Figure 34: Percentage disconnected nodes for DVBN+

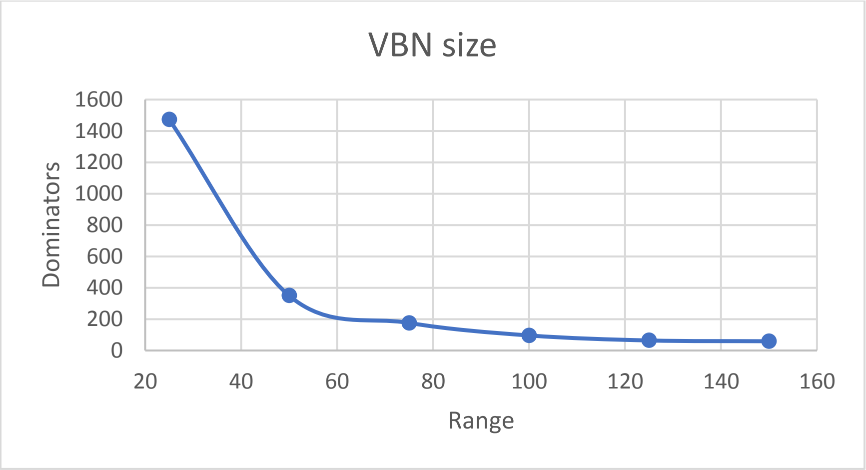


Figure 35: No of dominators vs range for DVBN+

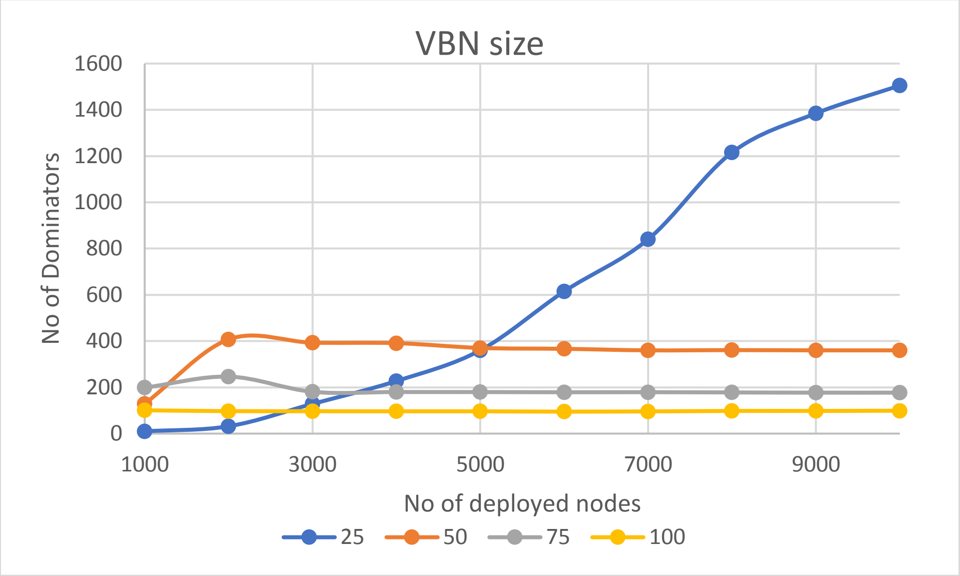


Figure 36: No of dominators vs deployed nodes for DVBN+

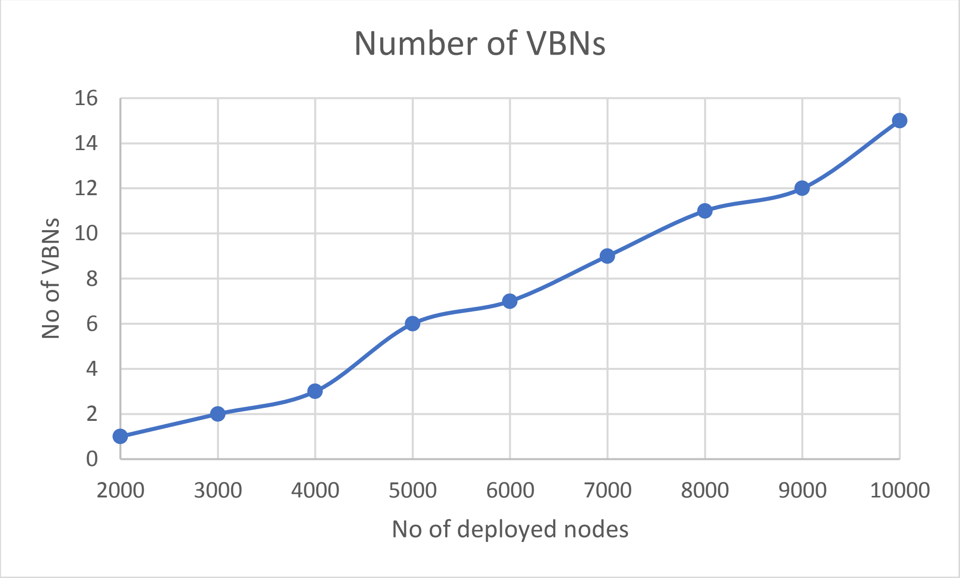


Figure 37: No of VBN vs deployed nodes for DVBN+ (for range of 50 units)

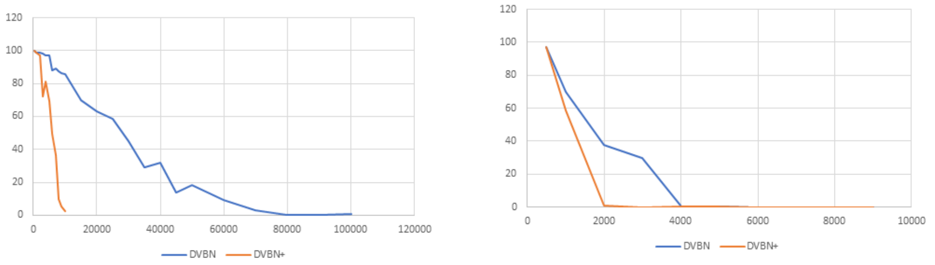
Following the same argument presented for DVBN, we use Figure 34 to compute Table 9, that gives the approximation ratio for different transmission ranges.

Transmission range r	Approx. ratio α
25	3.68
50	3.57
75	3.98
100	3.84
125	3.93
150	5.40

Table 9: Approximation ratio per range for DVBN+

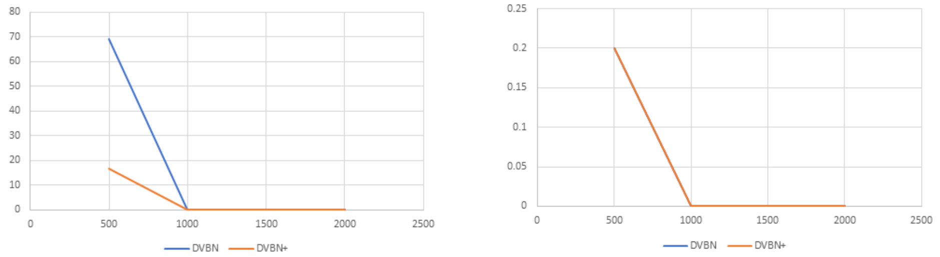
5.6 Comparing DVBN and DVBN+

Figures 38 and 39 compare DVBN and DVBN+. Figure 38 compares the two algorithms in terms of disconnected nodes at different ranges while Figure 39 does the same for the number of dominators. Percentage disconnected nodes demonstrates the ease with which a complete VBN is formed. This is indicated in the graph by the rate at which the graph approaches 0. Generally, the percentage of disconnected nodes approaches 0 as the number of deployed nodes increase. This is evident in all the graphs in Figure 38. However, in (a), the graph of DVBN+ falls to 0 far quicker than that of DVBN. This trend slows down until both graphs merge at a range of 100units as demonstrated in (d). This means that DVBN+ fairs better than DVBN for low range and when the network is scanty.



a. Range of 25 units

b. Range of 50 units



c. Range of 75 units

d. Range of 100 units

Figure 38: Percentage disconnected nodes for DVBN and DVBN+

The number of dominator is also an indication of how easy it is to form a VBN. Just like with percentage disconnected nodes, the number of dominators increases with the number of deployed nodes up to a point when a complete VBN is formed. Thereafter, it is expected that it will flatten up, when a VBN is formed. As can be seen in Figure 39, the number of dominators formed for DVBN+ is higher at low range, then converging as the range increase. This further confirm our assertion that DVBN+ is better than DVBN at low range and density.

We can then conclude that DVBN+ is more appropriate at low transmission range and low network density.

