

N° d'ordre : 02/2017 - D/INF.

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique

Université des Sciences et de la Technologie Houari Boumediene
Faculté d'Electronique et Informatique

THÈSE

Présentée pour l'obtention du grade de

DOCTEUR EN SCIENCE

En : INFORMATIQUE

Spécialité : Programmation et Systèmes

par

Bahia ZEBBANE

Sujet

*Election coopérative des coordinateurs dans les réseaux
de capteurs : le contrôle de topologie pour la
conservation d'énergie dans les réseaux de capteurs*

Soutenue publiquement le 13/03/2017, devant le Jury composé de :

| | | | |
|-------------------|------------|------------|--------------------|
| Mme. S. MOUSSAOUI | Professeur | USTHB | Présidente |
| Mr. N. BADACHE | Professeur | USTHB | Directeur de thèse |
| Mr. H. HAFFAF | Professeur | Univ. ORAN | Examineur |
| Mr. M. AISSANI | MCA | EMP | Examineur |
| Mr. Y. CHALLAL | MCA | ESI | Examineur |
| Mr. M. R. SENOUCI | MCA | EMP | Examineur |

PEOPLES DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
University of Science and Technology Houari Boumediene
Faculty of Electrical Engineering and Computer Science

THESIS

Presented to obtain the

DOCTORAL DEGREE

In : COMPUTER SCIENCE

Speciality : Programming and Systems

by

Bahia ZEBBANE

Subject

*Coordinators cooperative Election in Sensor Networks:
Topology Control for energy conservation in sensor
networks*

Publicly defended on 13/03/2017, in front of the Jury:

| | | | | |
|------|---------------|-----------|------------|-------------------|
| Mrs. | S. MOUSSAOUI | Professor | USTHB | President |
| Mr. | N. BADACHE | Professor | USTHB | Thesis Supervisor |
| Mr. | H. HAFFAF | Professor | Univ. ORAN | Reviewer |
| Mr. | M. AISSANI | MCA | EMP | Reviewer |
| Mr. | Y. CHALLAL | MCA | ESI | Reviewer |
| Mr. | M. R. SENOUCI | MCA | EMP | Reviewer |

Résumé

La technologie des réseaux de capteurs sans fil (RCSFs) a la capacité de capturer, traiter et transmettre de données critiques en temps réel. Cette technologie a permis de surveiller des zones hostiles, inaccessibles et éloignées. Elle répond à un besoin très important pour une surveillance en temps réel, en particulier dans les scénarios dangereux. Cependant, l'un des défis critiques des réseaux de capteurs sans fil est la conservation d'énergie. En effet, les RCSFs sont constitués d'un grand nombre de noeuds de capteurs déployés de manière aléatoire, et dans de nombreux cas, il est impossible de remplacer les capteurs en cas de défaillance. Ainsi, les applications ont tendance à déployer plus de noeuds que nécessaire pour faire face aux pertes possibles des noeuds et augmenter la durée de vie du réseau, ce qui conduit à créer une certaine redondance de captage et de communication. Toutefois, les capteurs dans la même région, peuvent recueillir et transmettre les mêmes données, ce qui fera perdre plus d'énergie.

Le contrôle de topologie est considéré comme l'une des techniques les plus importantes, utilisées dans les réseaux ad hoc et capteurs sans fil pour réduire la consommation d'énergie et les interférences radio. Il vise à exploiter la densité du réseau pour économiser l'énergie et prolonger la durée de vie du réseau en gardant seulement un petit ensemble de noeuds à l'état actif et désactiver ceux redondants. Ceci peut être accompli en adoptant une approche de planification de sommeil/éveil, sans sacrifier les fonctionnalités du réseau.

Plusieurs algorithmes de contrôle de topologie utilisant la technique d'ordonnement, ont été proposés dans la littérature. L'idée principale de ces protocoles est de gérer la topologie en identifiant les noeuds redondants et les faire basculer entre les modes actifs et sommeil afin de réduire la consommation d'énergie. Ils déterminent combien et quels noeuds devraient être autorisés à dormir, dans le but d'assurer la connectivité. Cependant, lors de l'exécution de ces algorithmes, le nombre de noeuds actifs n'est pas toujours le minimum et parfois la topologie formée est déconnectée.

Il existe un compromis entre la consommation d'énergie, la connectivité et le nombre de noeuds actifs. Si le nombre de noeuds actifs est minimisé, la connectivité réseau peut être perdue; Et si la connectivité est assurée lorsqu'un nombre acceptable de noeuds actifs est atteint, une énergie supplémentaire est dépensée.

En outre, une grande surcharge de communication sera produite au cours la recherche des noeuds redondants ce qui conduit à une grande consommation d'énergie. En outre, peu de travaux existants prennent en considération l'énergie gaspillée lors des transitions d'état dans leur stratégie de planification.

Dans ce travail, nous proposons, d'abord, une taxonomie des protocoles de contrôle de topologie basés ordonnancement. Nous les classifions en quatre catégories: protocoles plats, protocoles basés grilles, protocoles basés clusters et protocoles basés groupes. Deuxièmement, nous proposons un nouveau algorithme basé grille, appelé GTC: a Geographical Topology Control protocol, qui bénéficie des avantages des algorithmes existants basés grilles, pour identifier les noeuds redondants. GTC utilise une nouvelle stratégie de planification qui réduit le nombre de transitions entre les états actifs et sommeils afin de minimiser l'énergie de transition et assurer la longévité du réseau. Il minimise également la fréquence d'élection des noeuds actifs et permet une certaine stabilité dans la topologie qui évitera la perte de connectivité.

La deuxième contribution examine la catégorie de contrôle de topologie basée groupes, car elle utilise uniquement des informations de voisinage pour regrouper les noeuds redondants ensemble. Ainsi, nous proposons a Group-based Energy-Conserving Protocol (GECP) qui exploite la redondance des capteurs dans la même région en divisant le réseau en groupes afin qu'une topologie connectée peut être maintenue en gardant un seul noeud actif dans chaque groupe et en éteignant ceux redondants. GECP identifie les noeuds redondants, les organise en groupes avec une faible surcharge de communication. Il utilise la même stratégie d'ordonnancement que GTC qui vise à réduire le nombre de transitions entre les états actifs et sommeils et, par conséquent, de prolonger la durée de vie du réseau.

Dans la troisième contribution, un nouveau protocole distribué appelé RTCP (Redundancy aware Topology Control Protocol) est proposé et qui utilise seulement la liste des voisins d'un noeud comme métrique pour définir la redondance de communication, et ce pour éviter toute hypothèse sur la régularité de communication. RTCP identifie les noeuds redondants et les organise en groupes avec une faible surcharge de communication. Les noeuds redondants qui satisfont certaines règles d'éligibilité, peuvent former un groupe en fonction de leurs degrés de redondance, ce qui réduit le nombre de groupes sans perdre la connectivité du réseau. RTCP définit un seuil de niveau de connectivité qui permet aux applications de paramétrer le degré de connectivité souhaité dans la topologie réduite.

La dernière contribution consiste en l'amélioration de RTCP pour permettre à chaque noeud capteur de déterminer dynamiquement la valeur du seuil du niveau de connectivité. Le protocole proposé, appelé ERTCP: Enhanced Redundancy aware Topology Control Protocol, utilise une nouvelle métrique pour déterminer les noeuds équivalents.

Abstract

Wireless sensor network (WSN) technology has the capability of quick capturing, processing, and transmitting critical data in real-time. This technology has made possible to monitor hostile, inaccessible and remote areas. It fulfills a very important need for any real-time monitoring, especially in hazardous scenarios. However, one of the critical challenges in wireless sensor network is the energy conservation. Indeed, WSNs consists of a large number of sensor nodes randomly deployed, and, in many cases, it is impossible to replace sensors when a node failure occurs. Thus, applications tend to deploy more nodes than necessary to cope with possible node failures and to increase the network lifetime, which leads to create some sensing and communication redundancy. Nevertheless, sensors in the same region, may collect and forward the same information, which will waste more energy. Topology control is considered one of the most important techniques used in wireless ad hoc and sensor networks to reduce energy consumption and radio interference. It aims at exploiting the network density to conserve energy and extend the network lifetime by keeping a small set of nodes in active state and turn off the redundant ones. This can be accomplished by adopting a sleep/wake scheduling approach, without sacrificing the network functionalities.

Several topology control algorithms, using the scheduling technique, have been proposed in the literature. The key idea of these protocols is to manage the topology by identifying redundant nodes and schedule nodes for active or sleep modes in order to reduce energy consumption. They determine how many and which nodes should be allowed to sleep, with the purpose to ensure connectivity. However, when running these algorithms, the number of the active nodes is not always the minimum and sometimes the formed topology is not connected. There is a trade-off between power consumption, connectivity and the number of active nodes. If the number of active nodes is minimized, the network connectivity may be lost; And if connectivity is ensured when an acceptable number of active nodes is reached, an additional energy is expended. Furthermore, a large communication overhead will be produced during the search of the redundant nodes which leads to large energy consumption. Moreover, few existing works take into consideration the energy wasted during transitions in their scheduling strategy.

In this work, we first, propose a taxonomy of sleep scheduling based-topology control

protocols. We classify the topology control protocols based duty cycling into four categories: Flat protocols, Grid-based protocols, Cluster-based protocols and Group-based protocols. Second, we propose a new grid-based algorithm, called GTC: a Geographical Topology Control protocol, that benefits from the advantages of the existing grid-based algorithms to identify redundant nodes. GTC uses a new scheduling strategy which reduces the number of transitions between active and sleep states in order to minimize the transition energy and ensure the network longevity. It also minimizes the frequency of active nodes election and allows some stability in the topology that avoids the loss of connectivity.

The second contribution investigates the group-based topology control category, it uses only neighborhood information to group redundant nodes together. So, we propose a Group-based Energy-Conserving Protocol (GECP) which exploits the sensor redundancy in the same region by dividing the network into groups so that a connected backbone can be maintained by keeping only one active node in each group and turning off the redundant ones. GECP identifies redundant nodes, organizes them into groups with a low communication overhead. It uses the same scheduling strategy as GTC which aims to minimize the number of transitions between active and sleep states and, hence, extend the network lifetime.

In the third contribution, a new distributed Redundancy-aware Topology Control Protocol (RTCP) is proposed which uses only the neighbor set as the communication redundancy metric, to avoid any assumptions about communication regularity. RTCP identifies redundant nodes and organizes them into groups with a low communication overhead. The redundant nodes that satisfy some eligibility rules, can form group according to their redundancy degrees, which leads to a reduced number of groups without lose of network connectivity. RTCP defines a Threshold of connectivity level which allows applications to parameter the desired connectivity degree in the reduced topology.

The last contribution consists on the improvement of RTCP to allow to each sensor node to determine dynamically the value of the Threshold of connectivity level. The proposed protocol, called ERTCP: Enhanced Redundancy aware Topology Control Protocol uses a new metric to determine the equivalent nodes.

Acknowledgement

I thank the Almighty ALLAH, for giving me the ability, strength, health, and perseverance necessary to complete this work. All praise is due to Allah, the Lord of the Worlds. Peace and blessings be upon our prophet Mohammed.

I had the pleasure and honor to receive support from many wonderful people to whom I would like to express my deepest gratitude.

First of all, I would like to gratefully thank Prof. Nadjib BADACHE, my supervisor for his support, guidance, and encouragement throughout these years.

A very special thanks to my dear friend Chafika BENZAID, for her valuable help, for reviewing my papers and thesis.

I wish to deeply thank Prof. Nacera BENSOU for her precious help through complexity analysis.

I would like to express my gratitude to Prof. Samira MOUSSAOUI, Prof. Hafid HAFFAF, Dr. Mohamed AISSANI, Dr. Yacine CHALLAL, and Dr. Mustapha Redha SENOUCI my committee members, for their kind willingness to judge this work.

Many thanks are due to my family, my husband, as well as my friends (you know who you are) for their tremendous love, support and encouragement throughout all my thesis. These have been the driving force behind my accomplishments.

Special thanks to my dear mother, my reason for living, for her love, encouragement and constant support.

Contents

| | |
|--|------------|
| Résumé | i |
| Abstract | iii |
| Acknowledgement | v |
| 1 Introduction | 1 |
| 1.1 Problem Statement and Motivations | 1 |
| 1.2 Contributions | 3 |
| 1.3 Thesis Organization | 4 |
| 2 Energy conservation in wireless sensor networks | 7 |
| 2.1 Introduction | 7 |
| 2.2 Definitions | 7 |
| 2.2.1 Coverage | 7 |
| 2.2.2 Connectivity | 8 |
| 2.2.3 Network lifetime | 8 |
| 2.3 Energy consumption of wireless nodes | 9 |
| 2.3.1 Wireless node states | 9 |
| 2.3.2 Basic energy consumption | 9 |
| 2.3.3 Sources of energy waste | 10 |
| 2.4 Energy saving techniques in WSNs | 10 |
| 2.4.1 Energy harvesting | 12 |
| 2.4.2 Energy transferring | 12 |
| 2.4.3 Duty cycling | 13 |
| 2.4.4 Energy-efficient routing | 14 |
| 2.4.5 Data reduction | 16 |
| 2.4.6 Power control | 17 |
| 2.4.7 Node mobility | 18 |
| 2.4.8 Discussion | 19 |
| 2.5 Conclusion | 20 |

| | | |
|----------|---|-----------|
| 3 | Topology control in wireless sensor networks | 23 |
| 3.1 | Introduction | 23 |
| 3.2 | Definitions | 23 |
| 3.2.1 | Topology of the network | 23 |
| 3.2.2 | Topology Control | 24 |
| 3.3 | Motivations for Topology Control | 24 |
| 3.3.1 | Maximizing network lifetime | 25 |
| 3.3.2 | Increasing network capacity | 25 |
| 3.3.3 | Ensure coverage and connectivity | 26 |
| 3.4 | Topology control mechanisms | 27 |
| 3.4.1 | Power control | 27 |
| 3.4.2 | Sleep-scheduling | 28 |
| 3.5 | Need for sleep scheduling | 29 |
| 3.6 | Sleep scheduling-based topology control protocols | 30 |
| 3.6.1 | Flat protocols | 31 |
| 3.6.2 | Grid-based protocols | 37 |
| 3.6.3 | Cluster-based protocols | 39 |
| 3.6.4 | Group-based protocols | 43 |
| 3.7 | Discussion | 46 |
| 3.8 | Conclusion | 51 |
| | | |
| 4 | GTC: A Geographical Topology Control Protocol | 53 |
| 4.1 | Introduction | 53 |
| 4.2 | Motivation | 54 |
| 4.3 | GTC design | 54 |
| 4.3.1 | GTC overview | 55 |
| 4.3.2 | GTC state transitions | 56 |
| 4.3.3 | GTC parameters tuning | 58 |
| 4.3.4 | Scheduling in GTC | 59 |
| 4.3.5 | GTC analysis | 61 |
| 4.4 | Performance evaluation | 62 |
| 4.4.1 | Network lifetime | 64 |
| 4.4.2 | Data delivery rate | 64 |
| 4.4.3 | Scalability | 65 |
| 4.5 | Conclusion | 66 |
| | | |
| 5 | Group-based Energy-Conserving Protocol (GECP) | 69 |
| 5.1 | Introduction | 69 |
| 5.2 | Network model | 69 |
| 5.3 | GECP design | 71 |

| | | |
|----------|---|------------|
| 5.3.1 | Discovery phase | 71 |
| 5.3.2 | Group forming phase | 71 |
| 5.3.3 | Leaders election phase | 73 |
| 5.3.4 | GECP parameters tuning | 75 |
| 5.4 | Complexity analysis | 76 |
| 5.4.1 | Time complexity | 76 |
| 5.4.2 | Communication complexity | 77 |
| 5.5 | Performance evaluation | 77 |
| 5.5.1 | Group forming phase | 78 |
| 5.5.2 | Sleep/wake scheduling phase | 81 |
| 5.6 | Conclusion | 88 |
| 6 | RTCP: a Redundancy aware Topology Control Protocol | 91 |
| 6.1 | Introduction | 91 |
| 6.2 | Network model | 92 |
| 6.3 | RTCP approach | 93 |
| 6.3.1 | Neighbor Discovery phase | 93 |
| 6.3.2 | Group Forming phase | 93 |
| 6.3.3 | Leaders Election phase | 102 |
| 6.3.4 | RTCP complexity analysis | 103 |
| 6.3.5 | Communication complexity | 103 |
| 6.3.6 | Time complexity | 104 |
| 6.4 | Performance evaluation | 105 |
| 6.4.1 | Number of groups formed | 106 |
| 6.4.2 | Energy consumed in groups forming phase | 106 |
| 6.4.3 | Network lifetime | 107 |
| 6.4.4 | Network connectivity | 112 |
| 6.5 | Conclusion | 112 |
| 7 | Enhanced Redundancy aware Topology Control Protocol (ERTCP) | 117 |
| 7.1 | Introduction | 117 |
| 7.2 | Definition | 117 |
| 7.3 | Enhanced Redundancy aware Topology Control Protocol | 118 |
| 7.3.1 | Redundant nodes identification phase | 118 |
| 7.3.2 | Active nodes election phase | 122 |
| 7.4 | Performance evaluation | 122 |
| 7.4.1 | Number of groups formed | 123 |
| 7.4.2 | Energy consumed in the redundant nodes identification phase | 123 |
| 7.4.3 | Network lifetime | 124 |
| 7.4.4 | Network connectivity | 127 |

| | |
|-------------------------------|------------|
| 7.5 Conclusion | 129 |
| 8 Conclusion | 131 |
| 9 List of Publications | 135 |

List of Figures

| | | |
|------|---|----|
| 2.1 | Taxonomy of energy conservation approaches in WSNs. | 11 |
| 3.1 | Example of a network topology (a) without topology control (b) with topology control. | 26 |
| 3.2 | Topology control categories. | 27 |
| 3.3 | (a) The initial network topology represented by a communication graph. (b) The reduced topology under power control. (c) The reduced topology under sleep scheduling. | 29 |
| 3.4 | Classification of topology control protocols based on sleep scheduling. . | 31 |
| 3.5 | SPAN state transitions. | 32 |
| 3.6 | ASCENT state transitions. | 35 |
| 3.7 | OTC state transitions. | 36 |
| 3.8 | GAF state transition diagram. | 38 |
| 3.9 | TECA state transitions. | 40 |
| 3.10 | ECTC state transitions. | 42 |
| 3.11 | CPA state transitions. | 45 |
| 4.1 | Zones formation. | 55 |
| 4.2 | GTC state transitions. | 57 |
| 4.3 | The flowchart of election phase. | 57 |
| 4.4 | The scheduling policy used by GTC with connectivity gain. | 61 |
| 4.5 | Network lifetime. | 65 |
| 4.6 | Data delivery rate in the network. | 66 |
| 4.7 | The node density influence on the network lifetime. | 67 |
| 4.8 | Transition number Vs Node density. | 67 |
| 5.1 | (a) The initial network topology represented by a communication graph. (b) The network topology after the group forming phase. | 73 |
| 5.2 | GECP state transitions. | 75 |
| 5.3 | CPA group merging process. (a) Initial partition. (b) First iteration. . . | 79 |
| 5.4 | Consumed energy VS node density. | 80 |
| 5.5 | Network Lifetime for Density D1. | 82 |

| | | |
|------|---|-----|
| 5.6 | Network Lifetime for Density D2. | 83 |
| 5.7 | Network Lifetime for Density D3. | 84 |
| 5.8 | Network Lifetime for Density D4. | 85 |
| 5.9 | Number of transitions VS Node density. | 86 |
| 5.10 | Number of Sleeping nodes VS time. | 87 |
| 5.11 | The network graphs of no Topology Control, GECP and CPA algorithms at 65s. | 88 |
| 6.1 | (a) The initial network topology (Topo1) represented by a communication graph. (b) The resulting topology when $T_{cl} = 80\%$ with connectivity guarantee. (c) The resulting topology when $T_{cl} = 70\%$ with connectivity loss. | 95 |
| 6.2 | (a) The initial network topology (Topo2). (b) The network topology after the group forming phase when $T_{cl} = 60\%$ with connectivity guarantee. | 99 |
| 6.3 | RTCP state diagram transitions | 103 |
| 6.4 | Consumed energy VS Node density. | 108 |
| 6.5 | Fraction of survived nodes VS time for Node density D1. | 109 |
| 6.6 | Fraction of survived nodes VS time for Node density D2 when $T_{cl} = 50\%$ | 109 |
| 6.7 | Fraction of survived nodes VS time for Node density D2 when $T_{cl} = 60\%$ | 110 |
| 6.8 | Fraction of survived nodes VS time for Node density D2 when $T_{cl} = 70\%$ | 110 |
| 6.9 | Fraction of survived nodes VS time for Node density D2 when $T_{cl} = 80\%$ and $T_{cl} = 90\%$ | 111 |
| 6.10 | Fraction of survived nodes VS time for Node density D3. | 111 |
| 6.11 | Fraction of survived nodes VS time for Node density D4. | 112 |
| 6.12 | The network graphs of no Topology Control and RTCP algorithm at 65s for node density D2. | 114 |
| 6.13 | The network graphs of no Topology Control and GECP algorithm at 65s for node density D2. | 115 |
| 6.14 | The network graphs of no Topology Control and GAF algorithm at 65s for node density D2. | 115 |
| 6.15 | The network graphs of no Topology Control and CPA algorithm at 65s for node density D2. | 116 |
| 7.1 | (a) The initial network topology represented by a communication graph. (b) The resulting topology, after round 2, with connectivity guarantee. | 121 |
| 7.2 | The round robin strategy used by ERTCP in the active node election phase. | 122 |
| 7.3 | Consumed energy VS node density. | 125 |
| 7.4 | Fraction of survived nodes VS time for Density D1. | 125 |
| 7.5 | Fraction of survived nodes VS time for Density D2. | 126 |

| | | |
|-----|---|-----|
| 7.6 | Fraction of survived nodes VS time for Density D3. | 126 |
| 7.7 | Fraction of survived nodes VS time for Density D4. | 127 |
| 7.8 | The network graphs of no Topology Control, ERTCP and RTCP algorithms at 65s for D2. | 128 |

List of Tables

| | | |
|-----|---|-----|
| 2.1 | Typical values of energy costs relevant to various cards. | 9 |
| 2.2 | Objectives of energy conservation techniques: Network requirements and QoS metrics. | 21 |
| 3.1 | Output power settings and typical current consumption of CC2420. . . . | 28 |
| 3.2 | Energy consumptions of some IEEE802.15.4-compatible radios. | 29 |
| 3.3 | Comparative study of topology control algorithms based on duty cycling mechanism. | 48 |
| 4.1 | Example. | 59 |
| 4.2 | Simulation parameters. | 63 |
| 4.3 | The node densities and equivalent NTR. | 65 |
| 5.1 | 2's neighboring table and its EquivalentNeighborList. | 73 |
| 5.2 | Simulation parameters. | 78 |
| 5.3 | Number of groups formed in GECP and CPA under different node densities. | 79 |
| 6.1 | 6's and 9's neighboring tables and their equivalent neighbor lists for Topo1. | 95 |
| 6.2 | List of groups formed after the first round for Topo1. | 98 |
| 6.3 | 5's neighboring table and its equivalent neighbor list for Topo2. | 99 |
| 6.4 | List of groups formed after the first round for Topo2. | 99 |
| 6.5 | Simulation parameters. | 105 |
| 6.6 | Number of groups formed in RTCP, GAF and CPA under different node densities. | 106 |
| 7.1 | 6's, 7's and 9's, neighboring tables and their equivalent neighbor lists. | 121 |
| 7.2 | Simulation parameters. | 123 |
| 7.3 | Number of groups formed in ERTCP, RTCP, CPA, GAF and GECP under different node densities. | 124 |

Chapter 1

Introduction

1.1 Problem Statement and Motivations

One of the major advances in the field of surveillance technology is the deployment of distributed sensor networks. This has provided a mean to monitor areas, which are either unreachable or hostile to human existence. Wireless sensor networks (WSNs) have gained worldwide attention in recent years due to their potential applications for several domains such as military target tracking, traffic surveillance, health care, environment monitoring, etc. A WSN is a self-organized network that consists of a large number of sensor nodes deployed on the ground, in the air, in vehicles, on bodies, under water and inside buildings. Each sensor node is a tiny device that includes three basic units: a sensing unit for data acquisition from the physical environment, a processing unit for local data processing and storage, and a wireless communication unit for data transmission to a central collection point called sink.