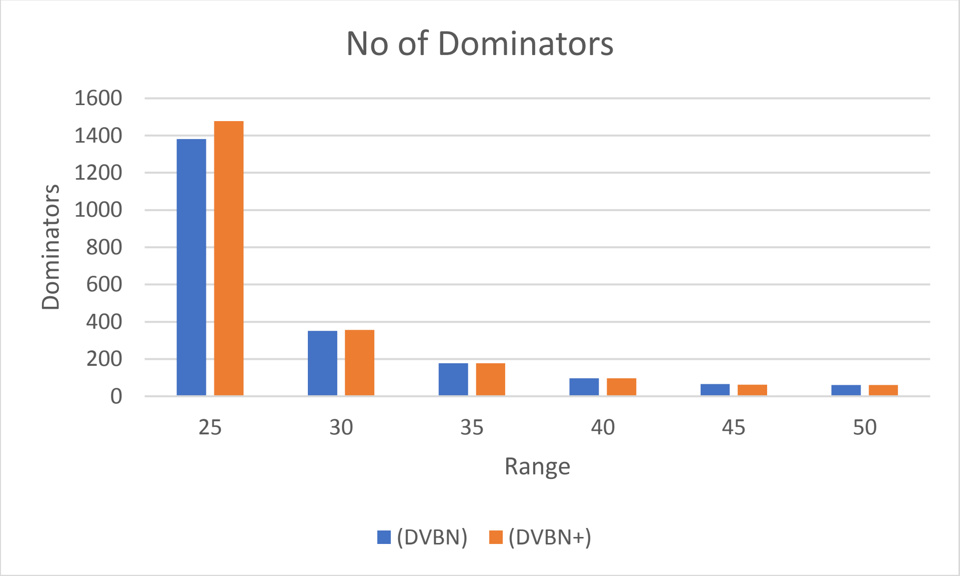


Figure 39: Number of dominator per deployed nodes for DVBN and DVBN+

5.7 Conclusion

In this chapter, we have proposed two localized algorithms to construct multiple disjoint virtual backbone networks using only node location or distance from each other. We have shown that our algorithms are able to construct VBNs that are dominating sets. Our algorithms are amongst the most efficient with approximation ratio ranging between 3.48 and 5.40.

Chapter 6

Sleep Relay Protocol

In this chapter we propose Sleep Relay Protocol (SREP) to be run upon any of the repetitive set formation schemes proposed in Chapter 4. In this protocol the duty cycle active time is relayed between the different sets of nodes formed. The proposal negates the rather conflicting reality linked with the application of duty cycling in WSN. While the application of duty cycle improves energy efficiency, it introduces degradations in network performances like latency, reliability and packet delivery ratio. Our proposal aims to demonstrate that duty cycling can be applied to this network without this performance degradation.

6.1 Introduction

While duty cycling can lead to a significant gain in energy, it imposes certain drawbacks linked to the fact that while a node is asleep, it is unreachable. These drawbacks include increased latency, low packet delivery ratio, unreliability and unavailability of the network to communicate. In most WSN applications, some of these issues are not very worrisome because the traffic is usually light (Suhonen et al., 2009). In certain others, some of these issues can be quite critical. Delay, for example, can be fatal in certain applications that involve human lives.

In synchronous duty cycling schemes, the effect of some of these drawbacks are not very pronounced. For asynchronous duty cycling scheme, the situation can be quite serious. Fixing may require adjustment of duty cycle values, their schedules, preamble lengths, etc. It may be advisable to fix some of these online as when done in advance may result to low flexibility. A dynamic duty cycle scheme may alleviate some of these problems. In other cases, there may be need for dual radio, where a low power radio is duty cycling to monitor network activity while the power-hungry radio only comes on for data transfers. In all cases, the goal of energy efficiency supersedes all the difficulties imposed by it.

An important performance limitation imposed by duty cycle is temporary unavailability of the network to communicate (Kim and Liu, 2008), since a node cannot communicate with its radio off. Availability has been expressed in (Suhonen et al., 2009) as the probability to receive a new measurement (data) from a node within a certain waiting period. We simplify this definition to; the percentage of time an active path exists between any source and the sink. In this work we will equate availability to packet delivery ratio (PDR). PDR measures the ratio of received to sent packets. A closely related issue imposed by duty cycling is reliability. This too is expressed as the probability of successful packet delivery from a source to a destination. In the face of duty cycling, especially in an asynchronous scheme, two strategies exist to improve reliability:

1. Retransmission: This ensures that lost packets are retransmitted continuously until either they are received after a number of trials or considered undeliverable and dropped.
2. Transmission range: Increase in transmission range makes network to be more reliable as the transmission has more chances of meeting many more nodes awake.

Two complementary strategies to reduce delay include:

1. Duty cycle: Transmission delay can be reduced by increasing duty cycle, but this is contrary to our requirement where we desire a small duty cycle for energy efficiency.
2. Relaying nodes: In an asynchronous duty cycle scheme, increasing the number of relaying nodes increases the chance of a transmission meeting another awake.

In this chapter we aim at improving energy efficiency through duty cycling while still upholding performance. We propose Sleep RElay Protocol (SREP), where nodes covering the area to be sensed are grouped into interleaving sets according to their location in the network. Nodes in same set sleep at the same time, relaying the sleep to other set(s) in one duty cycle period. Each set forms a connected path between the epicenter and the sink.

Only one set can be active at a time.

This protocol also ensures even distribution of energy depletion amongst nodes. That is, the energy of all nodes in the network depletes evenly. A WSN always follows the many-to-one paradigm where many more nodes transmit to the one sink. In this process, the near-sink nodes experience a bigger burden as they need to relay data from many more far-sink nodes. These usually leads to energy holes that can eventually segment the network. In other words, these nodes die faster than the far-sink nodes and once dead, the whole network becomes segmented, unusable or unavailable. Studies have shown that when energy hole occur in this way, 90% of the network energy goes with it (Liu et al. 2018). This is very wasteful indeed.

6.2 Review of Set Relay Protocols

Some works have been done in the alternation of the sleep schedule of sets of nodes in the literature of WSN. The aim of reviewing them here is to permit us better place our proposals.

Slijepcevic and Potkonjak (2001) proposed a heuristic that organizes the network into mutually exclusive sets with only one set active at a time while all other sets are made to sleep. The location of members of the same set span the entire coverage area. After some time, another set takes over being active while the former is deactivated. This arrangement leads to a significant saving in energy. The proposed algorithm aims at maximizing the number of sets. While the general form and goals may be quite similar, this proposal has the following differences with ours:

1. It targets coverage optimization. We are more interested in connectivity, assuming (for simplicity) that sensing range is the same as communication range
2. Deployment in this algorithm is not random as nodes are assigned to sets. In our case, nodes choose the sets to belong to according to their distances from the sink.

In the work of Ghosh and Givargis (2005), topology management by the formation of sets is done a bit differently though still resulting in a small set of nodes being active at the same time while the rest are *off*. The goal is to maintain total coverage like in the above case. The protocol is localized in that it is only executed at a dying node, precluding all others from participating. While this leads to a lot of saving in energy, it may result to fragmentation, causing early dead of the network. Moreover, it is also reactive in nature. That is, it is only executed when a change of state of the network can lead to a decrease of coverage degree. If the dying of a node does not have the potential of leading to a coverage hole, the update mechanism of the protocol is not executed. All these flavors; localization, reactivity, etc. are not included in our proposal.

A data collection scheme that is not only close to our proposal by some words in its name but also in procedure is proposed in (Liu et al., 2018). A source or relay node selects a set of reliable nodes as relay nodes to relay data to the sink. The Adaptive Virtual Relaying Set, AVRS scheme arranges nodes into circles according to their location from the sink, just like is done in our continuous set formation algorithm. Among inner circle nodes, (relaying set, RS), a node chooses a sub-set of nodes as its virtual relaying set, VRS. These are nodes that are more reliable to act as relay nodes. It is then very probable that this node will hit the awake time of one of these VRS nodes when it sends data to be relayed to the sink. This greatly enhances performance without affecting energy of nodes. In such a scheme, the duty cycle can be made very low. This scheme has no deliberate relaying of sleep from one set of nodes to another. Additionally, the duty cycle scheme followed is asynchronous while ours is synchronous. This proposal is therefore fundamentally different from ours.

In a WSN where the network monitors a mobile target, it is important to predict the location of the target at any point in time in order to ensure that only nodes around its trajectory will be awoken to monitor it as it moves. That way, all the other nodes out of its track at a particular time will be put to sleep to save energy of the network. In (Borkar and Bhoomarker, 2014), Probability-based Prediction and Sleep Scheduling Protocol (PPSS) is proposed to firstly predict the location of a mobile target and to proactively awaken a small and very precise set of nodes to monitor the target. The meaning of this is that, as the target keeps moving, so do the sets keep changing. There is therefore a relay from one set of nodes to another as the target keeps moving. These can lead to a huge saving in energy.

This protocol is not fundamentally different from ours, except in a number of ways:

1. Sets are formed on-demand instead of proactively as in our case.
2. It does not ensure even depletion of energy amongst nodes like in our case

In Arroyo-Valles et al. (2017), the authors stressed on the selection of an optimal minimal set of nodes to be activated based on the level of visibility of the phenomenon they are supposed to measure and their placement on the route to the sink. All other nodes are left to sleep. No information is given about replacing this set with another at some other time. Though, this saves energy, there is no relaying role in this protocol. Therefore, the protocol is quite different from ours.

In Oliveira and Castro (2015), the paper firstly presents BiO4SeL which like an earlier discussed protocol, is an autonomous protocol for monitoring moving targets. It does route discovery and maintenance using the ant model. Bio4SeL works in a distributed and autonomic way to find routes between sources and destinations. The ants deposit pheromones on nodes on the best route to the destination which evaporates with time. The pheromone in this case is remaining energy. The less energy a node has the faster the pheromone evaporates. In other words, nodes on best routes are marked according to how much energy they have. As the energy is depleting so does the mark evaporates, invalidating the route. The paper builds on BiO4SeL by adding node redundancy and duty cycling to its operations to propose BioSched. BioSched firstly looks into the problem of energy consumption distribution associated with BiO4SeL. That is, the BiO4SeL as described results in energy depletion not being evenly distributed but more concentrated on nodes in the best route. Secondly, it aims to optimize network lifetime. Thus, while the set of nodes that constitute the best route with enough energy is active, BioSched puts the subset of nodes with less residual energy to sleep while ensuring good delivery rate. This protocol has some similarities with ours but proceeds differently.

In Benzerbadj et al. (2018), another protocol which appears quite similar to ours but fundamentally different is proposed for monitoring a fenced area. Two types of nodes are identified with different roles:

1. Sentinel nodes: These are nodes that "see" the fence and are kept active at all times.
2. Duty-Cycled Relay Nodes (DC-RNs): These are nodes removed from direct view of the fence. These ones are duty cycled and used to relay data to the sink.

In this protocol, there is no relay of role from one set to another and thus, it is fundamentally different from ours. The fact that it has nodes with the word "relay" and "duty cycle" in their names calls for clarifications; reasons why it is mentioned here.

In Deb et al. (2001), a topology discovery algorithm, TopDisc for wireless sensor network aimed at managing the network is proposed where a set of distinguished (backbones) nodes provide information about their neighbors and thus the topology of the network to an initiator, the monitoring node. To save energy these distinguished nodes put the others (redundant nodes) to sleep. The distinguished nodes form a tree of cluster, TreC rooted at the initiator of the protocol, the monitoring node. The TreC, therefore forms the topology (logical organization of the nodes) of the network that enables proper management of the sensor network. To reduce communication overheads, only the distinguishing nodes reply to probes from the monitoring node. This protocol doesn't tell how the set of distinguished nodes relay their function to other sets.

As have been seen in this section, several methods exist to relay active time between sets. By the way the sets are created and the process of alternation between sets, our proposed protocol SREP is different from all other proposals.

6.3 Sleep Relay Protocol

We describe our procedure using the repetitive set formation. The same principle can be applied to the disjoint virtual backbone network.

After nodes have arranged themselves into repetitive sets, the sink relays the duty cycle, its start time and periodicity to all sets. This message is relayed between members of each set. Nodes thus calculate their sleep schedules and start following them. The duty cycle value is divided into slots of equal size according to the number of sets. All nodes belonging to a set follow one duty cycle slot and relay it to next set at the end of the slot. This is continued until all the slots are consumed in one cycle. Then the cycle continues. Only one set is active in a slot. Figure 40 diagrammatically shows the output of the repetitive set algorithm upon which SREP is built. It is also possible to apply this protocol to the disjoint virtual backbone network as shown in Figure 41.

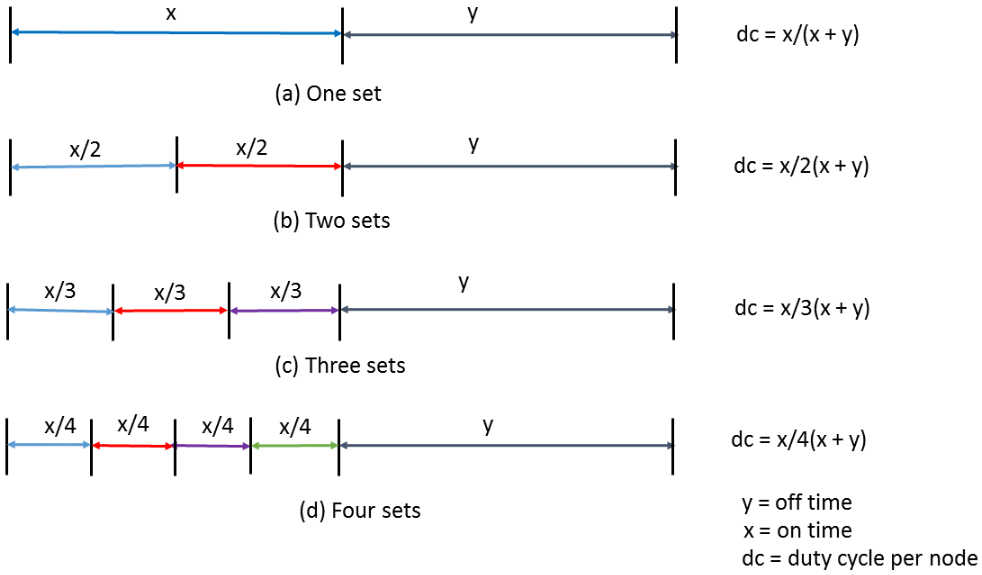


Figure 40: Conceptual view of active time with SREP from repetitive set formation

Once a source has information to send, it isolates itself from its set, checks its cache for the set that is *on* or will come *on* soonest. It then sends the data to a node in that set closest in direction to the sink. Since this set forms a continuous path to the sink, it is sure its transmission will get to the sink with the least delay. We note that in our case, it is only the radio that is duty cycling. Other subsystems of a node, particularly the sensor, is maintained in an active state.

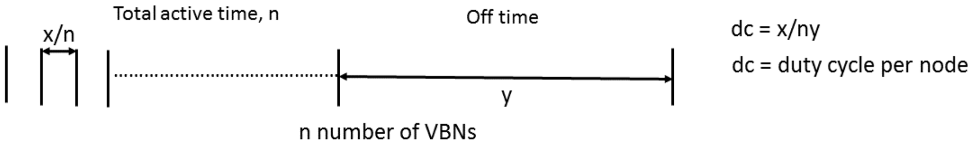


Figure 41: Conceptual view of SREP active time for disjoint VBN

We can see from the diagrams that the duty cycle is divided equally amongst the sets and the cumulative duty cycle is the same for all the formations. The setup message format and flowchart for this protocol are presented in Figures 42 and 43, respectively.

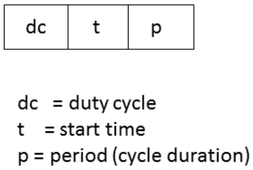


Figure 42:Message format for setup messages

As shown in Figure 42, the message format has the following fields

a. dc: this is the duty cycle for the whole network
b. t: this is the start time of the duty cycle
c. p: this is the periodicity of the duty cycle (duty cycle duration)

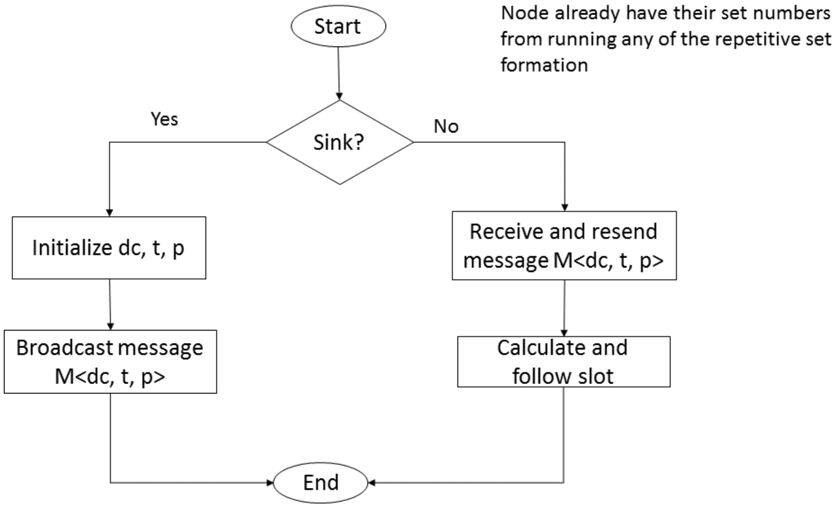


Figure 43: Flowchart for SREP

6.4 SREP Evaluation

In this section we evaluate the performance of our proposal using NS-2 simulator. As has been discussed, the network is random and redundantly deployed with the sink at the epicenter and nodes scattered around it in a circular formation. Leaf nodes as sources are placed at the extremes. For our simulations we take a cross-sectional cut of this network as shown in Figure 44. The figure shows how nodes are scattered around the sink.