Energy saving is one critical issue for sensor networks since most of sensor nodes are equipped with limited energy supplies, and it is inconvenient to recharge the batteries especially when nodes are deployed in a hostile environment or in locations with limited harvesting opportunities. When a sensor node is depleted of energy, it will die and disconnect from the network which can significantly impact the performance of the application. Moreover, WSNs are usually densely deployed on the target field; that is, many sensors can detect the same event, deliver and receive the sensed data simultaneously, which will cause redundant communication overhead and thereby leads to large amount of energy consumption. To achieve a sustainable and scalable WSN design, great efforts have been made to conserve sensors' energy.

In literature, besides the promising solutions which aim to recharge sensor batteries (e.g., [30]), most of the energy conservation works are mainly focusing on efficient energy consumption. In this sense, several energy-saving mechanisms exist and we can broadly classified them into five categories: Duty cycling (e.g., [18, 38, 80, 113]), energy-efficient routing (e.g., [75]), data reduction (e.g., [61]), power control (e.g., [130]) and node mobility (e.g., [12]). All these techniques have the same common objective

which is minimizing energy consumption to enhance the network lifetime.

Duty cycling has been considered as one of the most important techniques for energy conservation in WSNs [11]. Indeed, radio communication consumes a large amount of energy among all the node activities. Typically, a radio can operate in four distinct operation modes [117]: Idle, Receive, Transmit, and Sleep. While it is expected that the radio consumes most of the energy in the Transmit and Receive modes, running in the Idle mode is also costly. In most cases, operating in Idle mode results in significantly high energy consumption because the radio electronics are turned on and continually decode radio signals, even noise, to detect the presence of incoming packets. Thus, it is desirable to completely shut down the radio (enter into sleep mode) rather than transiting into the Idle mode. Putting some sensor nodes into sleep mode and allowing switches between sleep and active modes is referred to as *Duty Cycling*.

Duty cycling, also called sleep-scheduling technique, defines coordinated sleep/wakeup schedule among nodes in the network. This technique reduces significantly the energy consumption of sensor nodes as, ideally, it keeps nodes active only when there is network activity [11, 34, 82]. In addition, the duty cycling can be achieved by exploiting node redundancy and selecting only a minimum subset of nodes to remain active with the purpose of maintaining connectivity. Finding the optimal subset of nodes, which guarantees connectivity, is referred to as *Topology Control* [11].

Topology control is the primary technique to reduce energy consumption and radio interference [86]. It should ensure the quality of network coverage and connectivity. It aims at exploiting the network density to conserve energy and extend the network lifetime. Besides energy saving, topology control technique causes the number of transmitted messages to decline, which lowers signal interference and the failed transmission attempts [118].

Topology control approach can be classified into two dominant directions [11, 34, 55, 86]: *power control* and *sleep-scheduling*. The former reduces the transmission power of sensors, in order to extend the network lifetime, while the second defines coordinated sleep/wakeup schedule among nodes in the network. It is demonstrated that topology control based on duty cycling can increase the network lifetime by a factor of 2-3 with respect to a network with all nodes in wake-up mode [106].

Several topology control algorithms, using the scheduling technique, have been proposed for Ad Hoc and sensor networks. Based on the functionality of the network which aim to guarantee, they are classified into two categories [55, 83]: *network connectivity* and *network coverage*. The key idea of these protocols is to manage the topology by identifying redundant nodes and schedule nodes for active or sleep modes in order to reduce energy consumption.

In this thesis, we focus on the first category of protocols which aim to minimize energy consumption while maintaining a connected network. They determine how many

and which nodes should be allowed to sleep, with the purpose to ensure connectivity. However, when running these algorithms, the number of active nodes is not always the minimum and sometimes the formed topology is not connected. In fact, it is difficult to ensure connectivity without increasing the number of active nodes; when the number of active nodes is minimized, the topology formed by these nodes may be disconnected. Furthermore, a large communication overhead will be produced during the search of the redundant nodes which leads to large energy consumption. Moreover, few existing works take into consideration the energy wasted during transitions in their scheduling strategy. In fact, one of the major source of energy wastage is frequent switching between modes, especially switching from sleep to active mode that leads to significant energy consumption [10, 53, 93, 102, 117]. Therefore, it is important to consider this issue when designing energy-efficient protocols.

1.2 Contributions

The main contributions of this work are:

- A new classification of sleep scheduling-based topology control protocols, is proposed in this thesis. We classify the topology control protocols based duty cycling into four categories: Flat protocols, Grid-based protocols, Cluster-based protocols and Group-based protocols.
- The Proposition of a new grid-based topology control algorithm, called GTC: a Geographical Topology Control protocol, that benefits from the advantages of the existing grid-based algorithms to identify redundant nodes. GTC uses a new scheduling strategy which reduces the number of transitions between active and sleep states in order to minimize the transition energy and ensure the network longevity. It also minimizes the frequency of active nodes election and allows some stability in the topology that avoids a loss of connectivity. However, the proposed protocol requires the availability of location information provided by GPS, which is sometimes impossible to acquire. For this reasons, we should define a new mechanism to group nodes dynamically without using any localization system.
- The design of a Group-based Energy-Conserving Protocol (GECP) which exploits the sensor redundancy in the same region by dividing the network into groups so that a connected backbone can be maintained by keeping only one active node in each group and turning off the redundant ones. GECP identifies redundant nodes and organizes them into groups with a low communication overhead. It uses the same scheduling strategy as GTC which aims to minimize the number of transitions between active and sleep states and, hence, extend the network lifetime. However, in GECP, the criterion used to group equivalent nodes together

is too strong which leads to high number of active nodes. So, it is necessary to reduce the number of groups (hence, the number of active nodes) using another constraint to identify node redundancy without increasing the communication overhead.

- A new distributed Redundancy aware Topology Control Protocol (RTCP) is proposed. RTCP uses only the neighbor set as the communication redundancy metric, to avoid any assumptions about communication regularity. RTCP identifies redundant nodes and organizes them into groups with a low communication overhead. The redundant nodes that satisfy some eligibility rules can form a group according to their redundancy degrees, which leads to have a reduced number of groups without losing the network connectivity. RTCP defines a Threshold of connectivity level which allows applications to parameter the desired connectivity degree in the reduced topology. However, RTCP uses the node redundancy degree as criterion to group redundant nodes together with respect to Threshold of connectivity level. Finding the exact value of the Threshold which gives the minimum number of groups, is difficult when nodes do not have the same redundancy degree.
- The last contribution consists on the improvement of RTCP to allow to each sensor node to determine dynamically the value of the Threshold of connectivity level. The proposed protocol, called ERTCP: Enhanced Redundancy aware Topology Control Protocol uses a new metric to determine the equivalent nodes.

1.3 Thesis Organization

The organization of this thesis is as follows:

In Chapter 2, we propose a taxonomy of energy management approaches which can be used in WSN to improve the network lifetime. We present the different approaches, energy provision-based and energy consumption-based, that have been proposed for WSNs to meet several needs.

In Chapter 3, we introduce the concept of *Topology control* and the motivations behind the need to use this technique. We investigate the topology control strategies, namely *power control* and *sleep scheduling*. We focus, in this chapter, on topology control algorithms based on sleep scheduling which aim to reduce the number of active nodes without diminishing network connectivity and coverage. Thus, the existing sleep scheduling algorithms are reviewed and a taxonomy is provided.

Chapter 4, presents the design and the performance of our grid-based protocol GTC: a Geographical Topology Control protocol.

Chapter 5 is devoted to the design of a new group-based protocol, called GECP. Unlike GTC, GECP identifies redundant nodes without using any localization system.

In Chapter 6, another group-based topology control protocol, called RTCP, is proposed. RTCP tries to deal with the limitations of GECP and the other protocols, in terms of number of active nodes and communication cost.

Chapter 7 presents an extension of RTCP which aims to reduce the number of active nodes. The proposed algorithm is called ERTCP: Enhanced Redundancy aware Topology Control Protocol.

To conclude, Chapter 8 summarizes the main contributions of this thesis and discusses potential future work directions.

Chapter 2

Energy conservation in wireless sensor networks

2.1 Introduction

Wireless sensor networks (WSNs) have gained worldwide attention in recent years due to their potential applications for several domains such as military target tracking, traffic surveillance, health care, environment monitoring, etc. However, there are issues that need to be addressed for efficient operation of sensor network systems in real applications. Energy saving is one critical issue for sensor networks since most of sensor nodes are equipped with limited energy supplies, and it is inconvenient to recharge the batteries because nodes may be deployed in a hostile environment.

In order to sustain sensors to run for a long period of time with limited energy capacity, it is crucial to save energy in sensor operations. So, it is desirable to design energy efficient protocols while maintaining the network functionalities such as coverage and connectivity. Several energy conservation techniques can be used to design such protocols. As a consequence, the goal of any energy efficient technique is to maximize network lifetime.

We define, in this chapter, some important concepts such as coverage, connectivity and network lifetime. We introduce the different states that a wireless node can take and the different sources of energy waste in WSNs. Then, a taxonomy about energy conservation techniques is given followed by a discussion and recapitulative table.

2.2 Definitions

2.2.1 Coverage

Coverage and connectivity are important issues and requirements in wireless sensor networks. Coverage reflects how well the sensing field (area/target) is monitored by sensors.

Definition 2.1. Coverage: *A point p is covered or monitored by a node v if their*

Euclidian distance is less than the sensing range of v , R_s . Specifically, $|pv| < R_s$. [104]

2.2.2 Connectivity

Connectivity ensures that sensor nodes can communicate with each other in order to route the collected data to the base stations (sinks). Connectivity depends primarily on the existence of paths. Indeed, a sensor networks is considered to be connected only if there is at least one path between each pair of nodes in the network. A path can exist between two nodes u and v , if node u can receive a message from node v and vice versa. Formally,

Definition 2.2. Connectivity: Two nodes u and v can directly communicate with each other if their Euclidian distance is less than a communication range R_c . Specifically, $|uv| < R_c$. [104]

2.2.3 Network lifetime

Network lifetime strongly depends on the lifetimes of the single nodes that constitute the network. The lifetime of a sensor node basically depends on two factors [27]: how much energy it consumes over time, and how much energy is available for its use. According to different parameters that affect the lifetimes of single nodes as well as the overall network lifetime, several definitions of network lifetime exist. We present some important definitions in what follows [27]:

- Network lifetime based on the number of alive nodes:
 1. The time until the first sensor is drained of its energy [63, 67, 103];
 2. The time there is at least a certain fraction β of surviving nodes in the network [18, 26, 41];
 3. The time until all nodes have been drained of their energy [98].
- Network lifetime based on sensor coverage:
 1. The time the area of interest is covered by at least k nodes [71] ;
 2. The time each target is covered by at least one node [17, 60];
 3. The time the whole area is covered by at least one node [14].
- Network lifetime based on connectivity:
 1. The percentage of nodes that have a path to the base station [16];
 2. The number of total transmitted messages [13].
- Network lifetime based on both sensor coverage and connectivity:
 1. The time until connectivity or coverage are lost [68, 90].

2.3 Energy consumption of wireless nodes

Since the sensor lifetime depends largely on the energy amount of its battery, consumption must be well controlled in order to maximize its lifetime after deployment. In this section, we present the different operation modes of a radio and the sources of energy consumption in wireless networks.

2.3.1 Wireless node states

A typical sensor node can be in one of four distinct modes [117]: *Transmit*, *Receive*, *Idle listening*, or *Sleep*. In sleep mode, a sensor's radio is turned off; and the node is not capable of detecting signals (the majority of components are completely powered down). This state allows node to achieve a very low power consumption. If its radio is on and it is neither transmitting nor receiving, a sensor is said to be idle listening; an idle listening node is able to switch into receive mode if it hears a transmission. In most cases, operating in idle mode results in significantly high energy consumption, because the radio electronics have to be turned on and continually decode radio signals, even noise, to detect the presence of incoming packets [117]. When a sensor is transmitting, it can transmit packets using wireless channel. This state consumes the highest amount of energy of the four states. In the receive mode, the power is only slightly lower than the transmitting mode.

2.3.2 Basic energy consumption

In WSNs, sensors dissipate energy while *sensing*, *processing*, *transmitting* or *receiving* data to fulfill the mission required by the application. Measurements have shown that among of these major activities, a sensor expends maximum energy in data communication. This involves transmission, reception, and being idle. In addition, the transition between modes is energy and time consumer. Table 2.1 reports some typical values of Idle, Sleep and Transition energy costs, taken from various cards for the IEEE802.11 interfaces [88].

Table 2.1: Typical values of energy costs relevant to various cards.

| Card | $P_{Tx}(mW)$ | $P_{Rx}(mW)$ | $P_{Idle}(mW)$ | $P_{Sleep}(mW)$ | $P_{Transition}(J)$ |
|-------------------|--------------|--------------|----------------|-----------------|---------------------|
| Cabletron | 1400 | 1000 | 830 | 130 | 1.328 |
| Lucent Wavelan | 1327.2 | 966.9 | 843.7 | 66.3 | 0.6 |
| Cisco Aironet 350 | 1850 | 1590 | 1150 | 140 | 0.19 ~ 0.6 |

2.3.3 Sources of energy waste

In addition to the energy dissipated for wireless communication, there is a great amount of energy wasted in states that are useless from the application point of view, such as [69, 117]:

- **Overhearing:** meaning the node receive packets that are destined to other nodes. Indeed, in wireless networks, when a node transmits a packet, all nodes in its communication range receive this packet even if they are not the intended destination.
- **Collision:** collisions occur when multiple packets get transmitted simultaneously which magnifying the signal interference. All packets that cause the collision have to be discarded and the retransmission of these packets is required.
- **Control packet overhead:** the third source is the overhead of sending and receiving medium access control packets.
- **Idle listening:** energy dissipation happens when a node is listening for possible traffic.
- **Interference:** each node located between transmission range and interference range receives a packet but cannot decode it.
- **Wireless noise:** in which packets get corrupted and need to be retransmitted or to increase the transmission power to overcome the noise level.
- **State transitions:** frequent switching between modes, especially switching from sleep mode to an active mode, leads to significant energy consumption as shown in Table 2.1.

2.4 Energy saving techniques in WSNs

As sensor nodes are tiny battery-operated devices with limited energy, it is important to reduce the energy consumption of nodes, in order to maximize the network lifetime. Therefore, energy management (conservation) is a key issue in the design of systems based on WSNs. Energy management in WSNs can be defined as the set of rules to manage various energy supply mechanisms and efficient consumption of sensor nodes [48]. It is usually based on two design considerations [11, 48, 82]: *Energy provision* and *Energy consumption*. In literature, besides the promising solutions which aim to recharge sensor batteries, most of the energy conservation works are mainly focusing on efficient energy consumption.

In this section, we describe the major existing energy saving approaches, proposed in the literature. We propose a taxonomy of energy conservation techniques in WSNs, given in Figure 2.1.

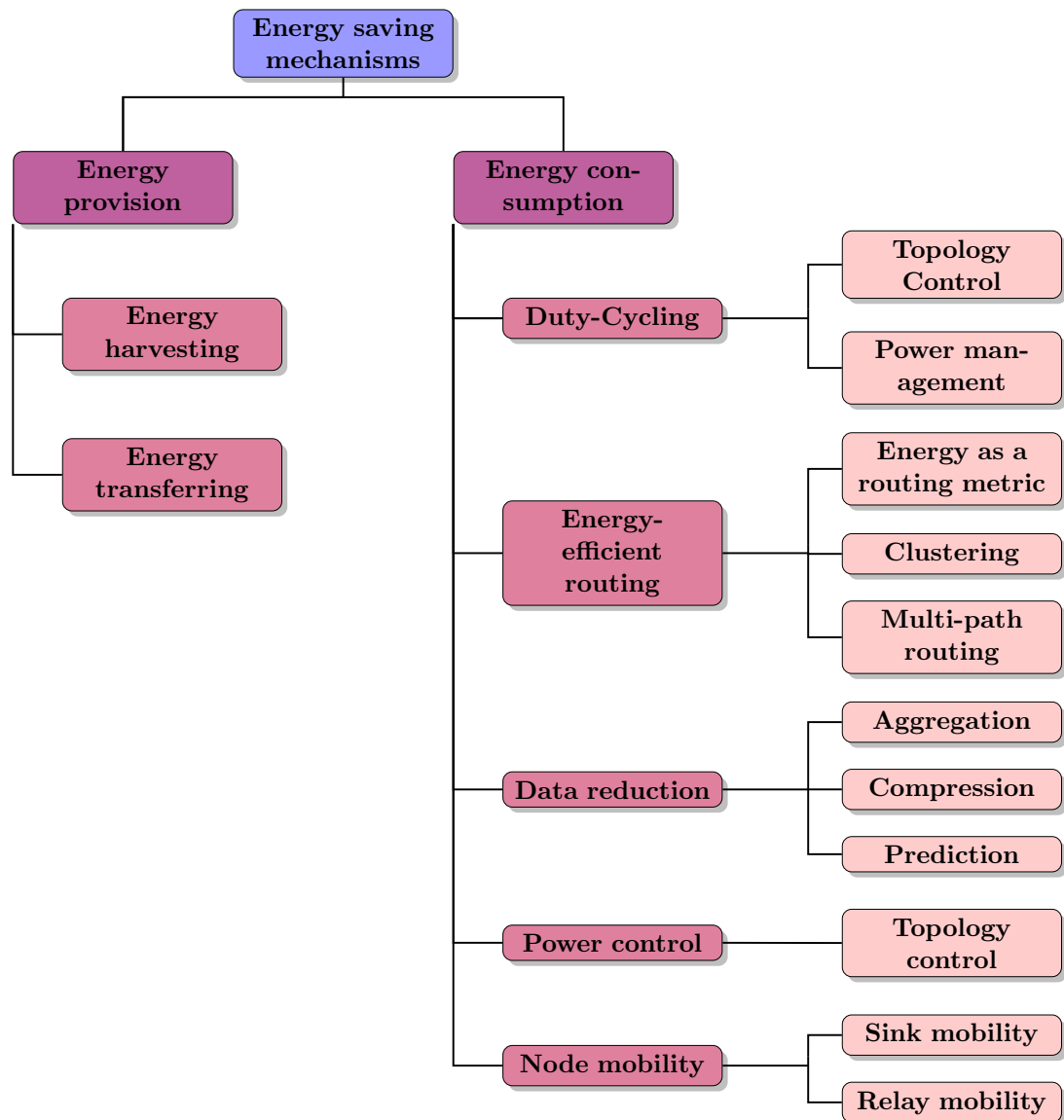


Figure 2.1: Taxonomy of energy conservation approaches in WSNs.

The limited energy of wireless nodes pushes many researchers to propose alternate energy provisioning mechanisms by exploiting ambient energy. The energy provision based schemes can be classified into *Energy harvesting* from the environment and *Energy transference* to charge nodes equipped with rechargeable batteries.

2.4.1 Energy harvesting

Ambient energy-harvesting technology is a promising approach to keep WSNs operating for a long period of time. Energy-harvesting is a technique that harvests or scavenges various ambient energy sources and converts the harvested energy into electrical energy to recharge the batteries [77, 87]. This technique enables sensors to extract energy from surrounding environment, such as solar power, wind and kinetic [30, 95], mechanical movement [70], heat [109] and fluid flow [7]. Energy harvesting based approaches provide several benefits such as reducing the dependency on battery power and provide long-term solutions. However, there exist some limitations as in some scenarios, a node may suffer from limited harvesting opportunity that does not meet its energy needs.

2.4.2 Energy transferring

Recently, energy transference based approaches are proposed, in which energy can be transferred on the field to an energy deficient node [50, 111, 112]. This technique can be used to transmit power between devices without the need of any contact between the transmitter and the receiver. The wireless energy is transferred through [108]: (1) electromagnetic radiation, (2) electric and magnetic resonant coupling to sensor nodes equipped with rechargeable batteries. In electromagnetic radiation mode, electric energy is generally converted into electromagnetic energy like microwaves or laser beams which can be radiated outward, then be received and converted back into electric energy with using a silicon rectifier antenna in the receiver. The principle of electric and magnetic resonant coupling mode is essentially the redistribution of the surface charges on the object. A high-frequency and high-voltage driver source excites the resonant transmitter to generate an alternating electric field which can couple with the resonant receiver. Energy will be delivered as soon as this coupling relation is set up.