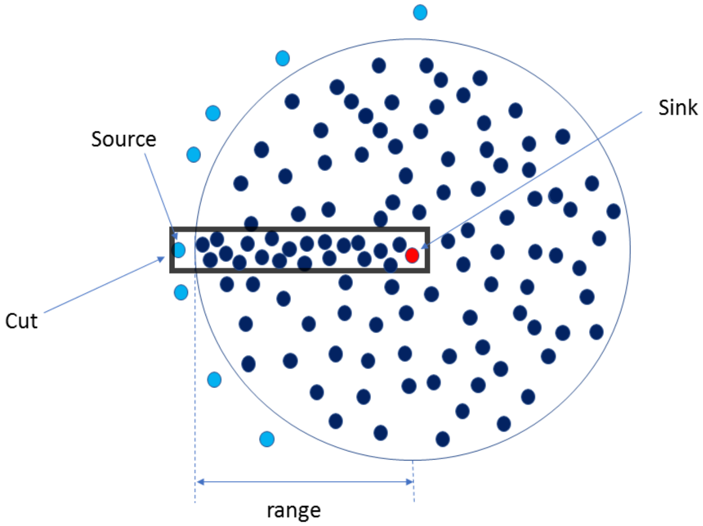


Figure 44: Cross-sectional view of network to be simulated

We work with 22 nodes, spaced at 12m apart. The source and the sink are placed such that they cannot directly communicate without a relay node. We firstly evaluate our proposal on the repetitive sets. Simulation parameters are presented in Table 10. Each simulation is done ten times independently and except indicated otherwise, average results taken.

6.4.1 Energy Efficiency

We see from Figure 45 that the average energy consumed decreases sharply with the introduction of the second set. Thereafter, the decrease slows down and almost flattens out as we approach 6 sets. These averages are on nodes taking part in routing data packets to the sink since not all nodes in the transmission range take part in relaying data in NS-2. In a real network, we expect the energy consumption to be uniform in all nodes, especially where no routing protocol is applied. Figure 46 accentuates the fact that not all relaying nodes take part in communication. It shows remaining energy per node for some simulation runs. From this Figure the nodes taking part in data relay and the savings derived can clearly be seen. These results are evident as the arithmetical increase in number of sets introduces but a geometrical decrease in duty cycle and thus energy consumption.

Parameter	**Value/Type**
Number of nodes	22
Distance between nodes	12 m
Simulation time:	50s
Transmission range	250m
Cumulative Duty cycle:	60%
Application:	CBR
Packet size:	512 bytes
Routing protocol:	AODV
Initial Energy:	10.1J
Transmit power:	0.66 W
Reception power:	0.395
MAC protocol:	IEEE802.11

Table 10: Simulation parameters for SREP

As Figure 45 shows, we cannot keep increasing the number of sets indefinitely as the gain in energy does not follow suit. For the case simulated, it is very clear that after the second set, the gain starts slowing down and almost flattens out by the sixth set.

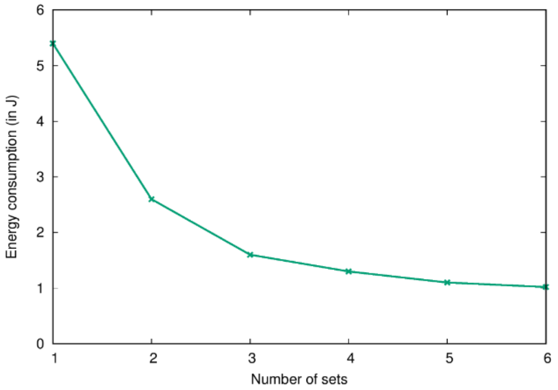


Figure 45: Energy consumption on active nodes vs no of sets

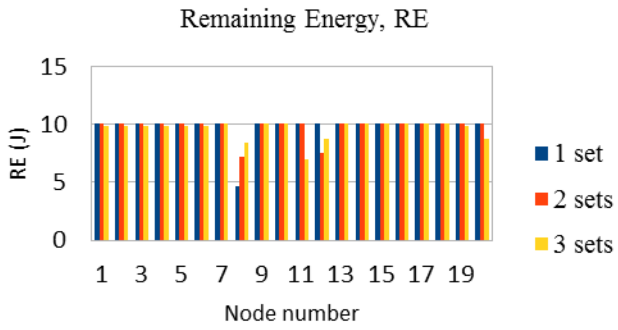


Figure 46: Remaining energy per node for SREP with 1, 2 and 3 sets

The benefits of set formation could include more than just energy efficiency. As we have seen in Chapter 5, set formation can enhance routing and data dissemination in a network. Uniform energy depletion amongst nodes could be another.

6.4.2 Packet Delivery Ratio and Latency

Figures 47 and 48 show the packet delivery ratio and latency. We can see them staying almost constant as the number of sets is increased. For latency, its constancy can be attributed to the fact that our duty cycle scheme is synchronous in nature. As for the PDR, its constancy is due to the fact that the overall (cumulative) duty cycle of the network stays the same irrespective of the number of sets.

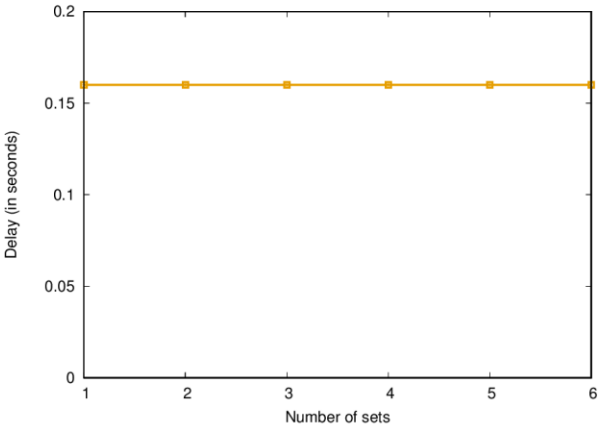


Figure 47: Delay vs no of sets

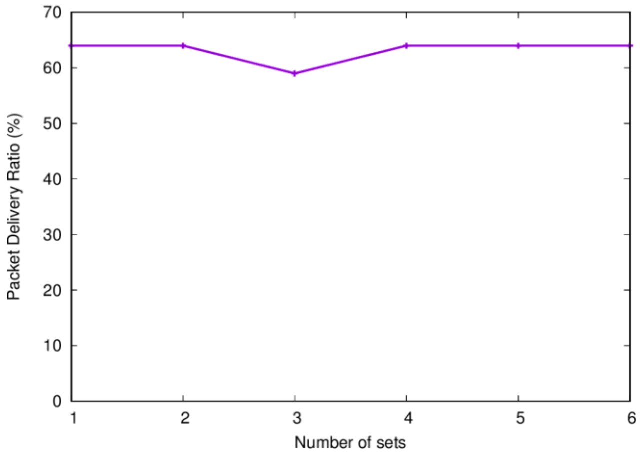
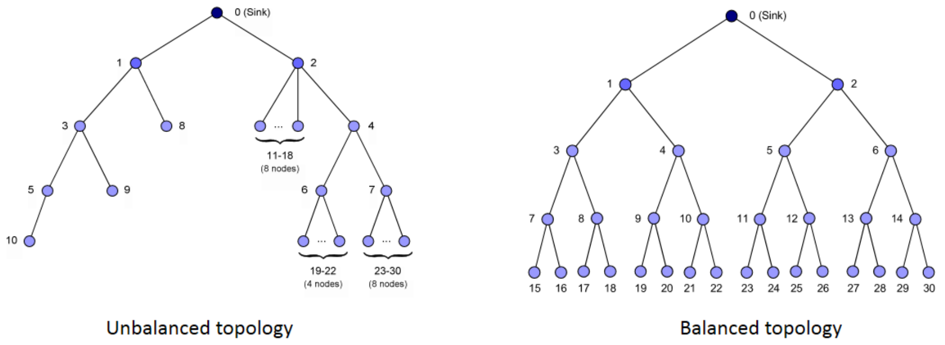


Figure 48: PDR vs no of sets

6.4.3 Comparing SREP with Adaptive Staggered Scheme

We now compare SREP with an adaptive staggered scheme proposed in Anastasi et al. (2006). Their topology is presented in Figure 49. Figures 50 and 51 show results obtained. This protocol has been reviewed in Section 4.2. In this simulation, we use the same parameters therein.

We observe that there is almost no difference in PDR. This is due to the fact that the cumulative duty cycle used for each protocol is the same. The significant gain in latency can be attributed to node scheduling within sets inherent in SREP compared to adaptive staggered protocol with a special scheduling within nodes.