Besides energy supply mechanisms which are considered as promising technologies, the nodes remain energy-limited when they are deployed in locations with limited harvesting opportunities. So, they still need to implement energy-saving mechanisms. In what follows, we present the most important efforts that have been made to develop efficient energy management schemes based on energy consumption.

2.4.3 Duty cycling

Duty cycling, also called node activity scheduling, is considered as the most effective technique to improve the sensor network lifetime. This technique allows nodes to alternate between wake-up and sleep modes to save their energy. It defines a coordinated sleep/wakeup schedule among nodes in the network. Indeed, only the sleep mode ensures energy saving since transmitting, receiving and idle listening lead to significant energy consumption. Duty cycling technique reduces significantly the energy consumption of sensor nodes as, ideally, it keeps nodes active only when there is network activity. So, the radio should be switched off as soon as there is no data to send/receive, and it should be resumed when there is a new data. Duty cycling can be achieved through two different and complementary approaches, namely: *Topology control* and *Power management*.

Topology control

Topology control is the primary technique of energy saving, and should ensure the quality of network coverage and connectivity. It exploits node redundancy, which is typical in dense sensor networks, and then adaptively selects a minimum subset of nodes to remain active, with the purpose of maintaining connectivity. Finding the optimal subset of nodes, that guarantees connectivity, is referred as *Topology Control* [11]. It is demonstrated that topology control based on duty cycling can increase the network lifetime by a factor of 2-3 with respect to a network with all nodes in wake-up mode [37, 106]. Topology control protocols refers to schemes which aim to dynamically adapt the network topology, based on the application requirements. Thus, they allow network operations while minimizing the number of active nodes (and hence, prolonging the network lifetime). Several criteria can be used to decide which nodes to activate/deactivate, and when. In this regard, topology control protocols can be broadly classified into two categories [11]: *Location driven* (e.g., [113]) and *Connectivity driven* (e.g., [20]). The former defines which node to turn on and when based on the location of sensor nodes which is assumed to be known. In the second one, sensor nodes are dynamically activated/deactivated so that network connectivity, or sensing coverage, are ensured.

Power management

Topology control and power management are complementary techniques that implement duty cycling with different granularity. Power management can be defined as duty cycling operated on nodes which have been selected as active to ensuring connectivity/coverage. Indeed, active nodes (i.e., may be nodes selected by topology control) can switch off the radio (enter sleep mode) when there is no network activity. Thus, they alternates between sleep and wake-up modes in order to save their energies. Power

management schemes are divided into two categories [11]: *Independent sleep/wakeup* protocols running at the network or application layer and MAC protocols with low duty cycle. Depending of the approach used to decide when sensor nodes should be activated, we can find three classes in this category: *On-demand* (e.g., [38]), *Asynchronous* (e.g., [76]) and *Scheduled rendezvous* (e.g., [47]). On the other hand, MAC protocols with low duty cycle can be classified into three main classes: *TDMA-based* (e.g., [80]), *Contention-based* (e.g., [115]) and *Hybrid* (e.g., [84]).

2.4.4 Energy-efficient routing

The main design goal of WSNs is to transmit the collected data to the sink nodes by employing energy efficient routing mechanisms. Routing in WSNs is very challenging since nodes are greatly constrained in terms of power supply, processing capability and transmission bandwidth. Routing protocols are responsible to discover routes between nodes and transmitting data packets, through the network, using these routes with the respect of energy efficiency. So, routing protocols are designed in such a way that they keep the network operational for a longer period by efficiently utilizing the available resources in the most economical way. Energy efficient routing mechanisms can be classified into three important categories: *Energy as routing metric* (e.g., [75]), *Clustering* (e.g., [40]) and *multi path routing* (e.g., [62]).

Energy as routing metric

In WSNs, nodes need to send the sensed data to the sink node for processing. To this end, nodes use paths chosen based on some cost functions. Routing based on cost functions has been studied extensively because of its distributed nature and good energy performance [59, 75, 78]. In such routing algorithms, a node having a packet to transmit decides locally which of its neighbors is the next hop based on some cost function. Indeed, they use cost functions to choose energy-efficient routes. A well-designed cost function will lead to energy-efficient decisions and prolonged network lifetime. There are many cost functions proposed in literature. The key idea is to maximize the network lifetime by defining link cost as a function of node remaining energy and the required transmission energy. The existed energy-efficient routing protocols often use residual energy, transmission power, or link distance as the metrics to select an optimal route. The energy metric that is used to evaluate routes is a very important component of the protocol. Depending on the metric, the characteristics of the protocol can change substantially.

Clustering

The hierarchical-based routing is considered to be more scalable and energy-aware in the context of WSNs. In hierarchical-based routing, nodes play different roles in the

network and typically are organized into clusters. Clustering is the method by which sensor nodes in a network organize themselves into groups according to specific requirements or metrics. Grouping sensor nodes into clusters has been widely pursued by the research community in order to achieve some objectives, such as [5] : energy saving, scalability, load balancing, fault tolerance. Every cluster has a leader, often referred to as the cluster-head (CH) and other ordinary member nodes. From a routing perspective, clustering allows to split data transmission into intra-cluster and inter-cluster communication. This separation leads to significant energy saving since the radio unit is the major energy consumer in a sensor node. In fact, member nodes are only allowed to communicate with their respective cluster heads, which are responsible for relaying the data to the sink with possible aggregation and fusion operations. In addition, clustering techniques can enhance energy efficiency by: (1) reducing the communication range inside the clusters which requires less transmission power and (2) enabling to power-off some nodes within the clusters while cluster heads ensure data forwarding. The most famous clustering-based routing protocols are LEACH [40] and TEEN [65].

Multi path routing

In the literature, several routing protocols have been proposed for WSNs [6, 8], many of which adopt single path routing techniques. Single path routing is simple and scalable, but does not efficiently satisfy the requirements of resource constrained WSNs [89]. While considering the characteristics of WSNs, single path routing is not efficient. Indeed, in the single path routing, the usage of the same optimal path several times may cause a faster energy depletion of nodes on the selected path, which may cause network partition and, hence, shorten the lifetime of WSNs. With the help of multi path routing, the available resources at each node can be used more efficiently. Multi path routing can overcome significant drawbacks of a single path routing scheme because it can provide fault-tolerance, reliability, security and load balancing [19, 89, 97], which are especially critical in the resource constrained system such as WSNs.

Multi path routing is an alternative technique which enables a source node to discover several paths towards the destination. The multi path routing approach can be used to recover from failures. Generally, one of the constructed paths is considered as primary and used for data transmission and the other ones are used when this path fails. In addition, multi path routing increases data reliability by sending the data along multiple redundant paths [72]. Even if some paths fail, the data will have a very high probability to be received by the destination node. The security can also be improved using the multi path routing. When data is sent from multiple paths, even in the presence of malicious paths, the original data can reach the final destination by using reliable paths. Extending network lifetime can also be achieved through using multiple paths for data transmission. In fact, multi path routing allows homogeneous

distribution of energy among the nodes by alternating forwarding nodes [62]. Load balancing using multi path routing helps to improve the network lifetime by delaying the appearance of network partition, although more data could be transmitted than that using single path routing [89].

2.4.5 Data reduction

Reduce the amount of data to be delivered to the sink node can have a major impact on energy consumption due to communication. Data reduction is another category of energy saving solutions which addresses the case of unneeded samples [11]. Sampled data generally have strong spatial and/or temporal correlations [101]. Therefore, there is no need to communicate the redundant data to the sink in order to avoid an useless energy consumption. Among the methods of data reduction, we find: *Aggregation*, *Compression* and *Prediction*.

Aggregation

Data aggregation, called also In-network processing, is defined as the process of aggregating the data from multiple sensors to eliminate redundant transmission and provide fused information to the base station [79]. In data aggregation schemes, intermediate nodes perform data fusion to reduce the amount of data forwarded to the sink node. For example, nodes re-transmit a new data resulting of computing the average or the minimum of the received data. The most appropriate data aggregation technique depends on the specific application and must be tailored to it. It is frequently used in data gathering applications, where it enables high benefits. In a data gathering application, a node sends to its parent a single message containing the values transmitted by its children, hence reducing wasteful transmissions. As data aggregation is application-specific, more details about in-network processing techniques can be found in [33].

Compression

Data compression can be applied to reduce the amount of information sent by source nodes. This method involves encoding information at nodes which generate data, and decoding it at the sink node. The information may be encoded in such a way that the number of bits needed to represent the initial message is reduced. Due the the number of bits to be transmitted, energy can be saved provided that the encoding and decoding computation cost is low [96]. Data compression algorithms can be classified into two categories [94]: a distributed data compression approach and a local data compression approach. Distributed data compression approaches in WSNs are usually applied in dense sensor networks, which can be broadly classified into four main techniques [66, 94, 105]: *distributed source modeling (DSM)*, *distributed transform coding*

(DTC), distributed source coding (DSC) and compressed sensing (CS) techniques. On the other hand, the local data compression approach performs data compression locally on each sensor node without distributed collaboration among other sensor nodes. As a result, these schemes usually exploit only temporal correlation of the data and do not depend on the specific WSN topologies. These schemes are suitable for sparse WSNs which have a low spatial correlation property. This data compression category can be broadly classified into two main techniques [42, 94]: *lossless compression algorithms*, which ensure the integrity of data during the compression/decompression process; and the *lossy compression algorithms*, which may generate some loss of information, but generally ensure a higher compression ratio.

Prediction

Data prediction is an important approach for reducing redundant data communications and saving sensor nodes' energy. It takes advantage of data coherence to remove redundancy and reduce transmissions among sensor nodes [79]. This can effectively improve the energy efficiency and extend the network lifetime. Data prediction consists in building an abstraction of a sensed phenomenon, i.e., a model describing data evolution [11]. The model used by the sensor nodes and the sink, can predict the values sensed by sensor nodes within certain error bounds. For example, in [107], three data prediction approaches are used to predict the data of the next period at both sensor and sink ends based on the same small number of recent data items. When data of the next period is sensed, the sensor node compares the predicted data with the sensed data. The sensor node does not forward the sensed data to the sink node when the prediction error is less than a pre-configured threshold value. In this case, the sink node considers the predicted data as the sensed data in current sensing period. Therefore, unnecessary transmission is eliminated and energy is saved.

In [11], data prediction approaches are overviewed and classified into three types: stochastic approaches, time series forecasting, and algorithmic approaches. Stochastic approaches (e.g., [23]) exploit the probabilistic and statistical properties of sensed data. The time series forecasting approaches may use different models to predict the readings of individual sensors at the sink node, and to detect when sensor nodes produce outlier readings (e.g., [52]). In algorithmic approaches (e.g., [39]), data are aggregated by exploiting the heuristic or behavioral characteristics of the sensing phenomena.

2.4.6 Power control

The transmission range reflects the maximum distance at which a sender can reach a nearby receiver. The longer the range is, the higher the power consumption would be [118]. Many of the advanced radios allow programmable transmission power so that a node can avoid consuming excessive energy to reach its receivers. Adjusting

dynamically the transmission power is another technique for reducing, significantly, energy consumption in WSNs. Instead of transmitting at the maximal power level, each node can choose a lower transmission power. Authors in [24, 64] summarize the goals to conduct the power control in WSNs as follows:

- Reducing the energy consumption since the power grows at least quadratically with distance;
- Reducing interference;
- Improving spacial reuse and mitigating the MAC layer medium contentions.

The essence of power control is to form proper connectivities among sensors in the network to reduce energy consumption and improve the network capacity [55]. So, power control is another way to manage a network's topology. Indeed, in a densely deployed WSN, a sensor node may have many neighbor nodes with which it can communicate when using a large transmission power. However, high transmission power requires more energy consumption which is not beneficial. To overcome these problem *topology control* can be applied.

Topology control

The fundamental idea behind topology control is to build and maintain a reduced topology that will save the small energy budget of sensors while preserving network connectivity and coverage [51]. This can be achieved by reducing the transmission power of sensors. In topology control based on power control, each node adjusts its power transmission instead of transmitting using the maximal power. Thus, energy dissipated in transmission is reduced and a new network topology is created. Several topology control based on power control solutions are proposed in the literature [56, 58, 130]. They are, generally, based on one of the three following algorithms minimizing the number of edges in the network graph while ensuring connectivity: *RNG*, *Relative Neighborhood Graph* [100], *MST*, *Minimum Spanning Tree* [35] and *LMST*, *Local Minimum Spanning Tree* [56].

As mentioned in section 2.4.3, duty cycling can be achieved through topology control. So, topology control can be classified into two categories: sleep-scheduling and power control. The first category relies on duty cycling to reduce the number of active nodes. The second one is based on power control to reduce the number of links in the network.

2.4.7 Node mobility

Recently, mobility has been considered as an alternative solution for energy-efficient data collection in WSNs. In WSNs, sensor nodes transmit data, following the multi-hop mechanism, to the sink node which is, generally, static. Nodes closer to the sink

should relay more packets than the other sensor nodes, leading to faster depletion of their batteries. Therefore, the network is subject to disconnect due to the nodes' premature energy depletion, even when energy conservation techniques are applied [54]. To deal with, mobile devices can be designated as responsible for data collection. So, mobility-based energy conservation can be achieved by considering few mobile nodes which move around the network to collect node information. Mobility-based energy conservation schemes can be classified, depending on the nature of the mobile element, into [11]: *Sink mobility* or *Relay mobility*.

Sink mobility

In the mobile sink based approach, a mobile sink is used to collect data from source nodes in the field in order to increase network lifetime. It is shown that mobile sinks can improve network lifetime by 5-10 times than static sinks [48]. Some examples of such approaches are Greedy Maximum Residual Energy protocol (GMRE) [12] and Scalable Energy-efficient Asynchronous Dissemination protocol (SEAD) [49].

Relay mobility

In a mobile relay-based approach, a well-known mechanism can be used which is message ferries schemes for data collection from source nodes [131]. These message ferries move in the field to collect data, carry the stored data and forward it to the destination node. Mobile relays have almost similar functionality as in data Mobile Ubiquitous LAN Extension (MULE) approaches [43, 91], where the vehicles periodically visit a network to collect data.

2.4.8 Discussion

To achieve a sustainable and scalable WSN design, great efforts have been made to conserve sensors' energy. It is clear that all energy saving techniques have the same common objective which is minimizing energy consumption to enhance the network lifetime. However, they can have different other requirements such as connectivity, coverage, security, scalability and fault tolerance. Besides these requirements, the energy conservation techniques try to meet some Quality of Services (QoS)¹. For example, the throughput is decreased when the number of nodes increases. In fact, nodes share the same transmission medium. When a node transmits, its neighboring nodes are prohibited from transmitting due to the interference. Therefore, techniques which reduce the interference ensure high throughput. They include energy efficient routing, mobility-based approaches and those that reduce the number of active nodes, i.e., duty

¹QoS can be regarded as the capability to provide assurance that the service requirements of applications can be satisfied [110]. Depending on the type of application, QoS in WSNs can be characterized, among others, by reliability, robustness, throughput, delay [21, 57].

cycling and the number of neighbors, i.e., power control. In addition, studies show that using multi path routing in dense networks results in better throughput than using single path routing [72]. Since there is more bandwidth available (high throughput), a smaller end-to-end delay may be achieved. However, there is a trade-off between reducing the delay and improving the throughput. For example, topology control-based power control aims to reduce the number of links by using a small transmission range while ensuring connectivity. It is clear that a small transmission range limits the interference and consequently leads to high throughput. However, a smaller transmission range means that a packet needs to be transmitted through more hops, which inevitably leads to higher transmission delay. Gamal et al. [36] show that the delay due to the multi-hop transmission is increased when the throughput scales.

On the other hand, data redundancy in WSNs can be exploited to improve information reliability. Indeed, nodes spend too much energy to transmit the redundant data. If proper techniques are used, this redundancy can help the network to be more reliable and fault tolerant as failures in individual nodes do not affect the overall network behavior [22]. This can be achieved by introducing data reduction techniques which can reduce data redundancy in order to save energy. However, data reduction techniques, especially, data aggregation can introduce much delay into the network. In fact, when the data aggregation is used, each node in the network must wait his predecessors to aggregate their data. Consequently, data aggregation can be a source of problems for applications with time constraints. To deal with, this method is often employed with clustering approach which can improve the network delay while satisfying scalability and energy efficiency. Furthermore, redundancy affects the energy consumption of the WSN as the same readings are gathered by physically close nodes. Duty cycling techniques can help to ensure reliability by reducing the number of active nodes without sacrificing the other QoS metrics such as throughput and delay.

Table 2.2 summarizes the objectives of each technique in terms of network requirements and QoS metrics.

2.5 Conclusion

As sensor nodes operate on limited battery power, energy usage is a very important concern in a WSN. When a sensor node is depleted of energy, it will die and disconnect from the network which can significantly impact the performance of the application. In this sense, a significant attention has been given to mechanisms that can harvest and save sensor nodes' energy, with the purpose to prolong the network lifetime. In this chapter, we gave a taxonomy of energy management approaches that can be used in WSN to improve the network lifetime. We presented the different approaches, energy provision-based and energy consumption-based, that have been proposed for WSNs to meet several needs. Energy provisioning mechanisms are promising technology which

Table 2.2: Objectives of energy conservation techniques: Network requirements and QoS metrics.

| Energy-efficient approaches | | Objectives | |
|-----------------------------|----------------------------|---|------------------------------------|
| | | Network properties | QoS metrics |
| Duty cycling | Topology control | Network lifetime Coverage Connectivity | Delay Throughput |
| | Power management | Network lifetime Connectivity | Reliability |
| Energy-efficient routing | Energy as a routing metric | Network lifetime Connectivity | Delay Reliability |
| | Clustering | Scalability | |
| | Multi path routing | Network lifetime Connectivity Security Fault tolerance | Delay Throughput Reliability |
| Data reduction | Aggregation | | |
| | Compression | Network lifetime | Reliability |
| | Prediction | | |
| Power control | Topology control | Network lifetime Connectivity | Throughput |
| Node mobility | Sink mobility | Network lifetime Security | Delay Throughput |
| | Relay mobility | Network lifetime | |

can exploit ambient energy to recharge the sensors' batteries. However, nodes remain energy-limited when they are deployed in locations with limited harvesting opportunities. So, they still need to implement energy-saving mechanisms which we classified into five categories: Duty cycling, energy efficient routing, data reduction, power control and node mobility. All these techniques have the same common objective which is minimizing energy consumption to enhance the network lifetime. Besides energy saving, other requirements and QoS metrics can be ensured when using energy conservation approaches. In fact, there are trade-offs between multiple requirements, including energy saving, which should be considered when designing protocols for WSN.

Finally, we can say that duty cycling approach, especially topology control, is the most important energy saving mechanism which meets several objectives. In fact, sensor network lifetime depends on the number of active nodes and connectivity of the network. By eliminating the redundancy, which is typical in WSN, the network lifetime will be maximized while meeting other requirements and QoS parameters as discussed in section 2.4.8.

In the next chapter, topology control approach will be discussed in more detail, and a taxonomy of topology control protocols will be proposed.

Chapter 3

Topology control in wireless sensor networks

3.1 Introduction

As mentioned in the previous chapter, several energy conservation schemes can be used to extend the sensor network lifetime. Among all the energy saving techniques, topology control is considered as the most important technique in terms of energy efficiency and network lifetime. This technique is very useful in WSNs because they are deployed densely which offers much redundancy in network coverage and connectivity. In many cases, the network deployment is done at random and it may be convenient to deploy more nodes than necessary to cope with possible node failures. The fundamental idea behind topology control is to build and maintain a reduced topology by exploiting the network density to conserve energy and extend the network lifetime.

In this chapter, we introduce the concept of *Topology control* and the motivations behind the need to use this technique. We investigate the topology control strategies which are *power control* and *sleep scheduling*. We focus, in this chapter, on topology control algorithms based on sleep scheduling which aim to reduce the number of active nodes without affecting network connectivity and coverage. Thus, the existing sleep scheduling algorithms are reviewed and a taxonomy is provided.

3.2 Definitions

3.2.1 Topology of the network

The topology of a network is the basis for its performance and all important properties such as connectivity, network capacity and routing efficiency, rely on it [4]. The topology of a WSN refers to the network layout; that is, the "set of communication links between node pairs used explicitly or implicitly by a routing mechanism" [81]. In other words, the topology of the network is determined by nodes and links that allow direct communication (one hop). Once a WSN is deployed, each node communicates with a

subset of nodes according to the distance between them. Therefore, communication links are established with those nodes that are close enough for the radio signal. These nodes and links establish the *topology of the network* [51].

3.2.2 Topology Control

Due to the high density and random deployment of WSNs, many sensor nodes may be localized very close to each other, which provides a large amount of redundancy in network coverage and connectivity, known as *spacial redundancy*. Spatial redundancy means the possibility to obtain information for a specific location from different sources [25]. In this case, the sensing information will tend to be very similar, thereby leads to large amount of energy consumption. Further, having redundant nodes active in the network is not even desired; they waste additional energy retransmitting packets that collide because of the increased network traffic. To deal with, it is necessary to manage the topology in order to reduce energy consumption as well as maintaining a connected network. This can be achieved by modifying the topology of the network taking advantage of the sensors transmission power adjustment and their different modes of operation. So, *Topology Control* can be defined as the mechanism which exploits the spacial redundancy and, adaptively, selects only a minimum subset of links and nodes while maintaining connectivity and/or coverage. Specifically,

Definition 3.1. *Topology Control is the reorganization and management of node parameters and modes of operation from time to time to modify the topology of the network with the goal of extending its lifetime while preserving important characteristics, such as network connectivity and sensing coverage [51].*

3.3 Motivations for Topology Control

Topology control is one of the most fundamental problems in wireless sensor networks. Controlling the topology of the network has emerged as an effective way to reduce the power consumption. Being an effective energy-saving technique, topology control has received a significant amount of attention. Topology control is concerned with how to create a good topology to be used for routing in WSNs. The question we should answer is: Why we need to control the topology? Simply because without using topology control mechanism, a randomly connected WSN may suffer from [4]: (1) short network lifetime, (2) high radio interference, (3) poor network capacity, (4) high end-to-end packet delays, and less robustness to frequent node failures. Therefore, topology control plays an important role in saving energy (and hence prolonging network lifetime), improving spatial reuse (and hence increase network capacity), reducing radio interference (and hence mitigating MAC contention), among others [129]. In what follows, we summarize the main design objectives behind the use of topology control in WSNs.