Unbalanced topology

Balanced topology

Figure 49: Topology of adaptive staggered scheme

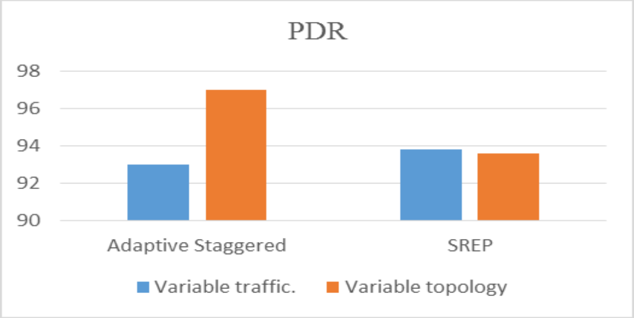


Figure 50: PDR comparison

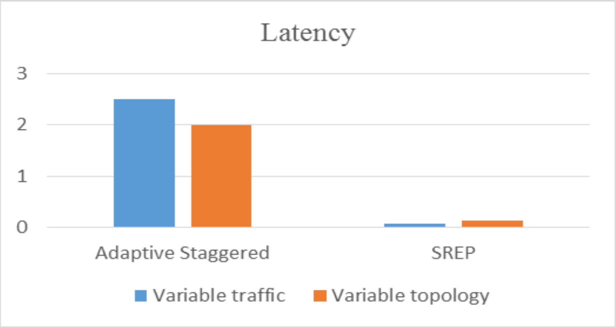


Figure 51: Latency comparison

6.5 Conclusion

In this chapter, we have proposed Sleep Relay Protocol (SREP) as a protocol that can be used to effectively manage duty cycles amongst nodes in a network. The protocol negates the impact that duty cycle has on performance if not improving it. It has been established that just increasing the number of sets for repetitive algorithm does not guaranty improvement in energy efficiency to the same level. Just with two sets, we observe that there is quite a huge gain in energy. This diminishes as we increase the number of sets. The procedure and conclusions that have been arrived at here can equally be extended to both DVBN and DVBN+ algorithms. Here, sets will correspond to VBNs.

SREP as proposed is based on synchronous duty cycling. As has been seen in Chapter 2, synchronous duty cycles have some drawbacks. As a way forward, we will look on how our protocol can be applied to an asynchronous duty cycling system. Moreover, we will evaluate both synchronous and asynchronous versions to more closely related protocols like (Slijepcevic and Potkonjak, 2001) and (Borkar and Bhoomarker, 2014).

Chapter 7

General Conclusions

In this chapter, we close this report with some conclusions and proposal on the way forward.

7.1 Resume of Our Contributions

Tremendous advances have been made in all areas of wireless sensor network technology, yet energy has refused to keep pace. This is based on the fact that battery still remains the main source of energy, though new ways of powering the system have evolved. Battery are quite suitable not only because of their sizes which matches that of nodes, but battery powered nodes are more easily deployed.

In many applications, once the battery power is depleted, the node becomes useless. This leaves us with network exploitation as the mean variable for extending battery life and thus that of the network. It is imperative that efficient exploitation of all systems of the network be put in place. This is where we come in.

With all these in mind, we set out to act in the area of topology management. Topology management has to do with nodes in the network arranging themselves in a particular order to enhance performance. In this case the arrangement is in order to apply duty cycle. Duty cycle has been shown to be the most efficient technique for energy management in WSN. Nodes alternate between awake and sleep in a cycle. While this improves the energy efficiency of nodes it also introduces some negative effects on network performance.

In this work, we aimed at applying duty cycle without losing out on network performance. Before applying the duty cycle, we need to let the nodes arrange themselves in particular manner. Thus, we proposed five algorithms which we group into two classes: repetitive and continuous class.

A. Repetitive Set Formation Algorithms: These arrange nodes into interleaving sets with a set covering the entire network. A set of nodes extends such that it ensures total connectivity of the network. The algorithms in this class are:

   1. Repetitive geographic algorithm
   2. Repetitive broadcast algorithm
   3. Multi-signal broadcast algorithm

In the repetitive geographic algorithm, each node can independently determine the set to which it belongs using only its knowledge of its location and that of the sink. In the repetitive broadcast algorithm, a setup message is broadcast by the sink to its neighbors who update the message before rebroadcasting to their own neighbors. These setup messages contend data which permit nodes to determine their sets. In the multi-signal broadcast algorithm, the sink broadcast multiple messages with multiple transmission ranges. These message once received are updated by neighbors who rebroadcast them to their own neighbors. The information in these message permit nodes to determine their sets.

B. Continuous Set Formation Algorithms: These algorithms arrange nodes such that different sets interact to ensure connectivity of the network. The number of sets formed depend on the transmission range of nodes and coverage area of the network. Algorithms in this class are:

   1. Continuous geographic algorithm
   2. Continuous broadcast algorithm

These two algorithms are derivatives of the first two in the repetitive set formation algorithm class. The only difference is that they are designed to produce sets in a consecutive order as opposed to a repetitive order.

C. Virtual Backbone Formation Algorithms: Here, two algorithms for the construction of disjoint virtual backbone networks are proposed with one being an advanced version of the other. The algorithm runs in rounds with each round producing a distinct backbone network. In the basic version, the sink selects (for the first round) the first dominator (Dominator 1) in the first circle based on its location from the sink. This dominator then selects other dominators, also based on their locations, in its circle and subsequent ones. This procedure is repeated until there are no more circles. circles. The sink then restarts the process for the next round, selecting only nodes that have not been previously selected.

   In the advanced version, whenever a Dominator 1 cannot find a neighbor in a subsequent circle, it transfers the search responsibility to the next dominator in its circle, who also tries to find. This is repeated until either a neighbor in a subsequent circle is found or the procedure ends.

It should be remembered that we initially started with the goal of energy efficiency. We intended to look for means of saving energy and extend the lifetime of the network for which we proposed the algorithms presented above. It is now upon us to see how we can manage the proposed topology schemes to save energy. It is on this premise that we make our last contribution.

D. Sleep Relay Protocol, SREP: This is a duty cycling based proposal that works on the repetitive set formation. Nodes belonging to the same set follow the same sleep schedule. They then relay the sleep (or awake) time to other sets in such a way that only one set is active at a time forming complete active path between any source and the sink. This saves energy while not deteriorating important network performance parameters like latency and packet delivery ratio.

7.2 Future Works

As we have seen, our proposals are relevant to the topologies upon which they have been tested, but we cannot guaranty their suitability in all configurations. It is therefore incumbent on us to ensure that our proposals are optimized for many more network topologies and configurations. We will therefore work towards adapting them to other topologies like ones where node deployment is not random, or with node mobility.

All the proposals that have been made in this dissertation have only been validated by simulations. Simulations provide initial appreciations for network proposals. Results obtained could be quite useful in building protocols. Nevertheless, there are certain network realities that can only be certified in a real network. It is important therefore, that our proposals be tested or real test-beds, understanding that testbeds come very close to real networks.

A closer look at the screenshots from our DVBN and DVBN+ algorithms reveal that in many cases, there are lots of disconnected nodes. This is not good for routing. As a future direction, we will look on integrating some routing protocol into our DVBN and DVBN+ solutions. This might include using parameters, like remaining energy and node degree as factors to determine which neighbor becomes a dominator.

Another possible improvement of DVBN+ can concern the optimization of the VBNs created by reducing the number of hops. As can be seen from the screenshots of Figures 23, 24, 32 and 33 of Chapter 5, the way the dominators are chosen in each circle by DVBN or DVBN+ by “going around” the circle can lead to several consecutive dominators of a circle not to have neighboring dominators in the preceding circle. This will oblige the messages relayed by these dominators to make a tour around a part of the circle, hence will increase the number of relaying nodes to the sink. A way of solving this problem is by adding a criterion on a dominator to select the next one in the circle, imposing on it to have at least one neighbor in the previous circle. It is then easy to measure the benefit of this new approach by computing the mean distance from each source to the sink with and without the use of this new criteria. However, this added selection criteria may decrease other performance metrics like the percentage disconnected nodes, hence it is perhaps necessary to balance such criteria with others.

Biography

Simon T. Obenofunde is a Ph D candidate of the Laboratoire d'Informatique de Bourgogne (LIB) of the University of Burgundy, France. He obtained an M Sc degree in communications and network engineering from U P M Serdang, Malaysia in 2004 and a B. Eng (Hons) degree in electrical engineering from the University of Ilorin, Nigeria in 1996. He is currently working on wireless sensor networks.

Simon has been into academics at different universities and levels for quite sometimes now. He just ended a 1-year ATER (Attaché Temporaire d'Enseignement et de Recherche) contract with the Unuversite Paris 8. Between 2010 and 2018 was a faculty member of the Faculty of Engineering and Technology in the University of Buea. During this time, had was concurrently a part-time lecturer with many other Universities in Cameroon. These include the Catholic University Institute of Buea, Buea and University Institute of the Gulf of Guinea, Douala.

Simon has held different management positions in many public, parapublic and private engineering institutions in Cameroon like Coordinator of Computer Engineering in the University of Buea, Deputy Director of Projects and Investments with Rural Electrification Agency of Cameroon, Soap Factory Manager with Pamol Plantaions PLC, Managing Director of Bajo Club Ltd, etc.

His research areas of interest include telecommunications, computer networking, computer systems, etc.

References

1. Abdou W., Ouled Abdallah, N. and Mosbah, M.: "ViSiDiA: a Java Framework for Designing, Simulating and Visualizing Distributed Algorithms". 2014 IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications (IEEE/ACM DS-RT 2014), Toulouse, France, October 2014.
2. Aby A.T., Guitton, A. and Misson, M. (2014) 'Asynchronous blind MAC protocol for wireless sensor networks'. Paper presented at IWCMC (International Wireless Communications and Mobile Computing Conference), 2014, Nicosia, Cyprus
3. Aguerre C., Morsellino T. and Mosbah M. "Fully-Distributed Debugging and Visualization of Distributed Systems in Anonymous Networks." 7th International Joint Conference on ComputerVision, Imaging and Computer Graphics Theory and Applications, Feb 2012, Rome, Italy. pp.764-767.
4. Akkaya K. and Younis M. "A Survey on Routing Protocols for Wireless Sensor Networks" Ad Hoc Networks 3(3):325-349, May 2005, DOI: 10.1016/j.adhoc.2003.09.010
5. Alfayez A., Hammoudeh M. and Abuarqoub, A. (2015) "A survey on MAC protocols for duty-cycled wireless sensor networks". Paper presented at The International Conference on Advanced Wireless, Information, and Communication Technologies (AWICT 2015)
6. Alkhatib A. A. A and Baicher, G. S. (2012) 'Wireless Sensor Network Architecture' Paper presented at the 2012 International Conference on Computer Networks and Communication Systems (CNCS 2012)
7. Altman E., and Jiménez T. "NS Simulator for Beginners" Morgan & Claypool, 2012.
8. Amdouni Ichrak, Adjih Cedric, Aitsaadi Nadjib and Muhlethaler Paul. "Experiments with ODYSSE:Opportunistic Duty cYcle based routing for wirelesS Sensor nEtworks". IEEE LCN 2016 -41st IEEE Conference on Local Computer Networks, Nov 2016, Dubai, United Arab Emirates.10.1109/LCN.2016.50. hal-01407525
9. Amri, S. Khelifi, F. Bradai, A. Rachedi, A. Kaddachi, M.L. and Atri, M. (2019) "A new fuzzy logic based node localization mechanism for Wireless Sensor Networks". Future Generation Computer Systems 93 (2019) 799–813
10. Anandababu, J. Siddaraju, K. and Yogish, H. (2016) 'Adaptive Duty Cycle Medium Access Control Protocol for Wireless Sensor Networks' International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 5, Issue 8, August 2016
11. Anastasi, G. Conti, M. Francesco, M. and Passarella, A. (2006) "An Adaptive and Low-latency Power Management Protocol for Wireless Sensor Networks" MobiWAC'06, October 2, 2006, Torremolinos, Malaga, Spain.
12. Anastasi G., Conti M., Francesco M. D "An Adaptive Sleep Strategy for Energy Conservation in Wireless Sensor Networks" Technical Report DII-TR-2009-03, University of Pisa,
13. Anastasi, G. Conti, M. Francesco, M.D. and Passarella A. "Energy conservation in wireless sensor networks: A survey" ELSEVIER, Ad Hoc Networks 7 (2009) 537–568
14. Anitha, K. and Usha, S. (2013) 'A Scheduled Based MAC Protocols for Wireless Sensor Network: A Survey' International Journal of Advanced Networking & Applications (IJANA) ISSN: 0975-0282
15. Arroyo-Valles R., Simonetto Andrea, Leu Geert "Consistent sensor, relay, and link selection in wireless sensor Networks". Signal Processing 140 (2017) 32–44
16. Benzerbadj A., Kechar Bouabdellah, Bounceur Ahcène, Hammoude Mohammad "Surveillance of sensitive fenced areas using duty-cycled wireless sensor networks with asymmetrical links". Journal of Network and Computer Applications 112 (2018) 41–52 Contents lists available at ScienceDirect
17. Bober, W. and Bleakley, C.J. (2014) 'BailighPulse: A low duty cycle data gathering protocol for mostly-off Wireless Sensor Networks'. Computer Networks 69 (2014) 51–65
18. Boukerche, A. and Sun, P. (2018) 'Connectivity and coverage-based protocols for wireless sensor networks'. Ad Hoc Networks 80 (2018) 54–69
19. Buettner Michael, Yee Gary V., Anderson Eric and Han Richard. "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks". Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, SenSys 2006, Boulder, Colorado, USA, October 31 - November 3, 2006
20. Camillò, A. Nati, N. Petrioli, C. Rossi, M. and Zorzi, M. (2013) 'IRIS: Integrated data gathering and interest dissemination system for wireless sensor networks. Ad Hoc Networks 11 (2013) 654–

671

21. Carlos-Mancilla M., López-Mellado E., and Siller M. “Wireless Sensor Networks Formation: Approaches and Techniques (Review Article). Hindawi Publishing Corporation. Journal of Sensors. Volume 2016, Article ID 2081902, 18 pages
22. Carrano R. C., Passos D., Magalhães L. C.S., Albuquerqu C. V.N. “A comprehensive analysis on the use of schedule-based asynchronous duty cycling in wireless sensor networks”. Ad Hoc Networks 16 (2014) 142–164
23. Chand, S., Singh, S. & Kumar, B. Heterogeneous HEED Protocol for Wireless Sensor Networks. *Wireless Pers Commun* 77, 2117–2139 (2014). https://doi.org/10.1007/s11277-014-1629-y
24. Chaudhary V., Khatoon A., Chaudhary N. “A Review of Routing Protocols for Wireless Sensor Network” International Journal of Modern Engineering Research (IJMER) | IJMER | ISSN: 2249–6645 | www.ijmer.com | Vol. 4 | Iss.9| Sept. 2014
25. Chelouah, L. Semchedine, F. and Bouallouche-Medjkoune, L. (2018) ‘Localization protocols for mobile wireless sensor networks: A survey’. Computers and Electrical Engineering 71 (2018) 733–751
26. Chen, B. Jamieson, K. Balakrishnan, H and Morris, R. (2002) “Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks” Wireless Networks Volume 8 Issue 5, September 2002 Pages 481 - 494
27. Cheng, L. Chen, C. Ma, J. and Shu, L (2012) ‘Contention-based geographic forwarding in asynchronous duty-cycled wireless sensor networks’ Int. Journal of Communications Systems. 2012; 25:1585–1602
28. da Silva R. I, Macedo D. F., Nogueir J. M. S. “Duty cycle aware spatial query processing in wireless sensor networks”. Computer Communications 36 (2013) 149–161
29. Dagdeviren O., Erciyes K., Tse S. “Semi-asynchronous and distributed weighted connected dominating set algorithms for wireless sensor networks”. Computer Standards & Interfaces 42 (2015) 143–156
30. Dargie, W and Poellabauer, C. (2010) ‘FUNDAMENTALS OF WIRELESS SENSOR NETWORKS, THEORY AND PRACTICE. 2010 John Wiley & Sons Ltd.
31. Deb B., Bhatnagar S., and Nath B., “A Topology Discovery Algorithm for Sensor Networks with applications to Network Management” Technical Report Technical Report DCS- TR-441, Department of Computer Science, Rutgers University, May 2001.
32. Díaz-Ibarra, M.A. Campos-Delgado, D.U. Gutiérrez, C.A. and Luna-Rivera, J.M (2019) “Distributed power control in mobile wireless sensor networks”. Ad Hoc Networks 85 (2019) 110–119
33. Diestel R., Graph Theory, fourth ed., in: Graduate Texts in Mathematics, vol. 173, Springer, Heidelberg, 2010
34. Dinh T., Kim Y., Gu T., Vasilako A. V. “L-MAC: A wake-up time self-learning MAC protocol for wireless sensor networks”. Computer Networks 105 (2016) 33–46
35. Dunkels, A, Gronvall, B. and Voigt T (2004) “Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors”. Paper presented at 29th Annual IEEE International Conference on Local Computer Networks, 16-18 Nov. 2004
36. El-Hoiydi Amre, Decotignie Jean-Dominique, Enz Christian and Leroux E. “WiseMAC, an ultra low power MAC protocol for the wiseNET wireless sensor network”. : Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys 2003, Los Angeles, California, USA, November 5-7, 2003. DOI: 10.1145/958491.958531
37. Elshrkawey, M. Samiha M. Elsherif, M. and Wahed, E. (2018) “An Enhancement Approach for Reducing the Energy Consumption in Wireless Sensor Networks”. Journal of King Saud University – Computer and Information Sciences 30 (2018) 259–267
38. Fu D., Han L., Liu L., Gao Q. and Feng Z. “An Efficient Centralized Algorithm for Connected Dominating Set on Wireless Networks”. The 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015)
39. Ghosh A and Givargis T. “LORD: A Localized, Reactive and Distributed Protocol for Node Scheduling in Wireless Sensor Networks” Design, Automation and Test in Europe, 2005 Conference Proceedings. DOI: 10.1109/DATE.2005.200
40. Goyal, N. Dave, M. and Verma, A.K (2017) “Data aggregation in underwater wireless sensor network: Recent approaches and issues”. Journal of King Saud University – Computer and Information Sciences (2017)
41. Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific

protocol architecture for wireless microsensor networks. In IEEE Transactions on Wireless Communications,1(4), 660-670