3.3.1 Maximizing network lifetime

The concept of topology control is strictly associated with network redundancy [11]. A dense and randomly deployed WSN may contain many redundant nodes that will consume additional energy unnecessarily. Topology control aims to build a reduced topology that will save energy and ensure some important network characteristics, such as connectivity and coverage. This can be achieved by exploiting node and/or link redundancy. So, the redundancy can be eliminated by either reducing the nodes' transmission power or by turning unnecessary nodes off. Both mechanisms could change the topology of the network and could extend, considerably, the network lifetime.

Reducing the transmission power of the sensor nodes allows eliminating some direct links and forces packets to go through multi hops which lowers down the communication cost. Since multi-hop relay usually costs much less energy than one long hop transmission, the network topology derived is more energy-efficient from communication cost reduction perspective, which increases the network lifetime.

On the other hand, turning the redundant nodes off have the same effect that reducing the transmission power in term of eliminating some direct links. However, more energy can be saved since sleep nodes don't have to receive packets during the routing process. The network topology derived, in this case, is more energy-efficient in the sense of reducing energy consumption in idle states which extends the network lifetime.

3.3.2 Increasing network capacity

The goal of topology control schemes is to eliminate redundancy, which can have a direct effect on minimizing energy consumption and a collateral effect on reducing the number of transmissions and the number of packet collisions. Indeed, in dense networks, data transmissions suffer from high collisions due to (1) the property of wireless communication and (2) the high node degree¹. A collision may occur if two nodes within each other's communication range try to transmit at the same time on the same frequency. Transmitting nodes influence the ability of other nodes to receive data. A node is not able to receive data from its neighbor if another neighbor is transmitting at the same time. This mutual disturbance of communication is called *interference* [44]. Reducing interference in the network leads to fewer collisions and packet retransmissions, which indirectly reduces the power consumption, extends the network lifetime and increases the network capacity. In fact, besides reducing energy consumption, topology control has the positive effect of reducing contention when accessing the wireless channel which can avoid interference and increase the network capacity [51, 86].

Turned off some of the redundant nodes in a WSN can conserve energy since fewer nodes transmit data to the base station. Moreover, this technique causes the num-

¹The node degree can be defined as the number of neighbor nodes.

ber of transmitted messages to decline, which lowers signal interference and the failed transmission attempts [118]. On the other hand, reducing the transmission power can eliminate (1) redundant links leading to a low node degree and (2) the exposed terminal problem. Thereby, if two nodes do not have overlapping radio ranges then they may be able to transmit at the same time without causing interference to each other which avoid collision [34, 51]. So, topology control mechanisms can reduce interference and boost the network throughput.

Figure 3.1 shows an example of a network topology with and without using topology control scheme.

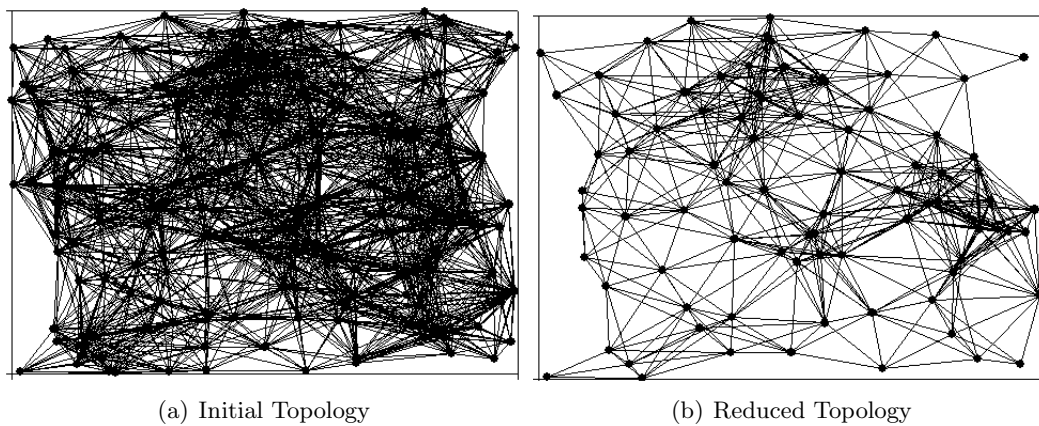


Figure 3.1: Example of a network topology (a) without topology control (b) with topology control.

3.3.3 Ensure coverage and connectivity

The primary objective of the topology control techniques in WSNs is to achieve sustainable coverage while maintaining network connectivity and conserving energy [118]. Network connectivity enables nodes to coordinate their action while performing a task, and to forward their readings to a base station or a sink [45, 85]. Therefore, the connectivity have a significant impact on the effectiveness of WSNs and should be sustained all the time.

For example, topology control approach is employed to conserve energy by switching off some of redundant nodes without degrading network coverage and connectivity. The basic idea is to only power on a small number of nodes that are sufficient to maintain network connectivity. However, the use of low transmission power limits the network connectivity since nodes would have fewer directly reachable neighbors. So, there is a tradeoff between increasing network capacity (i.e., avoiding interference) and connectivity: as links are removed to reduce interference, the paths in the network grow in length which leads to higher delay [44]. Unlike duty cycling topology control, power control is purely a link-layer technique which does not affect coverage or the

data-processing tasks that a node performs [118]. Network coverage describes how well the target field is monitored by the sensor nodes. The problem is how to achieve a reliable sensing area that satisfies certain application requirements while consuming less energy.

Therefore, minimizing the transmission power and scheduling the sensor nodes status allow the network operator to design a low energy consuming topology while preserving the desired sensing and communication connectivity and coverage [51]. Indeed, both topology control techniques maintain a reduced topology which must (1) be connected, so all active nodes can exchange information among themselves as well as with the sink node and (2) cover the area of interest.

3.4 Topology control mechanisms

Topology control in WSNs is, mainly, focused on how to prolong the network lifetime while satisfying some requirements, such as network connectivity and coverage. Topology control mechanisms are broadly classified into *power control* and *sleep-scheduling*², as shown in Figure 3.2. The former works via controlling the radio power level to achieve optimized connectivities among sensor nodes. The latter one works by maintaining a good active/sleep schedule of each sensor node to prolong the system lifetime while keeping almost equivalent system capability in the topological sense [55].

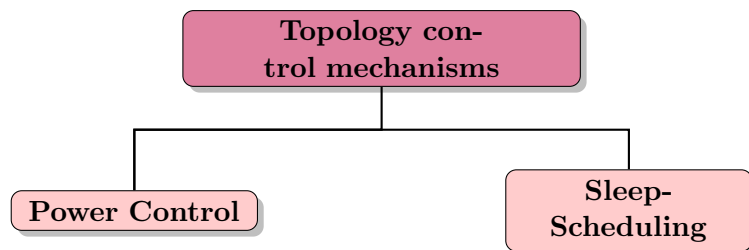


Figure 3.2: Topology control categories.

3.4.1 Power control

Since transceivers are one of the primary sources of energy consumption in the wireless unit, especially in WSNs, topology control mechanisms are fundamental to achieving a good network energy efficiency. The goal of topology control based on power control is to dynamically change the nodes' transmitting range in order to maintain connectivity, while reducing the energy consumed by node transceivers. So, power control refers to mechanism that adapt the transmission power level to optimize a single wireless transmission [11, 86]. The key idea of power control is that, instead of transmitting

²Also called sleep/wakeup scheduling, state scheduling or multi state.

packets using the maximum power, sensor nodes collaboratively adjust their transmission power while ensuring wireless connectivity. Table 3.1 provides some typical transmission power levels and their corresponding energy consumptions of CC2420 radio, according to [1]. The optimal transmission power (between 0 and the maximum value) can be decided by the power control based on the application requirement.

Table 3.1: Output power settings and typical current consumption of CC2420.

| Power Level | Output Power (dBm) | Current (mA) |
|-------------|--------------------|--------------|
| 3 | -25 | 8.5 |
| 7 | -15 | 9.9 |
| 11 | -10 | 11.2 |
| 15 | -7 | 12.5 |
| 19 | -5 | 13.9 |
| 23 | -3 | 15.2 |
| 27 | -1 | 16.5 |
| 31 | 0 | 17.4 |

3.4.2 Sleep-scheduling

Sleep scheduling is another approach for energy efficient consumption which allows some redundant nodes to sleep and conserve their energy. The basic idea is to activate a small subset of nodes and switch off the radio of the redundant one while fulfilling two requirements: connectivity and coverage. Unlike power control which aims to reduce energy consumption in wireless communications, sleep scheduling reduces that consumed in idle states. Indeed, radio communication consumes a large amount of energy among all the node activities. While it is expected that the radio consumes the most energy in the Transmit and Receive modes, running in the Idle mode is also costly. In most cases, operating in Idle mode results in significantly high energy consumption, because the radio electronics are turned on and continually decode radio signals, even noise, to detect the presence of an incoming packets. Many measurements (e.g, [1, 31, 88]) have shown that for many systems, radio idle listening dominates the system power budget. It is, thus, desirable to completely shut down the radio (enter into sleep mode) rather than transiting into the Idle mode. Authors in [31] reveal a considerable energy consumption gap between the active state and sleep state of some typical radios. Table 3.2 summarizes the energy consumptions of some mainstream radios according to [31].

Let's take an application example depicted in Figure 3.3(a) to better show the impact of the two topology control mechanisms on WSN topology. Suppose only 3 neighbor nodes in a WSN, where S is the source node, R is the relaying node and D

Table 3.2: Energy consumptions of some IEEE802.15.4-compatible radios.

| Radio | Rx(mA) | Tx(mA) | Sleep (μ A) |
|-------------------------|--------|--------|------------------|
| RF230 | 15.5 | 16.5 | 0.02 |
| EM260 | 28 | 28 | 1 |
| MC13192/MC13202/MC13212 | 37 | 30 | 1 |
| JN5121 | 38 | 28 | <5 |
| JN5139 | 37 | 37 | 2.8 |
| CC2420 | 18.8 | 17.4 | 1 |
| CC2430 | 17.2 | 17.4 | 0.5 |
| CC2520 | 18.5 | 25.8 | 0.03 |

is the destination node. If node S have to transmit packets to node D , there are two possible paths from node S to D . One is $S \rightarrow D$, the other is $S \rightarrow R \rightarrow D$. In the first case, a large amount of energy can be expended: (1) node S spends a significant energy (uses a high transmission level) to reach directly node D , and (2) node R expended considerable energy upon receipt of these packets that are not destined for it.

Power control can save energy by reducing the power level without losing connectivity. In other words, power control aims to eliminate link redundancy. The derived topology under power control is given in Figure 3.3(b). Contrary, sleep scheduling aims to lower down energy consumption in idle states. Since the network connectivity still holds without R , this later should go into the sleep mode in order to save more energy. Therefore, the network topology derived under sleep scheduling is shown in Figure 3.3(c).

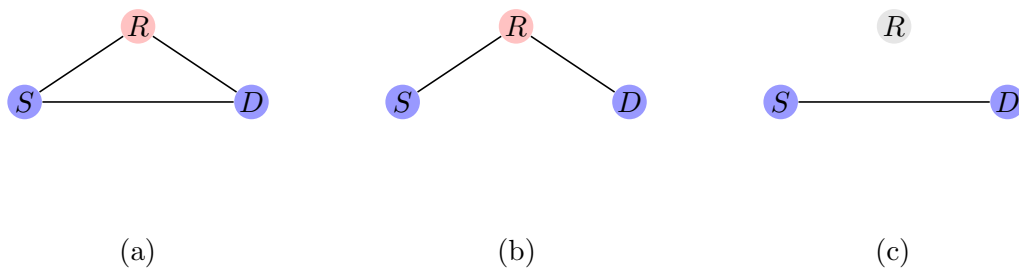


Figure 3.3: (a) The initial network topology represented by a communication graph. (b) The reduced topology under power control. (c) The reduced topology under sleep scheduling.

3.5 Need for sleep scheduling

As mentioned previously, since most energy is spent for radio communications, an effective approach for energy conservation is scheduling sleep intervals for redundant nodes,

while the other nodes stay active to provide continuous service. Thus, connectivity can be maintained with much fewer routing nodes and more energy saving. For example, an area that has ten nodes deployed may only require three nodes to get complete coverage of the area and two additional nodes to ensure connectivity. This means that five of the deployed nodes are considered redundant in terms of sensing and communication and may not always be necessary. When these redundant nodes are not required they will be put to sleep. Determining which nodes are redundant, putting them to sleep, and waking them up again are tasks of sleep scheduling algorithms.

Topology control based on sleep/wakeup scheduling contributes in power saving in two ways [106]: (1) it allows non-routing nodes (or sensing nodes) to maintain lower duty cycle because they don't have to receive packets for the routing purpose and (2) routing nodes can act as data aggregation points as all packets are forwarded through these nodes. Further of energy saving, this mechanism can increase the network lifetime and network capacity without losing network functionalities.

While topology control based on power control aims to save energy by reducing the transmission power, sleep scheduling minimizes, also the energy consumption in receiving and idle modes and sensing coverage. It can increase the network lifetime by a factor of 2-3 with respect to a network with all nodes in wake-up mode [37, 106]. Consequently, topology control based on duty cycling is the most important energy conservation technique for a high density wireless ad hoc and sensor networks. For all these reasons, our work is axed in this direction.

3.6 Sleep scheduling-based topology control protocols

Several topology control algorithms, using the scheduling technique, have been proposed for wireless ad hoc and sensor networks. Based on the functionality of the network which aim to guarantee, they are classified into two categories [55, 83]: *network connectivity* [2, 3, 15, 18, 20, 28, 113] and *network coverage* [32, 73, 99, 104, 128, 114]. The key idea of these protocols is to manage the topology by identifying redundant nodes and schedule nodes for active or sleep modes in order to reduce energy consumption. In this thesis, we focus on the second category which aims to minimize energy consumption as well as maintain a connected network. This category of algorithms determine how many and which nodes should be allowed to sleep, with the purpose to ensure connectivity.

In this section, we present and describe some representative existing sleep-scheduling algorithms, proposed in the literature. These protocols are classified and a taxonomy is proposed as given in Figure 3.4.

Topology control protocols based sleep-scheduling can be divided into two categories: flat and hierarchical protocols. In the first category, the election of active nodes is done in the entire network so as to allow the required number of nodes to remain active and ensure connectivity and/or reduce the rate of messages loss. In the second

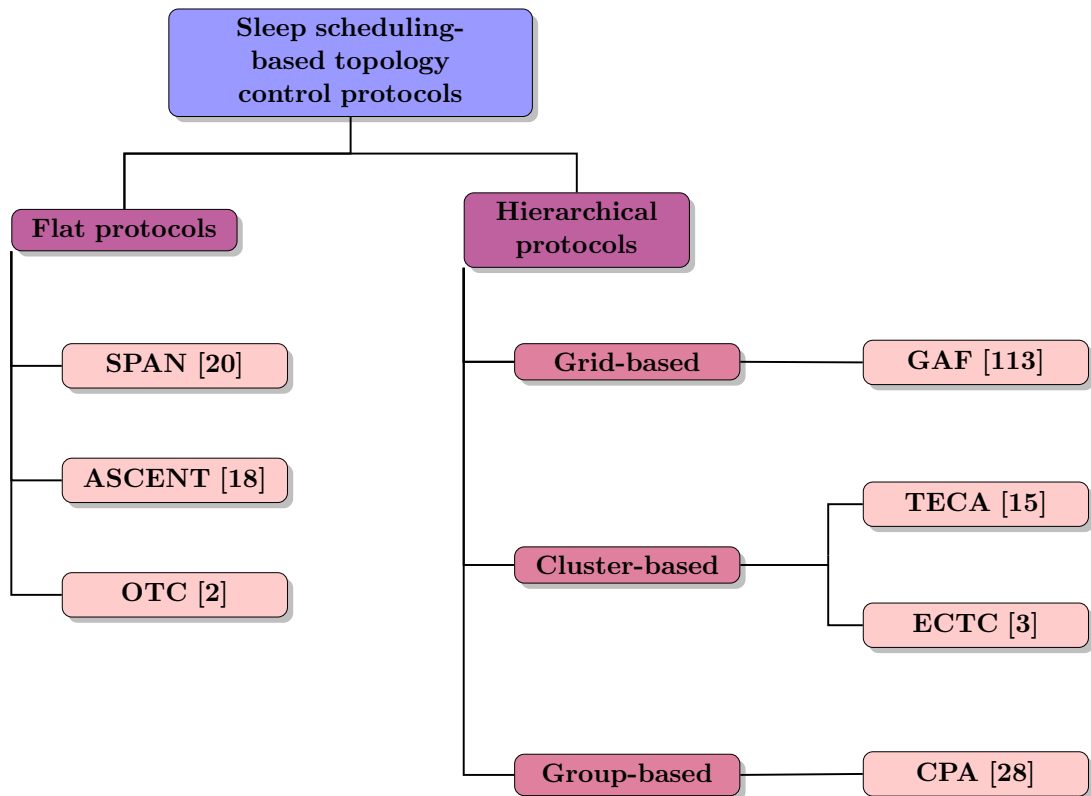


Figure 3.4: Classification of topology control protocols based on sleep scheduling.

category, nodes are first organized, either into clusters or groups. Then, active nodes are selected from each cluster or group. Cluster-based solutions require the activation of two or three types of nodes: cluster-heads, primary gateways, and secondary gateways. These types of nodes should remain active to ensure connectivity while other nodes can go into energy conservation mode, by turning off their radio (sleep mode). In the other hand, solutions that organize nodes into groups can identify active nodes, based on either location information (Grid-based), or connectivity information (Group-based). In the Grid-based solutions, the formation of groups is static and gives the same number of groups regardless of the number of nodes in the network, while, the Group-based solutions form groups dynamically. So, we classify the existing protocols into four categories: Flat protocols ([2, 18, 20]), Grid-based protocols ([113]), Cluster-based protocols ([3, 15]) and Group-based protocols ([28]).

3.6.1 Flat protocols

SPAN

Chen et al. [20] propose SPAN, a power saving topology control algorithm for wireless ad hoc networks that adaptively elects coordinators from all nodes to form a routing backbone and turn off other nodes' radio receivers most of the time to save energy. SPAN is a distributed, randomized algorithm where each node makes a local decision

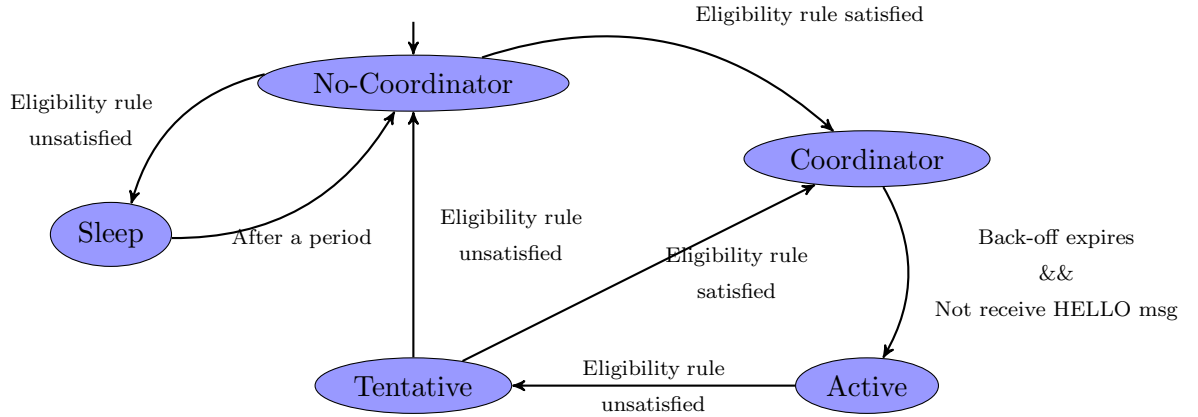


Figure 3.5: SPAN state transitions.

on whether to sleep, or to join a forwarding backbone as a *coordinator* based on an estimate of how many of its neighbors will benefit from it being awake, and the amount of energy available to it. SPAN coordinators stay awake continuously and perform multi-hop packet routing, while other nodes remain in sleep mode and periodically check if they should wake up and become a coordinator. SPAN achieves the four following goals [20]:

1. It ensures that enough coordinators are elected so that every node is in the radio range of at least one coordinator.
2. It rotates the coordinators in order to ensure that all nodes share the task of providing global connectivity roughly equally.
3. It attempts to minimize the number of nodes elected as coordinators, thereby increasing network lifetime, but without suffering a significant loss of capacity or an increase in latency.
4. It elects coordinators using only local information in a decentralized manner; each node only checks state stored in local routing tables during the election process.

Periodically, a non-coordinator node determines if it should become a coordinator or not. The following coordinator eligibility rule is used in SPAN:

A non-coordinator node should become a coordinator if it discovers, using only information gathered from local broadcast messages, that two of its neighbors cannot reach each other either directly or via one or two coordinators. Figure 3.5 shows node state transitions in SPAN.

Coordinators election process

Coordinators are elected based on local information supplied by a routing protocol.

In SPAN, each node periodically broadcasts HELLO messages that contain the node's status (i.e., coordinator or not), its current coordinators, and its current neighbors. From these HELLO messages, each node maintains a neighbor table which contains a list of its neighbors and coordinators, and for each neighbor, a list of its neighbors and coordinators. This neighbor table is maintained by each node periodically broadcasting a HELLO message, or a message simply stating the node is alive. A node uses information from its neighbor table to determine if it should announce or withdraw itself as a coordinator.