42. Hifi M., and Hallah R. M. “Review ArticleA Literature Review on Circle and Sphere Packing Problems: Models and Methodologies”. Hindawi Publishing Corporation Advances in Operations Research Volume 2009, Article ID 150624,22pagesdoi:10.1155/2009/150624
43. Hohlt, B. Doherty, L. and Brewer E. (2005) ‘Flexible Power Scheduling for Sensor Networks’ Paper presented at IPSN’04, April 26–27, 2004, Berkeley, California, USA.
44. Jang B, and Yoo W. “CCDC: A Congestion Control Technique for Duty Cycling WSN MAC Protocols” KSII Transactions on Internet and Information Systems 11(8):3809-3822 · August 2017
45. Jiang P., Wen Y., Shen X. and Xue A. “A Study of Routing Protocols in Wireless Sensor Networks”, in Proc. of the 6 th World Congress on Intelligent Control and Automation, vo l.1, pp266-2 70, June 2006.
46. Jiguo Y, Nannan Wang, Guanghui Wang, Dongxiao Yu “Connected dominating sets in wireless ad hoc and sensor networks – A comprehensive survey”. ELSEVIER Computer Communications 36 (2013) 121–134
47. Jing Li and Georgios Y. Lazarou “A Bit-Map-Assisted Energy-Efficient MAC Scheme for Wireless Sensor Networks”. IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks. April 2004 Pages 55–60. https://doi.org/10.1145/984622.984631
48. Karray, F. Jmal, M. W. Garcia-Ortiz, A. and Obeid, M.A.A.M. (2018) “A comprehensive survey on wireless sensor node hardware platforms”, Computer Networks 144 (2018) 89–110
49. Khan, J. A. Qureshi, H. K. and Iqbal A. (2015) ‘Energy management in Wireless Sensor Networks: survey’. Computers and Electrical Engineering, Elsevier, 2014, Volume 41, January 2015, pp. 159-176
50. Khatarkar S. and Kamble R. “Wireless Sensor Network MAC Protocol: SMAC & TMAC” Indian Journal of Computer Science and Engineering (IJCSE), Vol. 4 No.4 Aug-Sep 2013
51. Khemapech, I. Duncan, I. and Miller, A. (2014) “A Survey of Wireless Sensor Networks Technology”. In Proceedings of the 6th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, 2005
52. Kim D. and Liu M. “Optimal Stochastic Routing in Low Duty-Cycled Wireless Sensor Networks” Internet Mathematics Vol. 9, 2008. Nos. 2–3: 161–198
53. Krishna, K. H. Babu, Y. S. Kuma, T. (2016) ‘Wireless Network Topological Routing In Wireless Sensor Networks’ 7th International Conference on Communication, Computing and Virtualization 2016
54. Kulik J., Heinzelman W., and Balakrishnan H., "Negotiation-based protocols for disseminating information in wireless sensor networks", Wireless Networks, vol. 8, no. 2/3, Mar.-May 2002, pp. 169-185
55. Lee D. and Chung K. “A Congestion Control Scheme Using Duty Cycle Adjustment in Wireless Sensor Networks,”Journal of KICS, Vol.35, No.1, pp.154-161, 2010 (in Korean).
56. Lee D. and Chung K., "Adaptive duty-cycle based congestion control for home automation networks," in *IEEE Transactions on Consumer Electronics*, vol. 56, no. 1, pp. 42-47, February 2010, doi: 10.1109/TCE.2010.5439124.
57. Levis, P. Madden, S. Polastre, J. Szewczyk, R. Whitehouse, K. Woo, A. Gay, D. Hill, J. Welsh, M. Brewer, E and Culler, D. (2005) ‘TinyOS: An Operating System for Sensor Networks’ [online] https://link.springer.com/chapter/10.1007/3-540-27139-2_7 (Accessed 01 April 2019)
58. Lindsey S. and Raghavendra C.S., “PEGASIS: Power-efficient Gathering in Sensor Information System”, Proceedings IEEE Aerospace Conference, vol. 3, Big Sky, MT, Mar. 2002, pp. 1125-1130.
59. Lingaraj, K. Biradar, R.V. and Patil, V.C. (2018) ‘Eagilla: An Enhanced Mobile Agent Middleware for Wireless Sensor Networks’. Alexandria Engineering Journal (2018) 57, 1197–1204
60. Liu A., Chen ZhuangbinXion , Neal N. “An adaptive virtual relaying set scheme for loss-and-delay sensitive WSNs”. Information Sciences 424 (2018) 118–136
61. Liu G. and Yao G. “SRMAC: Staggered Routing-Enhanced MAC Protocol for Wireless Sensor Networks”. 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing
62. Mabrouki Issam, Lagrange Xavier, Froc Gwillerm. Random Walk Based Routing Protocol for WirelessSensor Networks. ACM International Conference Proceeding Series, Proceedings of the 2nd Interna-tional Conference on Performance Evaluation Methodologies and Tools, Oct 2007, Nantes, France.hal-02164541
63. Mehta, S. and Kwak, K.S. (2010) ‘H-MAC: A hybrid MAC protocol for wireless sensor networks’

International journal of computer networks and communications (IJCNC) Vol 2 No 2, March 2010

64. Merzoug, M.A. Boukerche, A. Mostefaoui, A. and Chouali, S. (2019) “Spreading Aggregation: A distributed collision-free approach for data aggregation in large-scale wireless sensor networks.” Journal of Parallel and Distributed Computing 125 (2019) 121–134
65. Mihaylov M, Le Borgne Yann-Ael, Tuyls Karl, and Nowe Ann “Reinforcement Learning for Self-Organizing Wake-Up Scheduling inWireless Sensor Networks”. International Conference on Agents and Artificial Intelligence, ICAART 2011: Agents and Artificial Intelligence pp 382-396 |
66. Mohanty J. P., Mandal C., Reade C., Das A. “Construction of minimum connected dominating set in wireless sensor networks using pseudo dominating set”. Ad Hoc Networks 42 (2016) 61–73
67. Mohanty, J. P., Mandal C., Read C. “Distributed construction of minimum Connected Dominating Set in wireless sensor network using two-hop information”. Computer Networks 123 (2017) 137–152
68. Mothku S. K. and Rout R. R. “Fuzzy logic based adaptive duty cycling for sustainability in energy harvesting sensor actor networks”. Journal of King Saud University – Computer and Information Sciences xxx (2018) xxx–xxx
69. Oliveira C. H. S. and de Castro Miguel F. “Autonomic Duty Cycling for Target Tracking in a Bio-Inspired Wireless Sensor Network”. ICONS 2015: The Tenth International Conference on Systems
70. Panag T. S., J. S. Dhillon “Heuristic search algorithm (HSA) for enhancing the lifetime of wireless sensor networks” International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:9, No:8, 2015
71. Parvin S. and Rahim M. S. “Routing Protocols for Wireless Sensor Networks: A Comparative Study”. International Conference on Electronics, Computer and Communication (ICECC 2008), University of Rajshahi, Bangladesh. ISBN 984-300-002131-3
72. Penumalli C. and Palanichamy Y. “An Optimal CDS Construction Algorithm with Activity Scheduling in Ad Hoc Networks”. Hindawi Publishing Corporation. The Scientific World Journal Volume 2015, Article ID 842346, 12 pages
73. Polastre J., Hill J., and Culler D., “Versatile low power mediaaccess for wireless sensor networks,” in proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys ’04), pp. 95–107, ACM Press, November 2004.
74. Qureshi H. K., Rizvi Sajjad, Saleem Muhammad. Khayam Syed Ali, Rakocevic Veselin and Rajarajan Muttukrishnan “Evaluation and improvement of CDS-based topology control for wireless sensor networks”. Wireless Netw (2013) 19:31–46
75. Raman, S. (2002) ‘TinyOS – An Operating System for Tiny Embedded Networked Sensors’. [online] http://crystal.uta.edu/~kumar/cse6306/papers/TinyOS_p aper.pdf (Accessed 01 April 2019)
76. Rault, T. Bouabdallah, A. Challal, Y. (2014) ‘Energy efficiency in wireless sensor networks: A top-down survey’. Computer Networks 67 (2014) 104–122
77. Renold A. Pravin and Chandrakal S. “Survey on state scheduling-based topology control in unattended wireless sensor networks” Computers and Electrical Engineering 56 (2016) 334–349
78. Rezaei, Z. and Mobininejad, S. (2012) ‘Energy Saving in Wireless Sensor Networks’ International Journal of Computer Science & Engineering Survey (IJCSES) Vol.3, No.1, February 2012
79. Rhee, I. Warrier, A.Min, J. and and Xu, L. (2006) ‘DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks,’ Proceeding of ACM MobiHoc, May 2006.
80. Ringwald M. and Romer K. “Deployment of Sensor Networks: Problems and Passive Inspection”. Published in 2007 Fifth Workshop on Intelligent Solutions in Embedded Systems
81. Roychowdhury, S. and Patra, C. (2010) “Geographic Adaptive Fidelity and Geographic Energy Aware Routing in Ad Hoc Routing”. Special Issue of IJCCT Vol.1 Issue 2, 3, 4; 2010 for International Conference [ACCTA-2010], 3-5 August 2010
82. Samant, T. and Datta, A. (2016) ‘Analysis and Comparison of SMAC and TMAC Protocol for Energy Efficient Dynamic Topology in Sensor Network’ International Journal of Electrical and Computer Engineering (IJECE) Vol. 6, No. 5, October 2016, pp. 2331~2337
83. Schurgers C., Tsiatsis V. and Srivastava Mani B. “STEM: Topology management for energy efficient sensor networks”. Aerospace Conference Proceedings, 2002. IEEE Volume: 3. February 2002. DOI: 10.1109/AERO.2002.1035239
84. Sha M., Hackmann G. and Lu C., "Energy-efficient Low Power Listening for wireless sensor networks in noisy environments," 2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Philadelphia, PA, 2013, pp. 277-288, doi: 10.1145/2461381.2461415.