In SPAN, when multiple nodes discover the lack of a coordinator at the same time, they decide to become coordinators which leads to an announcement contention. To resolve contention, SPAN delays coordinator announcements with a randomized back-off delay. Thus, each node chooses a delay value, and delays the HELLO message that announces the node's volunteering as a coordinator for that amount of time. The back-off delay, in SPAN, is defined according to equation (3.1) that takes into account two factors: (i) the amount of residual energy, and (ii) the number of nodes that will benefit from it being active.

$$Delay = \left(1 - \frac{E_r}{E_m}\right) + \left(1 - \frac{C_i}{\binom{N_i}{2}}\right) + R \times N_i \times T \quad (3.1)$$

where, N_i is the number of the neighbors of node i , C_i is the utility of node i , which represents the number of node pairs among its neighbors that will be connected if i becomes a coordinator, E_r is the remaining energy at node i , E_m is the maximum amount of energy available (initial energy) at node i , R is a random value picked from the interval $[0,1]$, and T is the round-trip delay for a small packet over the wireless link.

At the end of the delay, the node reevaluates its eligibility based on HELLO messages recently received, and makes its announcement if and only if the eligibility rule still holds.

In SPAN, each coordinator periodically checks if it should withdraw as a coordinator. A node should withdraw if every pair of its neighbors can reach each other either directly or via one or two other coordinators. In order to rotate the coordinator role among all nodes fairly, after some period of time, a coordinator marks itself as a tentative coordinator. A coordinator node can become a tentative coordinator if every pair of its neighbors can reach each other via one or two other neighbors, even if those neighbors are not currently coordinators. A tentative coordinator can still be used to forward packets.

SPAN provides energy saving while preserving network capacity and decreasing latency. However, the election algorithm does not provide the minimum number of coordinators required to merely maintain connectivity, it ensures that every populated radio range in the entire network contains at least one coordinator [20]. SPAN's election algorithm requires each node to advertise its coordinators and neighbors, if it is

a coordinator, a tentative coordinator, or a non-coordinator. Consequently, SPAN incurs a high communication overhead as the routing tables are exchanged periodically. Furthermore, SPAN does not guarantee a connected topology; nodes decide to sleep or join the network based on connectivity information supplied by a routing protocol [116].

ASCENT: Adaptive Self-Configuring sE nsor Networks Topologies

In order to reduce the energy consumption and ensure the network connectivity, ASCENT [18] exploits node redundancy and adaptively elects active nodes from all nodes in the network. Active nodes stay awake all the time and perform multi-hop packet routing, while the other nodes remain passive (or idle) and periodically check if they should become active.

Initially, only some nodes are active. The other nodes remain passively listening to packets but not transmitting. If the sink node gets very high packet loss from the source due to poor connectivity, it starts sending *help messages* to signal neighbors that are passive to join the network. When a neighbor node receives a help message, it may decide to join the network and becomes active. The existence of a new active neighbor is signaled to other passive neighbors by sending a *neighbor announcement message*. This process continues until the number of active nodes is such that the message loss rate experienced by the sink is below a pre-defined threshold. The process will re-start when some future network events (e.g., a node failure) or a change in the environmental conditions cause an increase in the message loss. On the other hand, if a node experiences high loss rates due to collisions, then it may decide to go to sleep mode.

In ASCENT, nodes are in one of four states: sleep, passive, test, and active, as shown in Figure 3.6. Initially, a node starts in the test state. Each node in the test state, sets up a timer T_t , and sends neighbor announcement messages. When T_t expires, the node enters the active state. Otherwise, if the number of active neighbors is above the neighbor threshold (NT), or if the average data loss rate (DL) is higher than the average loss (loss T_0), then the node moves into the passive state. The intuition behind the test state is to probe the network to see if the addition of a new node may actually improve connectivity. When a node enters the passive state, it sets up a timer T_p and sends new passive node announcement messages. This information is used by active nodes to make an estimate of the total density of nodes in the neighborhood. Active nodes transmit this density estimate to any new passive node in the neighborhood. After the expiration of T_p , the node enters the sleep state. If before T_p fires the number of neighbors is below NT, and either the DL is higher than the loss threshold (LT) or DL is below the loss threshold but the node received a help message from an active neighbor, it makes a transition to the test state. The objective behind the passive state

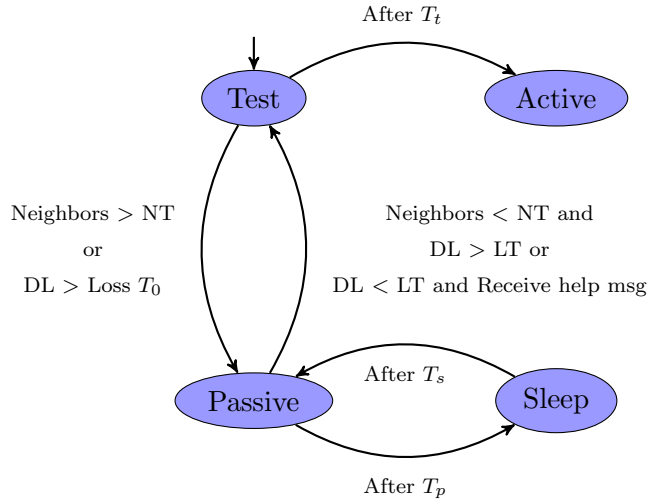


Figure 3.6: ASCENT state transitions.

is to gather information regarding the state of the network without causing interference with the other nodes. Nodes in the passive and test states continuously update the number of active neighbors and data loss rate values. Energy is still consumed in the passive state, since the radio is still on when not receiving packets. A node that enters the sleep state turns the radio off, sets a timer T_s and goes to sleep. When T_s expires, the node moves into passive state. Finally, a node in active state continues forwarding data and routing packets until it runs out of energy. If the data loss rate is greater than LT , the active node sends help messages.

Unlike SPAN, ASCENT is independent of the routing protocol and has the potential for significant reduction of packets loss. It limits the packets loss due to collisions because the node density is explicitly taken into account as a parameter (neighbor threshold value). The protocol has good scalability properties as well as increases energy savings. However, energy saving does not increase proportionally with the node density because it actually depends on passive-sleep cycle rather than the number of active nodes. In addition, maintaining the loss rate and node density may incur high communication overhead and adversely affect the node's decision of going to sleep or active. Another ASCENT's disadvantage is it has too many parameters to be configured, which make it difficult to be optimized [4]. Moreover, due to the fact that ASCENT mainly relies on the number of active neighbors, there might be situations where the network is partitioned [15].

OTC

Similar to SPAN and ASCENT, OTC [2] is another flat topology control protocol which selects a group of active nodes from all nodes in the network. Active nodes stay awake continuously and perform multi-hop packet routing while the other nodes remain in

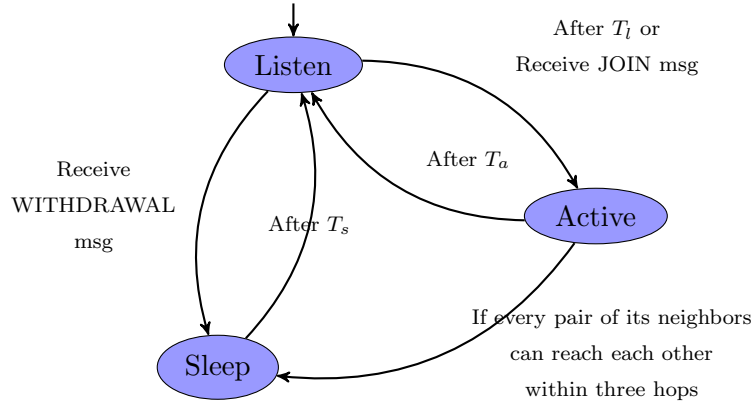


Figure 3.7: OTC state transitions.

power-saving mode, and periodically check if they should wake up and exchange their roles with the active node. OTC is built on the notion that when a region of a WSN has a sufficient density of nodes, significant energy saving is obtained by allowing redundant nodes to sleep. Using the two-hop neighborhood information, selected nodes sequentially select a subset of nodes to be active among all nodes in the neighborhood to ensure connectivity. OTC includes three mechanisms: neighbor discovery, role alternation, and selecting active nodes.

The discovery process allows a node to discover its neighborhood. So, each node periodically broadcasts a HELLO message containing: (i) Node ID, (ii) Message type (HELLO, JOIN or WITHDRAWAL messages), (iii) Node's status, (iv) Active list and (v) Neighbor list.

Upon reception of HELLO messages, every node in the network knows its neighbor set and for each neighbor its set of neighbors. After that, the active node selection process is launched which involves two main parts. The first part concerns the selection of an initiator called seed node (i.e., the first node to run the algorithm). The second part involves this node to recursively select its one hop neighbors to cover the entire network, while ensuring redundant nodes are made to sleep.

The selection of the seed node depends on its available energy and node degree. A node with more residual energy and higher node degree can be selected as seed node. Once a node has its one-hop and two-hop neighbor information, it can then select a minimum number of one-hop neighbors which covers all its two-hop neighbors. On the other hand, each node in the network selects a set of nodes from its one-hop neighborhood. This set of selected neighbor nodes is called its active list. This set is selected such that the node is able to reach all its two-hop neighbors via the active list. Once this list is formed, a node announces it by broadcasting JOIN message.

In OTC, a node can be in one of three states: sleep, listen and active. Figure 3.7 shows the state transition diagram of a node. Initially, all nodes start out in the listen

state. A node, in the listen state, turns on its radio and exchanges HELLO messages to gather information about its neighborhood. In addition, when a node enters the listen state, it sets up a timer T_l . When T_l expires, if the node did not receive an announcement message from its neighbor nodes, the node enters the active state. If before T_l expires, the node receives a WITHDRAWAL message (i.e., its node ID is not found in the sender's active list) from its neighbors, then the node turns off its radio and moves into the sleep state. Otherwise, if the node received a JOIN message (i.e., its node ID is found in the sender active list), then the node moves into active state and broadcast its own active set. When the node enters the active state, it sets a timeout value T_a to determine how long it should stay active. After T_a fires, the node moves back into the listen state. A node in the active state periodically checks if it should turn its radio off, and move into the sleep state. This decision is based on the following eligibility rule; it checks whether every pair of its neighbors can reach each other within two hops. A node delays its withdrawal announcement with a randomized back-off delay. When the back-off delay timer expires, the node reassesses its withdrawal eligibility. If the withdrawal it is still valid, it announces its withdrawal (i.e., sending an announcement message with status field value is set to nonactive) and transits to the sleep state. When transiting to the sleep state, a node cancels all timers, sets the sleep timer T_s and turns off its radio. A node in the sleep state returns to the listen state after an application dependent sleep time T_s .

OTC increases network lifetime while maintaining connectivity, guaranteeing multi-hop reachability from any source to any destination with a reasonable throughput. Moreover, to ensure fairness, the role of active nodes is rotated periodically to ensure energy-balanced operations. However, sending periodically the announcement messages increases the communication overhead. In addition, the period of sleeping is left to the application without taking into consideration the remaining energy of neighbors that are active. This situation can lead to network partitioning once the active nodes deplete their energies.

3.6.2 Grid-based protocols

GAF: Geographical Adaptive Fidelity

Geographical Adaptive Fidelity (GAF) [113] is a location-driven protocol that conserves energy by identifying nodes that are equivalent from a routing perspective and then turning off unnecessary nodes, keeping a constant level of routing *fidelity*. Based on the physical location, GAF identifies node redundancy and group equivalent nodes into *virtual grids* in such a way any two nodes from adjacent grids can communicate with each other and so only one active node in each grid is enough to maintain a connected network. Each GAF node uses location information (provided by GPS) to associate itself with a virtual grid, where all nodes in a particular grid square are equivalent

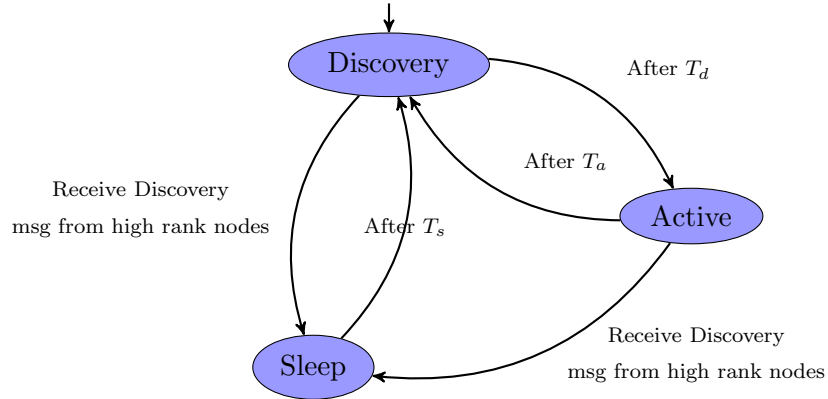


Figure 3.8: GAF state transition diagram.

with respect to forwarding packets. In order to meet the definition of virtual grid, the distance between two possible farthest nodes in any two adjacent grids, must not be larger than the nominal radio range R . So, the virtual grid size r is determined based on the radio range using the equation (3.2).

$$r^2 + (2r)^2 \leq R^2 \quad \text{or} \quad r \leq R/\sqrt{5} \quad (3.2)$$

In GAF, the nodes can be in one of three states: sleep, discovery, and active. A state transition diagram is shown in Figure 3.8. Initially, all nodes start out in the discovery state. When in the discovery state, a node turns on its radio and exchanges discovery messages - a tuple of node id, grid id, estimated node active time (enat), and the node state - to find other nodes within the same grid. A node determines the grid id using its location and the grid size information. When a node enters the discovery state, it also sets up a timer T_d . When the timer T_d expires, the node broadcasts its discovery message and enters the active state. This timer can, however, be terminated by other discovery messages received from node with high remaining energy (high ranking). It also reduces the probability of discovery message collision.

When a node enters the active state, it sets up a timeout value T_a to define how long this node should stay in the active state. GAF uses a load balancing strategy so that all nodes in the network are equally important. After a node remains in the active state for time T_a , it returns to the discovery state to give a chance to other nodes within the same grid to become active. While a node in the active state, it periodically broadcasts its discovery message at T_d intervals.

A node in the discovery or active state can transit to sleep state when it can determine some other equivalent node will handle the routing function. When transiting to sleep state, a node cancels all timers, sets the sleep timer and turns off its radio. A node in the sleeping state wakes up after a sleep time T_s and moves into discovery state.

The attractive feature of GAF is that it does not incur a high communication overhead as grids can be built statically once the communication range is known, but the assumption that the communication range is a perfect disc is far from reality. In GAF load balancing is achieved through a periodic re-election of the leader, i.e., the node that will remain active to manage routing in the virtual grid. The leader is chosen through a rank-based election algorithm which considers the nodes' residual energy, thus allowing the network lifetime to increase in proportion to node density. However, GAF may allow a node to sleep even if that node is actively participating in routing. This may cause interruptions in communication (loss of connectivity), which incurs high packet loss, and increase routing latency [34, 4].

3.6.3 Cluster-based protocols

TECA: A Topology and Energy Control Algorithm

Busse et al. [15] proposed a Topology and Energy Control Algorithm called TECA that is motivated by a common clustering approach. In clustering approach, each node is assigned to a cluster with one master node that acts as a cluster head. Once the network is divided into several clusters, TECA selects some nodes that act as bridges between two or more clusters. In doing so, a connected backbone topology is constructed. Using the concept of *virtual links*, passive nodes decide in a distributed way if they should join the topology and stay active or rather go into a low-powered sleep mode.

As shown in Figure 3.9, a sensor node running TECA can be in one of five states: initial, sleep, passive, bridge, or cluster head. After a node is powered on, it is in the initial state with its radio turned on until a timer T_i expires. In this state, nodes overhear packet transmissions, build a neighborhood table, and measure link qualities to adjacent nodes. After time T_i , a node changes its state to passive. Like in the initialization state, passive nodes overhear ongoing packet transmissions and keep their neighborhood table up-to-date. Additionally, in case the topology is still disconnected, they become active, either as a cluster head or as a bridge. Otherwise, they stay passive for a time T_p until they go to sleep to save energy.

Since TECA conserves energy by putting redundant nodes to sleep, a major challenge is to maintain connectivity in the network. So, after clustering the entire network, TECA maintains connectivity by selecting nodes as bridges connecting different clusters. The topology built by TECA is based on neighborhood information that nodes exchange periodically. These beacons are broadcasted by non-sleeping nodes each time an announcement timer T_a expires. Beacons contain the node's id, state, residual energy, a timeout value, and one hop neighborhood information. However, only information about active neighbors, i.e., cluster heads and bridges, are required and included into the packet.

TECA runs into two phases: cluster heads selection and bridges selection. The

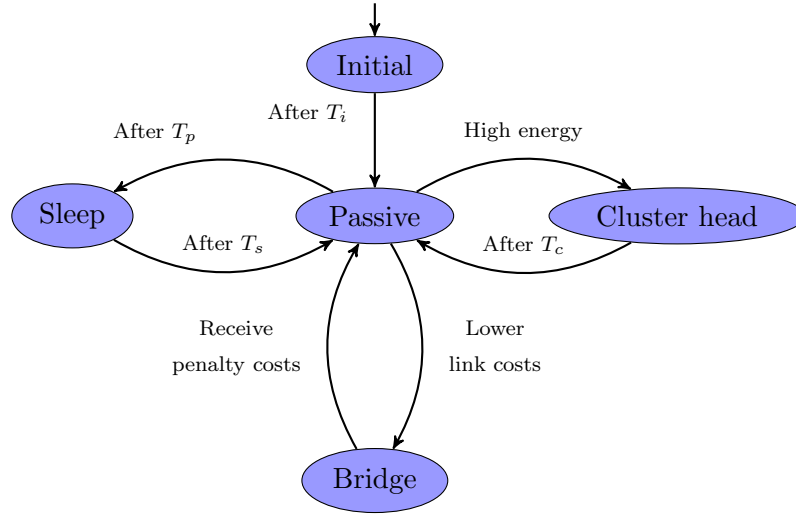


Figure 3.9: TECA state transitions.

clustering selection process is priority-based, in this case based on the remaining energy at the nodes only. Initially, each node announces itself as cluster head candidate, which propagates to its neighborhood. Based on one hop neighborhood information, the best cluster head can be found, using some cluster selection value, e.g., the node's residual energy. This node then becomes a cluster head. All other nodes in the cluster head's one hop neighborhood are assigned to it and are no longer cluster head candidates. With that mechanism, any node is finally either a cluster head itself or assigned to one. When a node becomes a cluster head, it remains active during a time period T_c . If T_c expires, it tries to find another cluster head it could join. The running time of T_c is defined based on their energy and a factor α , called cluster timeout factor according to the equation (3.3).

$$T_c = \min\{\alpha, energy_i\} \cdot E_i^{init} \quad (3.3)$$

with $\alpha \in [0, 1]$ called the cluster timeout factor, E_i^{init} is the initial energy of a node i , and $energy_i$ being its residual energy.

After the cluster head selection process, all non-cluster heads remain passive until their passive timer T_p expires. During this phase they listen to announcement packets of their neighbors. If a passive node is aware of the existence of two or more clusters, the node will become a bridge candidate. Determining which of these nodes finally become bridges is a great challenge. There are three main requirements that should be considered. Bridges must connect different clusters in an optimal way, i.e., (i) the packet loss rate between clusters should be minimized, (ii) the connection should be long-lived, and (iii) the number of bridges should be minimized in order to extend the operational lifetime of the network. So, TECA computes a local *Minimum Spanning Tree* between the cluster heads to find the optimal path to interconnect them. The

optimal path is found by using *links costs* based on the packet loss rate, the lifetime of a link, and a penalty function to further minimize the number of nodes. Finally, the nodes along this optimal path become bridge nodes and the cluster heads are connected.

After selecting cluster heads and bridge nodes, all remaining nodes stay passive until their passive timer T_p expires. If T_p fires, a passive node will go to sleep with its radio turned off to save energy. However, in order to rebuild the topology, the node must wake up after its cluster timeout, which is received during the cluster formation process. In addition, the node must consider the timeout from other cluster heads and active bridges in its neighborhood and wakes up in time.

The role of the cluster head and bridge node is rotated and residual energy is considered in cluster head selection. These attributes of TECA help prolonging the network lifetime. An advantage of TECA over many other clustering algorithms is the use of bridge nodes to maintain network connectivity. However, the use of bridge nodes also has a drawback; it requires more nodes to remain active, which will use more energy and hence, shorten the lifetime of the network [34]. Moreover, the communication overhead is much higher during the cluster forming process (Cluster heads and bridges selection) which leads to large energy consumption.

ECTC: Energy effiCient Topology Control

Like TECA, ECTC (Energy effiCient Topology Control) [3] is a hierarchical distributed topology control algorithm, which uses a clustering approach. It is built on the notion that when a region of a wireless sensor network has a sufficient density of nodes, significant energy saving is obtained by allowing redundant nodes to sleep. Using the two-hop neighborhood information, some nodes are selected as cluster heads and others act as bridges. Thus, the entire network gets connected. ECTC operates into two phases. The first phase is similar to OTC [2], in which a group of active nodes is selected from all nodes in the network to form the forwarding backbone.

After the active nodes are selected, the second phase of the algorithm is to select a number of these active nodes to serve as Cluster heads while the rest act as bridges between two or more Cluster heads. Initially, each active node announces itself as Cluster head. Announcement contention may occur when all active nodes decide to be a cluster head at the same time. ECTC resolves this contention by delaying cluster head announcements with a randomized back-off delay. Each node chooses a delay value based on its available energy and node degree. A node with more residual energy and higher node degree generates a smaller back-off delay; a node delays its announcement for that amount of time. At the end of the delay, the node broadcasts itself as a cluster head to its one-hop neighbors. Upon receiving this advertisement, any other active nodes join the cluster. These non-Cluster head nodes inform the appropriate Cluster heads that they will be members of the cluster. If before the back-off delay

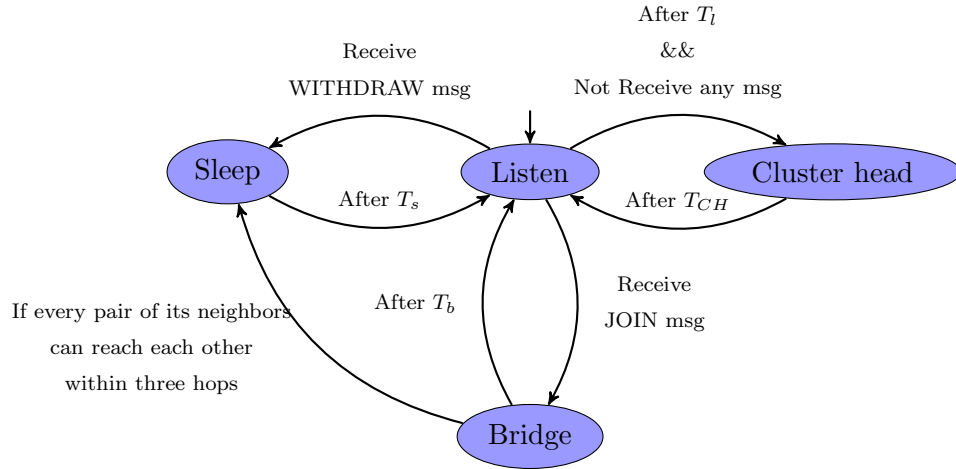


Figure 3.10: ECTC state transitions.

time expires, an active node receives a Cluster head announcement message from its one-hop neighbor it cancels its back-off delay timer and announces itself as a bridge node connecting two or more Cluster heads.