85. Sharma, G. and Kumar, A. (2018) "Improved range-free localization for three-dimensional wireless sensor networks using genetic algorithm". Computers and Electrical Engineering 72 (2018) 808–827
86. Sharma V., Patel R.B., Bhadauria H.S., Prasad D. "Deployment schemes in wireless sensor network to achieve blanket coverage in large-scale open area: A review". Egyptian Informatics Journal (2016) 17,4 5–56
87. Shi B., Sreeram V., Zhao D., Duan S. and Jiang J. (2018) "A wireless sensor network-based monitoring system for freshwater fishpond aquaculture" Biosystems engineering 172 (2018) Pages 57-66
88. Singh S. K, Singh M P, and Singh D K "Routing Protocols in Wireless Sensor Networks – A Survey" International Journal of Computer Science & Engineering Survey (IJCSES) Vol.1, No.2, November 2010, DOI : 10.5121/ijcses.2010.1206 63
89. Singh R., Rai B. K. and Bose S. K. "A Joint Routing and MAC Protocol for Transmission Delay Reduction in Many-to-One Communication Paradigm for Wireless Sensor Networks". IEEE Internet of Things Journal (Volume: 4 , Issue: 4 , Aug. 2017)
90. Singh S. P., Sharma S C "A survey on Cluster Based Routing Protocol in Wireless Sensor Networks" International Conference on Advanced Computing Technologies and Applications (ICACTA-2015).
91. Skiadopoulos, K. Tsipis, A. Giannakis, K. Koufoudakis, G. Christopoulou, E. Oikonomou, K. Kormentzas, G. and Stavrakakis, I. (2019) 'Synchronisation of Data Measurements in Wireless Sensor Networks for IoT Applications'. Ad Hoc Networks (2019), 2019.03.002
92. Slama Ines, Shrestha Bharat, Jouaber Badii, Zeghlache Djamal and Erke Tapio J. "DNIB: Distributed Neighborhood Information Based TDMA Scheduling for Wireless Sensor Networks". Vehicular Technology Conference, 1988, IEEE 38th. DOI: 10.1109/VETECF.2008.25
93. Slijepcevic S., Potkonjak M. "Power Efficient Organization of Wireless Sensor Networks". ICC 2001. IEEE International Conference on Communications.
94. Stathopoulos, T. Lukac, M. Heidemann, J. Estrin, D. McIntire, D. and Kaiser, W. J. (2007) 'End-to-end Routing for Dual-Radio Sensor Networks' Paper presented at IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, 6-12 May 2007
95. Suhonen J., Hamalainen Timo D and Hannikainen Marko "Availability and End-to-end Reliability in low Duty Cycle Multihop Wireless Sensor Networks". Sensors 2009, 9, 2088-2116; oi:10.3390/s90302088
96. Surendran S., Vijayan S. "Distributed Computation of Connected Dominating Set for Multihop Wireless Networks". Seventh International Symposium on Applications of Ad hoc and Sensor Networks (AASNET'15)
97. Sun Yanjun, Gurewitz Omer and Johnson David B. "RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks". Proceedings of the 6th International Conference on Embedded Networked Sensor Systems, SenSys 2008, Raleigh, NC, USA, November 5-7, 2008. DOI: 10.1145/1460412.1460414
98. Thaskani, S., Kumar, K.V. and Murthy, G.R. "Mobility tolerant TDMA based MAC protocol for WSN". Computers & Informatics, 2011, Pages 515-519.
99. Tong F, Ni M, Shu L and Pan J. "A Pipelined-forwarding, Routing-integrated and effectively-Identifying MAC for large-scale WSN". 2013 IEEE Global Communications Conference (GLOBECOM). DOI: 10.1109/GLOCOM.2013.6831075
100. Tong, F. Tang, W. Xie, R. Shu, L. and Kim, Y. (2011) 'P-MAC: A Cross-Layer Duty Cycle MAC Protocol Towards Pipelining for Wireless Sensor Networks' Paper presented at IEEE International Conference on Communications, ICC 2011, Kyoto, Japan, 5-9 June, 2011
101. Tseng Y.C., Ni S.Y., Chen Y.S., Sheu J.P. The broadcast storm problem in a mobile ad hoc network. Wirel. Netw. 2002;8:153–167. doi: 10.1023/A:1013763825347.
102. Tsoumanis, G. Oikonomou, K. Aïssa, S. and Stavrakakis, I. (2018) 'A recharging distance analysis for wireless sensor networks' Ad Hoc Networks 75–76 (2018) 80–86
103. Valera, A.C. Soh, W. and Tan, H. (2014) "Survey on wakeup scheduling for environmentally-powered wireless sensor networks". Computer Communications 52 (2014) 21–36
104. van Dam T. and Langendoen K.. An adaptive energy-efficient mac protocol for wireless sensor networks. In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems, Nov. 2003.
105. Vazirani V. V., *Approximation Algorithms*, Springer-Verlag Berlin Heidelberg, 2003
106. Verma, N. and Singh, D. (2018) "Data Redundancy Implications in Wireless Sensor Networks".

Paper presented at International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

107. Wang Y., Henning I. D., Li X. and Hunter D. “SOTP: A self-organized TDMA protocol for wireless sensor networks”. Proceedings of the Canadian Conference on Electrical and Computer Engineering, CCECE 2006, May 7-10, 2006, Ottawa Congress Centre, Ottawa, Canada. DOI: 10.1109/CCECE.2006.277307
108. Wu, L. Zhuo, S. Wang, Z. and Wang, Z. (2015) ‘pQueue-MAC: An Energy Efficient Hybrid MAC Protocol for Event-Driven Sensor Networks’. International Journal of Distributed Sensor Networks Volume 2015, Article ID 160167, 11 pages
109. Wu Shaobo, Niu Jianwei, Chou Wusheng and Guizani Mohsen “Delay-Aware Energy Optimization for Flooding in Duty-Cycled Wireless Sensor Networks”. IEEE Transactions on Wireless Communications (Volume: 15, Issue: 12, Dec. 2016). DOI: 10.1109/TWC.2016.2615296
110. XU J., WANG L., MA C., SHU L. “Study of impacts of duty-cycle on overlapping multi-hop clustering in wireless sensor networks”. The Journal of China Universities of Posts and Telecommunications, October 2012, 19 (Suppl. 2): 19–22
111. Ye W., Heidemann J, Estrin D, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks”,n INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2002, Volume: 3, Page(s):1567 - 1576 vol.3
112. Zaparanuks D., and Hauswirth M. “Algorithmic profiling”. *Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation,* June 2012 Pages 67–76.
113. Zorzi M., Rao Ramesh R. “Geographic Random Forwarding (GeRaF) for adhoc and sensor networks: multihop performance”. IEEE Transactions on Mobile Computing, vol. 2, n. 4, Oct.-Dec. 2003