In ECTC, a node can be in one of four states, as shown in figure 3.10: sleep, listen, cluster head and bridge. Initially, all nodes start out in the listen state. When in the listen state, a node turns on its radio, sets up a timer T_l and exchanges HELLO messages to gather information about its neighborhood. When T_l expires, if the node did not receive any message from its neighbor nodes, the node enters the cluster head state and set a timeout value T_{CH} . After T_{CH} expires, the node moves to listen state. If before T_l fires, the node receives a WITHDRAW message from its upstream neighbor, the node turns off its radio and moves into the sleep state, or if the node received a JOIN message, then it moves into bridge state. When the node enters the bridge state, it sets a timeout value T_b to determine how long it should stay active. At the end of T_b , the node moves back into the listen state. A node in the bridge state periodically checks if it should turn its radio off, and move into the sleep state. This decision is based on the following eligibility rule: it checks whether every pair of its neighbors can reach each other within three hops. A node delays its withdrawal announcement with a randomized back-off delay. When the back-off delay timer expires, the node reassesses its withdrawal eligibility. If the withdrawal is still valid, it announces its withdrawal and transits to the sleep state. When transiting to the sleep state, a node cancels all timers, sets the sleep timer T_s and turns off its radio. A node in the sleep state returns to the listen state after an application dependent sleep time T_s .

ECTC shares the same advantages and drawbacks as OTC and TECA, described above.

3.6.4 Group-based protocols

CPA: a Connectivity-based Partition Approach

In [28], an adaptive partitioning scheme for node scheduling and topology control, is presented with the aim of reducing energy consumption. The authors formulate the node scheduling with topology control as a constrained optimal graph partition problem, and propose a Connectivity-based Partition Approach (CPA), which is a distributed heuristic algorithm. CPA partitions sensors into groups, based on the measured connectivity between pairwise nodes, such that a connected backbone network can be maintained by keeping only one arbitrary node from each group in active status while putting others to sleep. The proposed partition should satisfy the following constraints [28]:

1. Any node is within one hop away from all the other nodes in the same group. Under such a constraint, each node can be covered by the communication backbone; that is, each node is either in the backbone network if it is an active node or directly connected to the backbone network if it is a sleeping node.
2. The backbone network formed by active nodes at each snapshot must satisfy some connectivity properties such that it does not suffer a significant loss of communication quality as compared with the original network.
3. In order for all the groups to remain alive together as long as possible, the energy needs to be evenly distributed among groups. So, the uniform energy consumption, for sensor nodes, is assumed.
4. A smaller number of groups is preferred without degrading the communication quality of the original network, because more energy conservation can be achieved by decreasing the number of active nodes at each time.

CPA is a distributed iterative process which aims to find an M -partition while satisfying that any backbone graph is at least K -connected. It starts from the initial partition where each node forms a unique group. CPA continuously merges two groups into a larger one until further merging will break the constraints of the problem aforementioned. In CPA, there are two kinds of nodes in each group: ordinary nodes and a head node. Each kind of node maintains its node ID and associated group information, including its group ID, the IDs of other group members, and the ID of the head node in its group. One head node is selected from each group to maintain some additional information on the connectivity between its group and the neighboring groups in the current M -partition. CPA goes through a group merging process iteratively before it gets to the final partition.

In the group merging process, the head nodes of each two completely adjacent groups³ exchange group connectivity information to decide whether their groups should merge. Only completely adjacent groups can merge so that the new group is also a clique. Contentions may occur when multiple neighboring groups want to merge simultaneously. To deal with, a randomized back-off delay is used when the two groups announce their willingness to merge. If no contention is observed at the end of the delay, the two groups about to merge will announce their decision to all of their neighboring groups. In order to obtain a good partition, CPA considers several factors to decide which two groups are preferred to be merged first in the current partition. These factors can be reflected as a utility function, noted f as shown in (3.4) for two groups A_i and A_j :

$$f_{ij} = k_1 \left(1 - \frac{|N_2(A_i) \cap N_2(A_j)|}{|N_2(A_i) \cup N_2(A_j)|} \right) + k_2 \left(\frac{g(A_i) + g(A_j)}{E_{total}} \right) \quad (3.4)$$

where $N_2(A_i)$ and $N_2(A_j)$ are the set of neighboring groups completely adjacent with groups A_i and A_j , respectively. $g(A_i)$ and $g(A_j)$ denote the energy in group A_i and A_j , respectively. E_{total} is the total energy of all nodes in the network, and k_1 and k_2 are two coefficients.

With the purpose to achieve K-connectivity, each group keeps track of its degree value during the group merging process, which is used to decide whether pairwise groups should be merged. If a group merge may cause the degree of some groups to drop below K, then these two groups will give up their intention to merge. The group merging process will be terminated when no groups can be merged. So, CPA guarantees that the generated backbone network, based on this partition, ensures the minimum degree (mindeg) of K which will be mindeg-connected, where:

$$mindeg = \min\{|N_2(A_i) \cap N_2(A_j)| \mid \cup \{deg(s) - 1/s \in N_2(A_i) \cup N_2(A_j)\}\} \quad (3.5)$$

where $deg(s)$ is the degree of node s .

Ding et al. [28], present two implementations of CPA: centralized and distributed. Figure 3.11 illustrates the state transition diagram of each head node for the distributed version of CPA.

Initially, each node constitutes a group (or is a head node). At the start, each node broadcasts an UPDATE_MSG to its neighbors containing its group information. In the Decision state, the head node calculates the priority value for each of its completely adjacent groups and selects the group with the highest priority to merge so that the fusion will not break the connectivity constraint. Let A_x be the current group and assume that the best group to merge with is A_y . If any completely adjacent group of A_x or A_y is in the Merging or Holding state, then the head node will go into the Waiting

³Two groups are completely adjacent if all nodes in the first group can communicate with all nodes in the second one and vice versa.

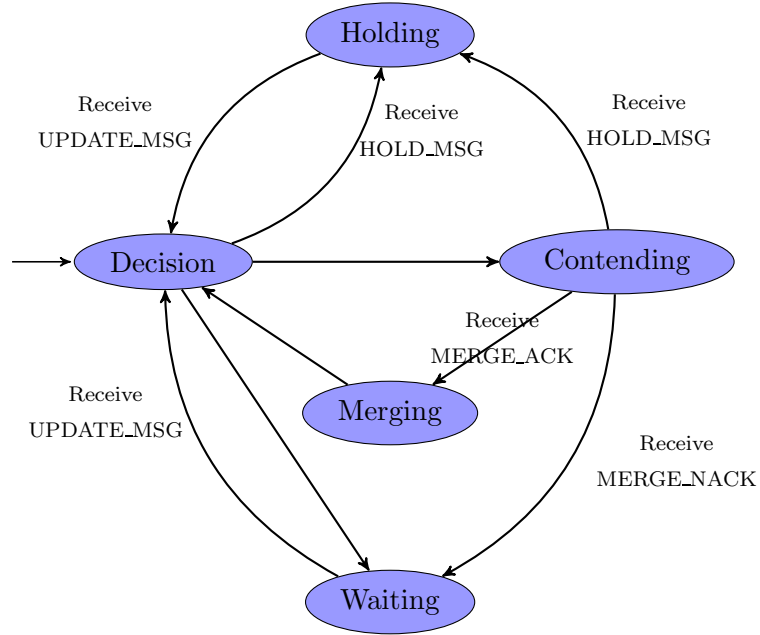


Figure 3.11: CPA state transitions.

state, because the merge of A_x and A_y will collide with the merge of some other groups. Otherwise, the head node will go into the Contending state. In this state, the head node will send a `MERGE_REQ` message to A_y , expressing its willingness to merge with it. If the contention succeeds (A_x receives a `MERGE_ACK` message from A_y), it goes into the Merging state, and otherwise (A_x receives a `MERGE_NAK` message from A_y), it enters the Waiting state. In the Merging state, it first broadcasts a `HOLD_MSG` to the neighboring groups of A_x and A_y to put them into Holding state and then updates its group information. After the merge is finished, it broadcasts an `UPDATE_MSG`, which contains the information of the newly merged group, to its neighboring groups. If a head node has not yet entered the Merging state, it will enter the Holding state when it receives a `HOLD_MSG` from neighboring groups. When a head node is in the Holding state or Waiting state, it will enter the Decision state again once it has received an `UPDATE_MSG`.

Once the group merging process is achieved, nodes in each group are switched between active and sleep status periodically. CPA uses a load balancing strategy to schedule nodes in each group. It assumes that all the nodes in each group can be synchronized. In order to elect an active node, each group member broadcasts a message to the whole group stating its willingness to become active. Each node waits for a certain time delay before its announcement. The earliest announcement will suppress the others so that the corresponding node will become the active node in the group. The time delay for each node is set to be inversely proportional to its residual energy. Therefore, the node with the maximum residual energy will be selected. Then, the selected active node informs other nodes of the time it will remain active, after which

all the nodes need to reselect an active node again within the group.

One of the best characteristics of CPA is that the partition is based on the measured connectivity of nodes instead of guessing connectivity by their positions. CPA can generate partitions while ensuring K -connectivity of the backbone network even under unideal radio propagation models. However, the proposed scheme suffers from several problems: (1) the collection of the total energy of all nodes in the network, increases the complexity of the proposed solution in terms of communication and run time, (2) a large communication overhead will be produced during the merging process, which leads to a high energy consumption, and (3) a deadlock problem can occur during the merging process [74, 92].

3.7 Discussion

As described above, several algorithms have been developed to extend the network lifetime by managing the network topology. These protocols generally share the same objectives which are energy saving and connectivity. All aforementioned topology control protocols aim to construct reduced topologies with certain desired properties, in distributed and localized manner. Table 3.3 summarizes the key features, objectives, major assumptions of the existing sleep scheduling-based topology control protocols. The category of each protocol is identified and advantages and disadvantages are included.

Most previously proposed algorithms which are connectivity-based can be divided into two categories: flat and hierarchical protocols. The main drawbacks of the flat protocols are (1) it is difficult to provide a minimum number of active nodes in each state of the network, and (2) the formed topology is not always connected. This due to the fact that nodes decide to be active regardless of the decision of the other nodes; they decide to sleep or join the network based on either connectivity information supplied by a routing protocol or the loss rate and node density which can affect the integrity of node's decision of going to sleep or active. In addition, the communication overhead is high (about $O(n^2)$) for most of flat protocols. On the other hand, protocols in the second category, organize nodes either in grids, clusters or groups.

Clustering solutions require the election of two types of nodes: cluster-heads and bridges. These types of nodes must remain active to ensure connectivity while other nodes can go into energy conservation mode, by turning off their radio (sleep state). Therefore, these protocols do not provide a good conservation of energy that directly affects the network lifetime. First, these protocols require an important message exchange at every election phase for the choice of cluster-heads and then of the gateways. Second, the number of active nodes in the network is high, as two nodes, at least, per cluster need to be activated. Therefore, reducing the number of active nodes greatly extends the network life without sacrificing connectivity. Solutions organizing nodes

into grids or groups so that each grid/group is represented by only one active node at each network state have been already proposed. The grid-based protocols form grids based on location information while group-based ones form groups based on connectivity information.

The attractive feature of grid-based protocols is that they incur a low communication overhead (about $O(n)$) as grids can be built statically once the communication range is known. However, the assumption that the communication range is a perfect disc is far from reality. Moreover, these protocols require the availability of GPS at each node to determine the location information, which is sometimes impossible to acquire.

Group-based topology control protocols use the communication equivalence to organize nodes into groups. For example, in CPA [28], the nodes are organized into groups based on the degree of connectivity between nodes. Thus, a small number of groups formed while satisfying K -connectivity. However, some limitations are enumerated such as: (1) the complexity of the proposed solutions in terms of communication and run time, is very high (about $O(n^3)$), (2) a connectivity loss or partitioning in the network is possible, and (3) the sending of a merge request is done cyclically and each node waits for a response from the other candidate; which creates a deadlock situation and the group forming process will never end.

In summary, we can notice that some proposals offer low message overhead but rely on unrealistic assumptions, while some other solutions provide a small number of active nodes but incur high message overhead. Moreover, when running some proposed protocols, the formed topology is not always connected (e.g., CPA [28]). Indeed, it is difficult to ensure connectivity without increasing the number of active nodes. So, there is a trade-off between power consumption, connectivity and the number of active nodes. If the number of active nodes is minimized, the network connectivity may be lost; And if connectivity is ensured when an acceptable number of active nodes is reached, an additional energy is expended.

Table 3.3: Comparative study of topology control algorithms based on duty cycling mechanism.

| Protocols | Category | Objectives | Major Assumptions | Load Balancing | Connectivity degree | Communication overhead | Key characteristics | Drawbacks |
|-------------|----------|--|---|----------------|---------------------|------------------------|--|---|
| SPAN [20] | Flat | Connectivity Energy saving Mobility | The presence of an ad hoc polling mechanism. | Yes | K | High | Distributed algorithm No location information It depends of routing protocol. | A connected topology is not guaranteed. The minimum number of coordinators is not ensured. |
| ASCENT [18] | Flat | Packet Delivery Connectivity Energy saving | No partitions initially Losses Detection mechanism exists. CSMA and TDMA are available. | No | K | High | Distributed algorithm No location information Scalable Packet loss, active nodes and residual energy are used to activate or turn off some nodes. | Too many parameters to configure. Network partitioning. The minimum number of active nodes is not provided. |
| OTC [2] | Flat | Energy saving Connectivity | No location information The network graph is not a Unit Disk Graph. | Yes | K | High | Distributed algorithm Two-hop neighborhood is used to select active nodes. No location information | Computation time is high. The sleep periods are not defined according to the remaining energy of active neighbors. |

Table 3.3: Comparative study of topology control algorithms based on duty cycling mechanism (Continued).

| Protocols | Category | Objectives | Major Assumptions | Load Balancing | Connectivity degree | Communication overhead | Key characteristics | Drawbacks |
|-----------|---------------|---|--|----------------|---------------------|------------------------|---|---|
| GAF [113] | Grid-based | Energy saving Connectivity Mobility | Ideal radio propagation model | Yes | 4 | Low | Distributed algorithm Simple and Scalable It is independent of routing protocols | It needs GPS which consumes an extra energy. The assumption of an ideal radio propagation model is not realistic. It may cause a loss of connectivity. |
| TECA [15] | Cluster-based | Energy saving Connectivity End-to-end delay | Nodes have the same transmission range Link loss information is available | Yes | K | Very High | Distributed algorithm Packet loss and residual energy are considered to select CHs and bridges. No location information | Computation time is high. It requires more active nodes in each cluster. The maintenance of the clustering structure is expensive in terms of messages. |
| ECTC [3] | Cluster-based | Energy saving Connectivity | Collision and Error-free | Yes | K | Very High | Distributed algorithm Two-hop neighborhood is used to select active nodes. No location information | Computation time is high. The sleep periods are not defined according to the remaining energy of active neighbors. It requires more active nodes in each cluster. The maintenance of the clustering structure is expensive in terms of messages. |

Table 3.3: Comparative study of topology control algorithms based on duty cycling mechanism (Continued).

| Protocols | Category | Objectives | Major Assumptions | Load Balancing | Connectivity degree | Communication overhead | Key characteristics | Drawbacks |
|-----------|-------------|-------------------------------|--|----------------|---------------------|------------------------|--|---|
| CPA [28] | Group-based | Connectivity Energy saving | Uniform energy consumption Time synchronization | Yes | K | Very High | Centralized and Distributed algorithm It ensures K-vertex connectivity of the backbone network under regular and irregular radio propagation models. No location information | The complexity is high in terms of communication and run time. It may cause a loss of connectivity. A deadlock problem can occur. |

3.8 Conclusion

In this chapter, we conducted a comprehensive study on topology control issues in WSNs. We presented the motivation behind the need for topology control mechanisms, especially sleep scheduling-based topology control. We provided a classification for topology control protocols using duty cycling techniques to reduce the topology while ensuring the network connectivity. The chapter, also, provides a comparative study to investigate and evaluate the existing topology control protocols and understand some tradeoffs.

Finally, we may conclude that distributed and localized topology control protocols are preferred when they select a small number of active nodes with a low communication overhead and without sacrificing connectivity. In the next chapter, we will present our new proposed grid-based algorithm.

Chapter 4

GTC: A Geographical Topology Control Protocol

4.1 Introduction

In the previous chapter, we have presented the most relevant sleep scheduling-based topology control protocols for WSN. The key idea of these protocols is to manage the topology by identifying redundant nodes and schedule nodes for active or sleep modes in order to reduce energy consumption. Among these protocols, we found that grid-based topology control protocols have an attractive feature that is related to the reduced communication cost and the energy efficiency when determining the redundant nodes. Indeed, they do not incur a high communication overhead during the grid forming phase, as grids can be built statically once the communication range is known.

On the other hand, we noticed that most of the proposed protocols use a load balancing strategy which is achieved through a periodic re-election of the nodes that remain active to manage routing. This may lead to significant energy consumption during transitions between sleep and active modes. Moreover, it may increase the election frequency and the communication overhead and, as consequence, waste of energy.

For these reasons, in this chapter, we propose a new grid-based algorithm, called GTC: a Geographical Topology Control protocol [119, 120, 122], that benefits from the advantages of the existing grid-based algorithms to identify redundant nodes. GTC uses a new scheduling strategy which reduces the number of transitions between active and sleep states in order to minimize the transition energy and ensure the network longevity. It also minimizes the frequency of active nodes election and allows some stability in the topology that avoids a loss of connectivity.

In this chapter, first, we give the motivation of our work in Section 4.2. Then, the detailed description and analysis of the proposed algorithm are provided in Section 4.3. In Section 4.4, we evaluate our proposal by comparing it with GAF's approach, considered as a reference protocol. Finally, Section 4.5 concludes the chapter.

4.2 Motivation

One of the critical challenges in wireless sensor network is the energy conservation. In fact, sensor nodes are tiny battery-operated devices with limited energy. Recharging or replacing them may be impossible because the sensors can be deployed in a hostile environment. Typically, a wireless sensor can be in one of the four distinct states : Idle, Receive, Transmit, and Sleep. While it is expected that the radio consumes most of the energy in the Transmit and Receive modes, running in the Idle mode is also costly. Many measurements have shown that idle listening consumes 50 ~ 100% of the energy required for receiving. Thus, it is desirable to completely shut down the radio (enter into sleep mode) rather than transiting into the Idle mode.

As the energy consumed in sleeping state is smaller than the energy consumed in any other state, keeping sensor nodes in the sleep state is a good way to save energy. Since only a small set of nodes is involved in packet transmission and reception, in dense sensor network, it will be most effective to turn off the wireless interfaces of the redundant nodes. This can be accomplished by adopting a sleep scheduling approach, without sacrificing the network functionalities.

Most of sleep-scheduling protocols, which exploit the nodes redundancy, use a load balancing strategy to schedule nodes and allow them to be periodically active. However, as shown in Section 2.3.3, one of the major source of energy wastage is frequent switching between modes, especially switching from sleep to active mode that leads to significant energy consumption [10, 53, 93, 102, 117]. Indeed, few works have referred to energy waste, during the scheduling, because switching a radio on and off very frequently can sometimes result in even more energy consumption than leaving the transceiver unit in Idle mode. In addition, an active node election is carried out at each transition which consumes more energy and may disrupt communication.

Motivated by this, a distributed energy-conserving protocol called GTC (Geographical Topology Control) is proposed in this chapter. The ultimate goals of GTC are to (1) identify redundant nodes and organize them into zones (or grids) with a low communication overhead, (2) minimize the energy consumption, during the scheduling phase, by reducing the number of transitions between active and sleep states.

4.3 GTC design

In this section, we present our proposal for the sensor network model that consists of one (or more) sink(s) and a high number of sensor nodes, deployed over a large geographic area (sensing field). Data is transferred from sensor nodes to the sink node through a multi-hop communication paradigm [9]. Both the sink and sensor nodes are assumed to be static and all nodes have the same radio communication range. In addition, we don't consider the nodes' crash except that for batteries depletion.

4.3.1 GTC overview

Wireless sensor network is characterized by its density in number of deployed sensors. Generally, sensors in the same region collect and forward, the same information. In order to conserve nodes energy, it is necessary to keep a set of nodes active while others can sleep. We propose a distributed energy-conserving protocol (GTC) that exploits the sensors' redundancy, in the same region, in order to extend the network lifetime. So, in GTC, the whole network is divided into small zones. A zone is defined so that, for two adjacent zones Z_1 and Z_2 , all nodes in Z_1 can communicate with all nodes in Z_2 and vice versa. Thus, all nodes in each zone are equivalent for routing and only one node can be active in each one, while others can go to sleep mode.

GTC is a sleep-scheduling topology control protocol for a stationary wireless sensor network. It runs above the link and MAC layer and below the routing layer. GTC is not a routing protocol but a topology control protocol. It simply decides which nodes should join the routing infrastructure, in order to extend the network lifetime. The routing protocols are complementary to topology control protocols. Indeed, GTC runs with any routing protocol because it is completely independent. Our protocol works in two phases: Zones formation phase and Active nodes election phase.

Zones formation phase

In this phase, we use the same mechanism as the one considered in [113]. Each node determines its current location, provided by GPS or other location systems (for example, [29]). Based on location information and zone size, the node associates itself with a zone and determines the zone ID. The zone size r is calculated according to the radio communication range R as follows:

$$r^2 + (2r)^2 \leq R^2 \quad \text{or} \quad r \leq R/\sqrt{5} \quad (4.1)$$

The formula (4.1) guarantees that the nodes, located in the same zone, can communicate with each other. This equation also ensures that any two possible farthest nodes, in two adjacent zones, can communicate.

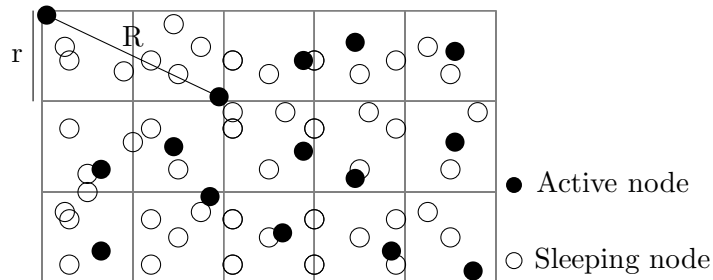


Figure 4.1: Zones formation.

Active nodes election phase

Once the zones formation phase is completed, nodes, in each zone, starts the stage of sleep/wake scheduling to select the active nodes. In this phase, only one node in each zone is required to be active while others turn their radio off. We select only one node to be active, in each zone, because all nodes in the same zone are equivalent for routing packets. This property is given by the way the topology is built. We use a *priority-based scheduling strategy* to schedule nodes in the same zone. This strategy will be described in Section 4.3.4. The election of an active node is based on its current energy level. Therefore, in each zone, the node with high energy level will be elected as active node of this zone. The elected node remains active until depletion of its battery while the others go to sleep mode during this period.

4.3.2 GTC state transitions

In GTC, nodes are in one of three states: *discovery*, *active*, and *sleep*. Figure 4.2 shows a state transition diagram and the election process is given in Figure 4.3.

Discovery state: Initially, all nodes start in the discovery state. When a node enters this state, it sets up a timer T_d (T_d is a random value), and sends a discovery message which contains the information needed to elect an active node: node id, zone id, energy level, and node state. If a node receives a discovery message from an active node, or from a node with a high energy level, before the expiration of T_d , it cancels the T_d timer and enters the sleeping state for T_s period. Otherwise, when T_d expires, a node enters the active state for T_a period and broadcasts a discovery message, in order to inform the neighbors that it is the elected active node for T_a period.

Active state: In active state, a node remains active and forwards data until the depletion of its battery.

Sleeping state: A node, in a sleep state, wakes up after the expiration of T_s timer and moves into a discovery state, in order to elect a new active node. There are two cases:

- The leader's remaining energy level is greater than a threshold Min_Turnofftime . In this case, the node returns to sleep mode for another T_s period.
- The leader's remaining energy level is less than a threshold Min_Turnofftime . In this case, the node does not return to sleep mode but sets a timer T_d and can be the new leader if it has a higher energy level than the others. In other words, a node can be the new active one if its energy level, is greater than that of the other nodes.

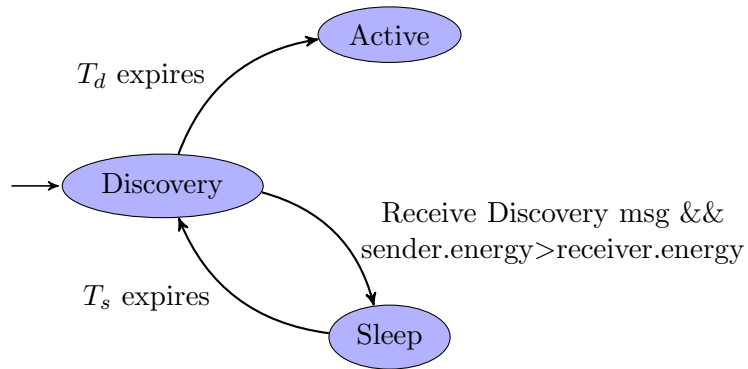


Figure 4.2: GTC state transitions.

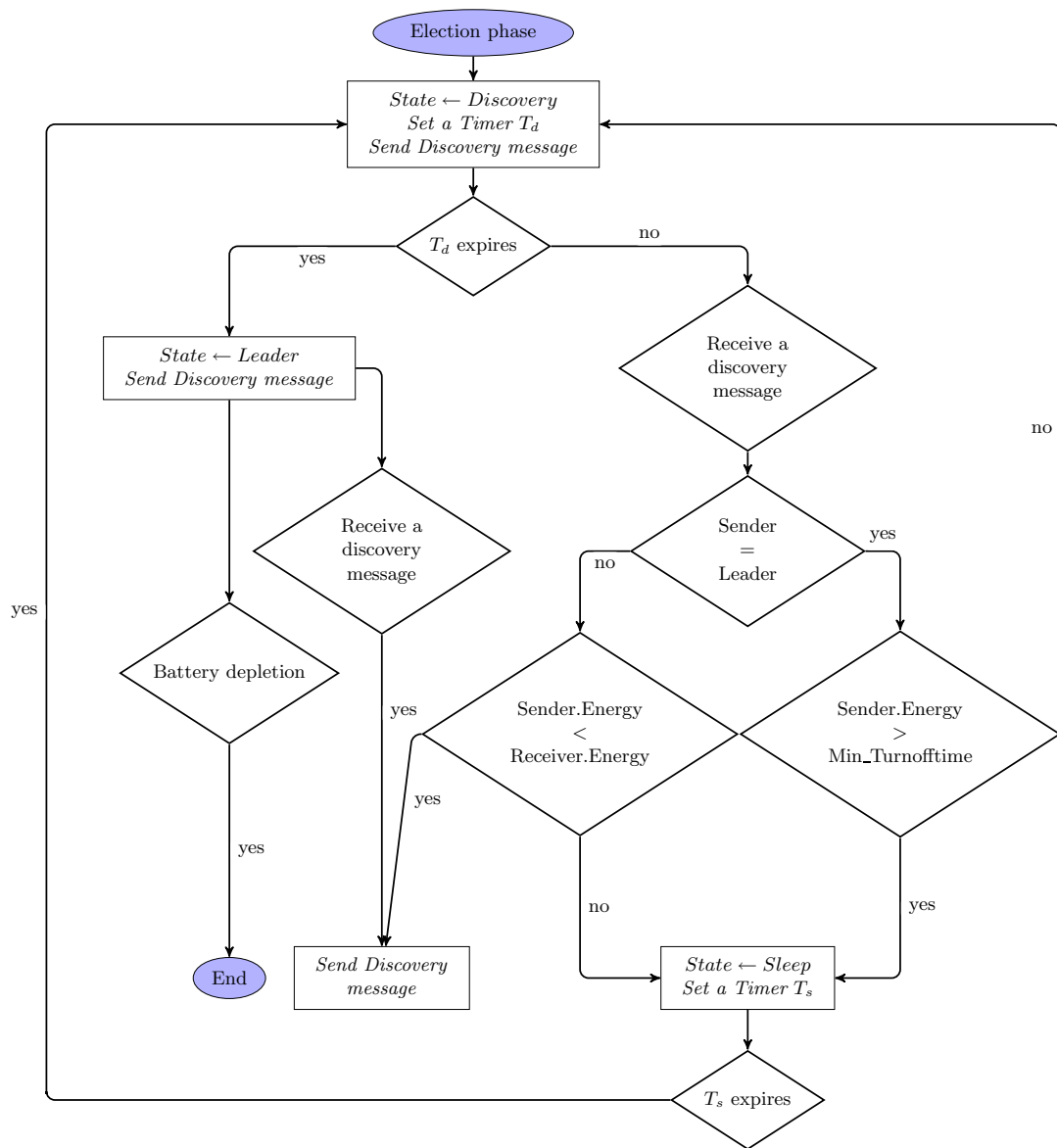


Figure 4.3: The flowchart of election phase.

4.3.3 GTC parameters tuning

In this section, we explain the parameter choices, made in the current GTC algorithm, but it should be noted that the choice of the parameters T_d , T_g , and Min_Turnofftime is left to the application.

Node active time (T_a) represents the leader's duration of activity. Initially, it sets to the expected node lifetime. The expected node lifetime is calculated, according to the following assumption: if a node functions while using a maximum of resources, how much this latter will hold before the exhaustion of its battery? In reality, the real value of T_a depends on the network traffic.

Discovery message interval (T_d) determines the maximum time a node remains in the discovery state. This parameter used to avoid node synchronization. It is chosen as a random value between two constants $\text{Min_}T_d$ and $\text{Max_}T_d$. We leave the application the choice of these constants.

Min_Turnofftime is a threshold that allows the active node to go on ensuring the packet routing, during a new leader's election. The connectivity loss can also be avoided.

Node sleeping time (T_s) is equal to the duration of the leader's activity minus a certain grace period T_g . T_s is chosen as a random value in the interval $[T_a - \text{max_}T_g, T_a - \text{min_}T_g]$. We note that if nodes wake up after T_s and the leader can ensure its activity (i.e. T_a is greater than Min_Turnofftime), nodes adjust their T_s to the remaining duration of the leader's activity minus grace period. We use the grace period to ensure that nodes must awake before the leader's battery depletion in order to avoid the connectivity loss.

Grace period (T_g) represents the sufficient amount of time to complete the process of a new leader's election. This parameter ensures that nodes wake up before the leader's remaining energy reaches the threshold.

We must note that, when selecting parameters T_d , T_g and Min_turnofftime , we must meet the following constraint, in order to avoid connectivity loss.

$$\text{max_}T_d < \text{min_}T_g \quad \text{and} \quad \text{max_}T_g \leq \text{Min_Turnofftime} \quad (4.2)$$

The following example illustrates this situation. Suppose that in a zone, a node S_i wakes up at time t and finds that the remaining energy of the active node allows it to operate for a period of 50s, for example. Two cases are to be discussed, depending on the values of selected parameters. Table 4.1 summarizes the two cases, discussed in what follows:

Case1: $\text{Min_turnofftime} < \text{max_}T_g$

A node S_i goes back to sleep mode because the leader's energy has not reached the threshold (i.e., $50s > 1.5s$). It chooses a random value of T_s in the interval $[48,49]$.

Table 4.1: Example.

| | min_T_g | max_T_g | $min_Turnofftime$ | min_T_d | max_T_d |
|--------------|------------|------------|--------------------|------------|------------|
| Case1 | 1s | 2s | 1.5s | 0s | 0.9s |
| Case2 | 1s | 2s | 2.5s | 1s | 2s |

Let's assume that the node S_i chooses the value $T_s = 48.5s$. After that period expires, it wakes to check if the leader can ensure packet routing. It finds that the leader's remaining energy has not reached the threshold and it may perform its functions during 2s. S_i still returns to the sleep state (because $2s > 1.5s$) for another period T_s chosen in the interval $[0,1]$. Set the node adjust its T_s to 1s. After 1s, it wakes up and finds that the active node energy allows it to operate for just 0.5s. The fact that this period is less than `min_turnofftime`, S_i launches a new leader's election phase and chooses the value of T_d in the interval $[0,0.9]$. If we suppose that it chooses the value 0.9, any node in the zone can ensure routing for up to 0.4s because the leader node dies after 0.5s and the node S_i must wait a period T_d before it enters the active state. This leads to a connectivity loss. For this reason, the value max_T_g should be less than `min_turnofftime`.

Case2: $max_T_d > min_T_g$

The node S_i goes back to sleep mode because the leader's energy has not reached the threshold (i.e., $50s > 2.5s$). It chooses a random value of T_s in the interval $[48,49]$. Let's assume that the node S_i chooses the value $T_s = 49s$. After that period expires, it wakes to check if the leader can ensure packet routing. It finds that the leader's remaining energy allows it to perform its functions during 1.5s. The fact that this period is less than `min_turnofftime`, it starts a new leader's election phase and chooses the value of T_d in the interval $[1,2]$. If we suppose that it chooses the value 2, any node in the zone can ensure routing for up to 0.5s because the leader node dies after 1.5s and the node S_i must wait a period T_d before it enters the active state. This leads to a connectivity loss. For this reason, the max value of T_d should be less than min_T_g .

4.3.4 Scheduling in GTC

After dividing a network into small zones, the election process of an active node can begin. The priority-based scheduling strategy, used by GTC, is based on nodes energy level. It consists on electing only one active node in each zone. We give a priority to nodes with high energy level. If there are nodes with the same energy, we use the node identity as selection criterion. In this case, the priority is given to the node having the highest identity.

Initially, each node compares its energy level with the other nodes in its zone. If

its energy is the highest, it becomes the leader of this zone because it has the highest priority. A node, in active state, continues forwarding data and routing packets until it runs out of energy. This strategy seems to be a very effective solution, for more energy saving, because it decreases the leader's election frequency. During the election process, nodes must exchange messages in order to elect an active node with a highest energy level. If we decrease the number of elections, we also decrease the communication overhead and as consequence save nodes' energy. The number of times that the leader's election process is carried out is equal to the number of nodes in a zone. Consequently, minimizing the number of a leader's election also minimizes the number of exchanged discovery messages, which leads to energy conservation.

On the other hand, if we use a load balancing strategy, we must change the active node, from time to time, to give a chance to other nodes within the same zone to become active. However, the transition between sleep and active states consumes energy, especially at the end of the nodes' lifetime where the election frequency increases. To avoid this energy waste, we used this new strategy that allows the leader to remain active until its battery exhaustion and the other nodes enter the energy conservation mode by turning off the radio and going to sleep mode for a period T_s . After this period (i.e., T_s period), the nodes can start the election process if the leader's energy level is less or equal to the threshold `Min_Tunofftime`. In this case, the priority is given to another node with high energy level to become the new leader of this region. So, a node in active state never returns to sleep state, which minimizes the number of transitions between states. Thus, the value of activity period T_a is defined as the expected node lifetime.

Another way to minimize the number of transitions, between sleep and discovery states, is the choice of the value of the sleep period. It should be noted that the value of T_s is defined from T_a . So, the sleeping period should not allow nodes to wake up quite early but just before the active node's battery exhaustion. We propose that nodes, in the Sleeping state, remain in this state, for a T_s period of time which must be less than T_a . For this, we define a T_g period called *grace period* that is infinitely small, compared to T_a . Thus, the nodes awake T_g seconds earlier than expected (i.e. $T_s = T_a - T_g$). This will enable them to elect a new active node, regardless of the node in service. Before the active node's battery exhaustion, we are sure that another active node is elected to replace it. Consequently, this strategy leads to a connectivity gain.

Figure 4.4 shows the application of this strategy for 3 equivalent nodes. Initially, each node sets a timer T_d and broadcasts a discovery message. In this example, *Node3* remains active because it has the highest priority. *Node1* and *Node2* enters a sleep state for a T_s period. The value of T_s must be less than that of the *Node3* activity period, in order to give nodes enough time to elect a new leader before the loss of the old one. After T_s timers of *Node1* or/and *Node2* expire, they wake up and send a

discovery message to *Node3*. The latter, responds with a discovery message, containing its remaining energy level. In this example, we suppose that *Node3* remaining energy is less than a threshold *Min_Turnofftime*. In this case, at the reception of this message by *Node1* or/and *Node2*, they trigger an election process. We note that *Node3* does not participate in the election process. If we suppose that *Node2* is elected as a new leader; then *Node1* can go into sleep mode. The number of transitions from sleep to active mode is equal to the number of equivalent nodes. We also note that, for a certain period of time, we can find two active nodes in the zone which leads to a connectivity gain.

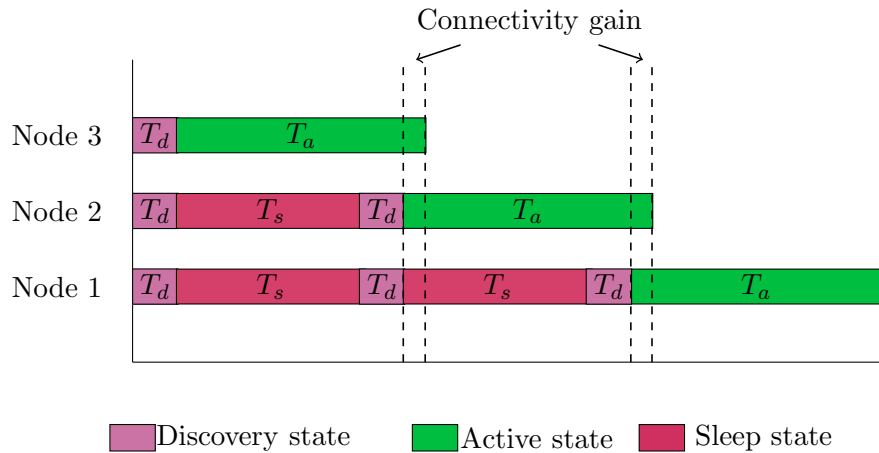


Figure 4.4: The scheduling policy used by GTC with connectivity gain.

4.3.5 GTC analysis

GTC aims to conserve energy as well as preserve the network connectivity which is the fundamental requirement of topology control. The topology control problem is divided into two major steps, identification of equivalent nodes and scheduling nodes for Sleep and Active modes, based on this equivalence. Like other topology control protocols, GTC runs into two phases: Zones formation phase and Leaders election phase.

In the first phase, the network is divided into zones so that nodes, within the same zone, are equivalent for routing and no matter which node is selected from each zone to be a leader. GTC identifies the redundant nodes with a low communication overhead. The communication complexity of GTC, in the first phase, is $O(n)$, where n is the number of nodes in the network. Indeed, each node broadcasts a discovery message to its neighbors once it has determined its zone.

In the second phase, GTC selects only one node in each zone to be a leader; while the others enter the sleep mode. A leader node remains active until its battery depletion. This can reduce the number of transitions between sleep and active mode. This scheduling policy reduces (1) the transition energy and (2) the frequency of elections,

which reduces the number of messages exchanged. In what follows, we give the communication complexity of the load balancing scheduling and the scheduling used by GTC.

Let n , the number of nodes, deployed in an area A of size XY and R the communication radio range for each node. The maximum size of a zone can be set to $R/\sqrt{5}$ (According to (4.1)). The minimum number of zones, noted m , would be $m = \frac{A}{(R/\sqrt{5})^2}$. Consequently, the average number of nodes, located in each zone will be $\frac{n}{m}$.

The load balancing strategy gives nodes the possibility to work periodically and ensure routing data for several periods. So, the number of the activity periods of a node, noted p , must be greater than 1 (i.e., $p > 1$). The number of activity period of all the nodes, in one zone, is $O(p\frac{n}{m})$. For each period, only one node must be elected to be active. The number of exchanged messages, in one zone, is of $O(\frac{n}{m})$. As the minimum number of zones in the network is m , the complexity becomes $O(p\frac{n^2}{m})$.

By cons, the scheduling until the node's battery depletion allows each node to work for one period. So, the number of activity periods of all the nodes, in one zone, is $O(\frac{n}{m})$. The number of times, that the election process is triggered, is equal to $\frac{n}{m} - 1$. The election of the first node requires an exchange of messages of $O(\frac{n}{m})$, the second one $O(\frac{n}{m} - 1)$, and so on, except for the last which is active without election. In addition, the election process is triggered when nodes receive a message from the leader and its remaining energy reaches the Threshold. So, the number of messages, sent by leaders, in each zone, is $O(\frac{n}{m})$. Consequently, the number of exchanged messages in one zone is $O((\frac{n}{m})^2 - \frac{(\frac{n}{m}-1)(\frac{n}{m}-2)}{2})$. As the minimum number of zones in the network is m , the complexity becomes $O(m((\frac{n}{m})^2 - \frac{(\frac{n}{m}-1)(\frac{n}{m}-2)}{2}))$.

4.4 Performance evaluation

To validate and evaluate our solution, we use simulation to study the effects of GTC's scheduling approach on the network lifetime, energy saving and data delivery, using the ns-2 simulator. Thus, to prove the effectiveness of our scheduling policy and to show the energy gain, we compare it to a load balancing policy. We prove it by using a scheduling that minimizes the number of transitions between sleep and active states. For this reason, we compare our protocol to GAF, a grid-based protocol, using load balancing strategy. GAF is considered as a reference protocol and it is the first topology control protocol based sleep-scheduling and most protocols, that have been proposed subsequently use the same technique.

Our simulation environment is characterized by an area of 100m x 100m. Sensors are uniformly deployed and supposed to be static (no mobility). All the nodes in the network have the same radio transmission range which is equal to 25m. We used the Two-Ray-Ground as the radio propagation model and IEEE 802.11 as MAC protocol. The energy model used is that proposed in [46]. We make a 600-node simulation with

100 Joule as initial energy for each node, in order to compare the the protocols GAF and GTC, in terms of network lifetime, data delivery and scalability property. Table 4.2 summarizes simulation parameters, used in the simulation.

Table 4.2: Simulation parameters.

| | |
|--------------------------|----------------|
| Topography | 100m*100 m |
| Number of nodes | 600 nodes |
| Initial Energy | 100 joule |
| Propagation model | Two-Ray-Ground |
| MAC protocol | 802.11 |
| Radio transmission range | 25m |
| Transmit power | 1.9W |
| Receive power | 1.5 W |
| Idle power | 0.75 W |
| Sleep power | 0.025W |

In addition, we show the relevance to use a topology control protocol, complementary to a routing protocol. We know that, if the nodes are all the time active, this consumes uselessly their energy, especially when only a reduced number of nodes participate to routing packets. To conserve energy, the topology control protocol must be used to select just a minimum number of nodes that can ensure routing packets. For this reason and to show the improvement, in terms of energy conservation, we compare GTC and GAF to the routing protocol AODV.

In our simulation, we evaluate the performances of our algorithm, using the following metrics:

1. *Network lifetime*: this is a rate that expresses the evolution of the number of nodes with positive energy as a function of time. It is given by the following equation:

$$Lifetime_Rate_{(t)} = \frac{N_nodes_{(E>0)}}{N_nodes_{(t=0)}} \quad (4.3)$$

where $Lifetime_Rate(t)$ is the percentage of the living nodes at t, $N_nodes_{(E>0)}$ is the number of nodes having a positive energy, and $N_nodes_{(t=0)}$ is the initial number of nodes.

2. *Data delivery rate*: the delivery rate is a ratio that expresses the relationship between the number of emitted packets and the number of received packets. This relationship is given by the following equation:

$$Deliverance_Rate = \frac{Received_PQ_{[t_i, t_j]}}{Emitted_PQ_{[t_i, t_j]}} \quad (4.4)$$

where $Received_PQ_{[t_i, t_j]}$ and $Emitted_PQ_{[t_i, t_j]}$ are, respectively, the number of received packets and the number of emitted packets in the interval of time $[t_i, t_j]$.

3. *Scalability "NTR"*: the scalability property is studied in order to show how the density (i.e., the number of nodes) affects the network lifetime and to study the network behavior when the number of nodes increases. For this metric, we used the same idea as [113], where this property is represented by the number of nodes in the transmission range (NTR), given by the following formula:

$$NTR = \frac{\pi R^2 n}{A} \quad (4.5)$$

where R is the transmission range; n , the number of nodes and A is the topography dimension (simulation area).

4.4.1 Network lifetime

The graph, in the Figure 4.5, shows the network overall lifetime, by monitoring the fraction of nodes having a positive energy. The first thing we observe is that with AODV, all nodes run out of energy at the same time, at about 125s. This is due to the fact that AODV does nothing to conserve energy and all nodes take part in the routing. On the other hand, we show that GAF improves the network lifetime at least 4 times more than the routing protocol (AODV), as the nodes die at about 525s. However, we can clearly observe that GTC has a convergent curve and all nodes die at the time 725s. Moreover, with GTC the network lifetime approaches a value 5 times higher than AODV's but with a fraction of nodes much smaller than GAF's. Secondly, we observe that GAF tries to keep a maximum of nodes in life, as long as possible. Then, starting from a certain value, all the nodes become exhausted of only one blow. This is due to the load balancing strategy, adopted by GAF. Hence, all nodes remain up and running together, as long as possible. From a certain value, all nodes will have exhausted their batteries equitably. In contrast, we find that the curve on GTC as a 6-step staircase because we made a simulation with an average of 6 nodes per zone. This is due to the fact that GTC adopts a scheduling policy with priority until the battery exhaustion. This makes the nodes serve group by group.

Finally, we can say that topology control protocols have extended the network lifetime of 70% for GAF, and 95% for GTC, compared to AODV. In addition, with our GTC protocol, 30% of nodes are still alive after 525s, compared to GAF, where all nodes are dead at this time.

4.4.2 Data delivery rate

The data delivery, rate as function of the network lifetime, is given by the graph in the Figure 4.6. We remark that GAF roughly quadruples AODV lifetime and GTC

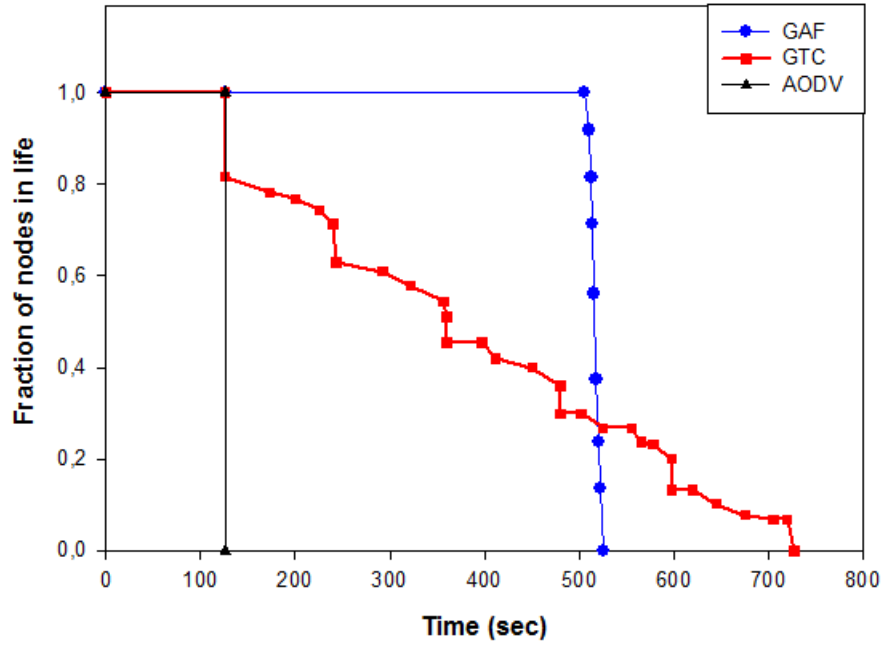


Figure 4.5: Network lifetime.

Table 4.3: The node densities and equivalent NTR.

| Node Density | 100 | 200 | 400 | 600 | 900 |
|----------------|-------|-------|-------|-------|--------|
| Equivalent NTR | 19.63 | 39.26 | 78.53 | 117.8 | 176.71 |

increases 5 times AODV lifetime. We also notice that the curve which represents the GAF delivery rate is decreasing until it reaches the rate of 30%. From this value, the curve makes a positive peak up to the value 50% then the rate goes down again at a null rate. In GAF, this is due to the leader's elections which are expensive, both in terms of Discovery-messages sending and connectivity loss. So, during the election process, nodes exchange discovery messages that create collisions and consequently the possibility of losing packets. Moreover, the more the leader election frequency increases, the more node energy decreases and the more the rates of connectivity loss increase. However, in our protocol, the delivery rate remained stable between 90% and 95% because : (1) the leader's election process is executed only the number of nodes (6 times in each zone for our scenario), (2) no connectivity loss and (3) fewer messages are exchanged.

4.4.3 Scalability

In order to check this property, we varied the density of the nodes, and calculated the NTR (Table 4.3).

The simulation results, for this parameter, are given by the graph in Figure 4.7. This graph represents the variation of the network lifetime as a function of the node densities.

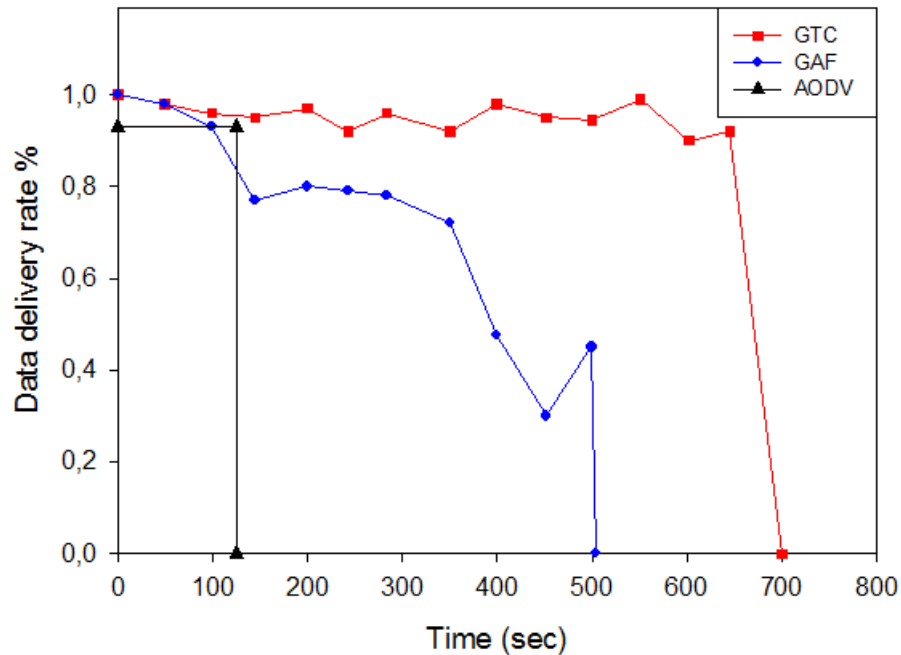


Figure 4.6: Data delivery rate in the network.

We observe that the more the NTR increases, the more the network lifetime increases for both GAF and GTC protocols, while AODV's lifetime is kept flat. However, the line, which represents the GTC lifetime, regarding density, has a slope higher than that which represents the GAF lifetime. For this reason, when the nodes density reaches 900 nodes, the network lifetime, using GTC, reaches its higher bound, 7.5 times the AODV network lifetime. While, the network lifetime, using GAF as a topology control protocol, reaches only 5.5 times of AODV's lifetime. So, we find that our GTC protocol extends the network lifetime by 20% compared to GAF. This improvement is due to the scheduling policy, used by our protocol, that minimizes the number of transitions. We note that, the more the node density increases, the more the number of transitions increases. Figure 4.8 shows the number of transitions for both protocols GAF and GTC. From the figure, we find that the number of transitions in GAF is much higher, especially when the density reaches 900 nodes. With 900 nodes, the number of transitions in GAF is 4 times greater than the number of transitions in GTC.

4.5 Conclusion

In this chapter, we investigated the issue of energy conservation, in WSNs, by keeping a minimum number of sensor nodes to operate in active mode, as well as ensure network connectivity. To address this problem, we proposed a grid-based topology control protocol named GTC that can lengthen the network lifetime. GTC divides the network into small regions and only one active node is selected in each region. It uses a

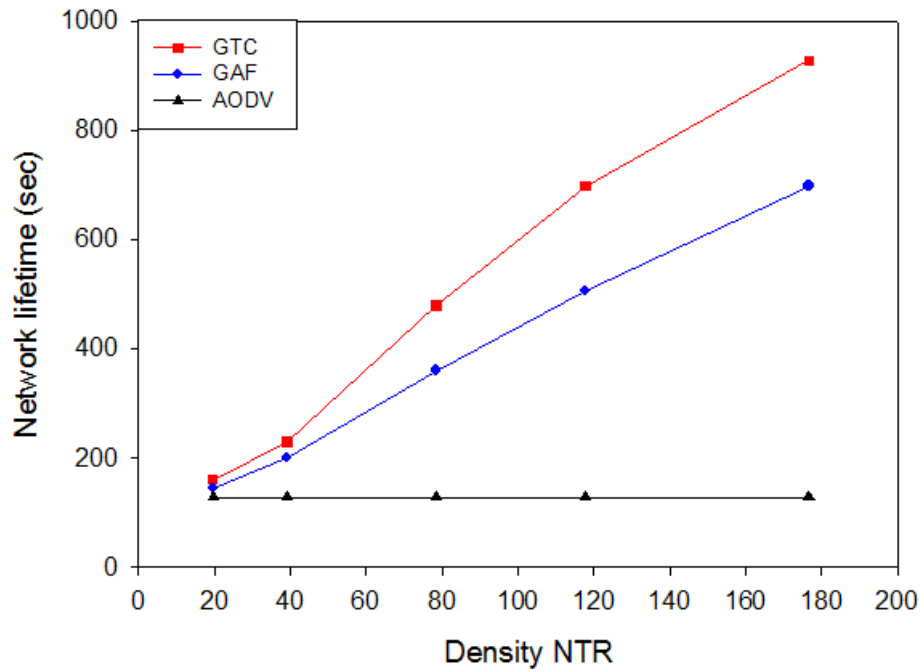


Figure 4.7: The node density influence on the network lifetime.

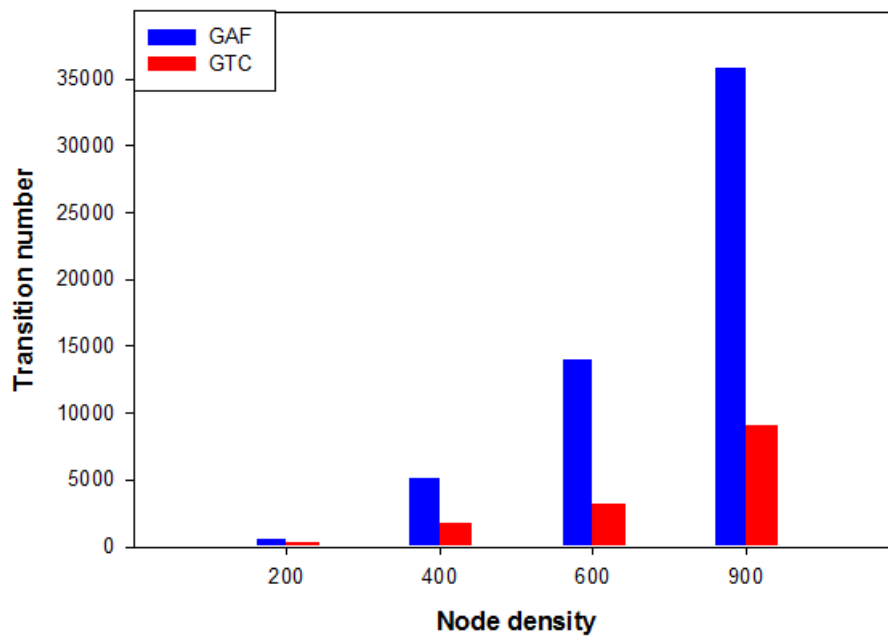


Figure 4.8: Transition number Vs Node density.

scheduling strategy with priority that aims to reduce the transition energy. Besides, the proposed protocol is characterized by: (1) a reduced number of messages, (2) a low frequency of leaders' election, (3) a minimum number of state transitions, and (4) a connectivity gain by introducing the grace period. Simulation results show that the proposed protocol displays better performance in terms of network lifetime, data de-

livery, scalability, and message overhead, compared to GAF's approach. Our protocol extends the network lifetime by 20%, compared to GAF, because we use a new scheduling which minimizes the communication overhead and the election frequency that leads to an energy conservation. Note that our protocol can provide a long network lifetime, if we used an appropriate energy model for sensor nodes, whose consumed energy is measured in mWatt and not in Watt. However, the proposed protocol assumes that the communication range is a perfect disc which is far from reality. Moreover, as the most grid-based protocols, GTC requires the availability of location information provided by GPS, which is sometimes impossible to acquire. For this reasons, we should define a new mechanism to group nodes dynamically without using any localization system.

In the next chapter, a new group-based topology control protocol is proposed, which uses only neighborhood information to identify redundant nodes.

Chapter 5

Group-based Energy-Conserving Protocol (GECP)

5.1 Introduction

In this chapter, we focus on how to identify redundant nodes with a low communication overhead without using any localization system. The group-based topology control category can be investigated since it uses only neighborhood information to group redundant nodes together. In this direction, a Group-based Energy-Conserving Protocol (GECP) [124, 127] is proposed and presented in this chapter. GECP exploits the sensor redundancy in the same region by dividing the network into groups so that a connected backbone can be maintained by keeping only one active node in each group and turning off the redundant ones. It identifies redundant nodes, organizes them into groups with a low communication overhead. It uses the same scheduling strategy as GTC which aims to minimize the number of transitions between active and sleep states and, hence, extend the network lifetime.

The rest of this chapter is organized to define, first, the network model and some important concepts in Section 5.2. The detailed description of the proposed algorithm is presented in Section 5.3. Section 5.4 gives the complexity analysis of GECP and another group-based topology control protocol CPA [28]. In Section 5.5, we evaluate the performance of our approach using an appropriate energy model for sensor nodes. Finally, a conclusion is given in section 5.6.

5.2 Network model

In this chapter, we consider a homogeneous wireless sensor network, consisting of one (or more) sink(s) and a high number of sensor nodes, deployed over a large geographic area. Both the sink and the sensor nodes are assumed to be static; and all nodes have the same radio communication range. We assume that each node is assigned a unique identifier, *NodeID*. In addition, we don't take into consideration the nodes' crash

except the depletion of their batteries. The deployed sensors should have the ability to self-organize in order to monitor the environment and detect an event when it occurs. Based on energy efficiency, we propose a distributed topology control protocol for WSN which can work conjointly with any routing protocol. Before describing the proposed protocol, we define some important concepts.

Definition 5.1. (*Neighbor List*). Node v is a neighbor of node u , if a beacon message had been received from v . The neighbor list of node u , noted $NList(u)$, is the set of sensor nodes that are neighbors to u .

Definition 5.2. (*Node Degree*). The degree of node u , noted $NDegree(u)$, is the number of neighbors, i.e., $NDegree(u) = |NList(u)|$.

Definition 5.3. (*Connectivity*). Node u is said to be connected to node v , if there exists a path between u and v , i.e., u is neighbor to v and vice versa.

Definition 5.4. (*Same Degree Neighbor List*). The same degree neighbor list of node u , noted $SDNList(u)$, is the set of nodes that are neighbors to u having the same node degree as u .

Definition 5.5. (*Shared Neighbor List*). The shared neighbor list of two nodes u and v , noted $SNList(u,v)$, is the set of nodes that are neighbors to both u and v , i.e., $SNList(u,v) = NList(u) \cap NList(v)$.

Definition 5.6. (*Node Equivalence*). Node v is said to be equivalent (or redundant) to node u , in term of communication, if v can reach (or cover) a subset of u 's neighbors, i.e., $SNList(u,v) \neq \emptyset$.

Definition 5.7. (*Equivalent Neighbor List*). The equivalent neighbor list of node u , noted $EquivNList(u)$, is the set of equivalent neighbors providing required communication redundancy with u . Formally, $v \in EquivNList(u)$, if the following formula is satisfied:

$$\frac{|SNList(u,v)|}{|NList(u) - \{v\}|} * 100 = 100\% \quad (5.1)$$

where $v \in SDNList(u)$, and 100% is the connectivity level.

Definition 5.8. (*Completely adjacent groups*). Two groups G_i and G_j are fully adjacent if all nodes in G_i can communicate with all nodes in G_j and vice versa. Formally,

$$\forall u \in G_i, \forall v \in G_j / u \in NList(v) \text{ and } v \in NList(u)$$

Definition 5.9. (*Arbitrary adjacent groups*). Two groups G_i and G_j are arbitrary adjacent if at least one node in G_i can not communicate with all nodes in G_j and vice versa. Formally,

$$\exists u \in G_i, \exists v \in G_j / u \notin NList(v) \text{ and } v \notin NList(u)$$

We will use the term *Equivalent Neighbor List* and *group* interchangeably in the chapter.

5.3 GECP design

The wireless sensor network is characterized by its density in number of deployed sensors. In reality, sensors in the same region, generally, collect and forward the same information. So, to save energy, it should be necessary to schedule the sensor activity so that it allows redundant sensors to enter the sleep mode while the others are kept active. To design such a sensor-scheduling scheme, we should answer the following questions: (1) How to determine redundant nodes? (2) Which rule should each redundant sensor follow to determine whether to enter active mode or sleep mode? (3) How long should a sensor remain in the active mode?

With the purpose of conserving energy (and, hence prolonging the network lifetime) as well as ensuring the network connectivity, we propose a group-based distributed scheme called GECP, which is used to allow each sensor to switch between active and sleep modes to save energy. Indeed, the proposed scheme divides the network into groups so that nodes within the same group are equivalent for routing so that no matter which node is selected from each stock group, all the active nodes can form a connected backbone network. Thereafter, only one node is active in each group (called leader), while others go to sleep state. Therefore, GECP works in three phases: (1) Neighbor Discovery Phase (2) Group Forming Phase and (3) Leaders Election Phase.

5.3.1 Discovery phase

Initially, each node will discover its neighbors by exploiting beacon messages exchanged between sensor nodes that contain the transmitting node identifier *NodeID*. In this way, each node builds a neighborhood list, noted *NList*.

5.3.2 Group forming phase

In this phase, the energy dissipated during the group forming must be negligible in order to not to affect the leaders election phase and network lifetime. This can be achieved by minimizing the number of exchanged messages, i.e., communication overhead. The use of the *Neighbor List* is the key concept in our solution. That is the reason why we use it as the basis in defining and measuring communication redundancy and equivalence.

After the discovery phase, the process of group forming begins and takes place as follows:

- Each node broadcasts a discovery message, containing its *NodeID* and its *NList*. This process enables each node to acquire the 2-hop neighbors' information which would be enough to perform subsequent steps.

- Each node, receiving this message, locally computes its equivalent neighbor list *EquivNList*:
- First, each node selects the neighbors having the same degree as it to get the same degree neighbor list *SDNList*. The interest of building the set *SDNList* is to reduce the computation cost because there are great chances for the neighbors with the same degree to have the same neighbors in a dense network.
- Thereafter, each node computes the shared neighbor list *SNList* with each of its neighbors and determines the *EquivNList* or its group members according to equation (5.1). In this way, the groups are independently formed so that nodes with the same set of neighbors will be in the same group. All group members have the same knowledge of the group's members without exchanging information. Groups formed, after this phase, are completely adjacent and this ensures the network connectivity, regardless of the selected node to represent the group and ensure data routing.

The process of group forming is summarized in Algorithm 1.

Algorithm 1 : Group forming

```

1: EquivNList( $u$ ) :=  $\{u\}$ ;
2: //Find neighbors with the same degree;
3: SDNList( $u$ ) :=  $\{\}$ ;
4: NDegree( $u$ ) :=  $|NList(u)|$ ;
5: for all  $v \in NList(u)$  do
6:   NDegree( $v$ ) :=  $|NList(v)|$ ;
7:   if NDegree( $u$ ) == NDegree( $v$ ) then
8:     SDNList( $u$ ) := SDNList( $u$ )  $\cup$   $\{v\}$ ;
9:   end if;
10: end for;
11: //Find equivalent neighbors;
12: if SDNList( $u$ )  $\neq \emptyset$  then
13:   for all  $v \in SDNList(u)$  do
14:     SNList( $u, v$ ) :=  $NList(u) \cap NList(v)$ ;
15:     if  $|SNList(u, v)| == |NList(u) - \{v\}|$  then
16:       EquivNList( $u$ ) := EquivNList( $u$ )  $\cup$   $\{v\}$ ;
17:     end if;
18:   end for;
19: end if;

```

To further illustrate the steps presented above, we present an example scenario before moving into the node scheduling phase (leaders election phase). Figure 5.1(a) shows a part of topology consisting of 11 nodes connected via wireless links, from which we can find the node neighbor lists, the node degrees, the set of nodes having the same degree and the list of equivalent neighbors as shown in Table 5.1. Figure 5.1(b) shows

the resulting topology after the group forming phase. Taking the example of node 2, its neighboring table consists of 7 nodes. Nodes having the same degree as it, are 5, 6, 10. After running the group forming algorithm, nodes 2 and 5 are equivalent and form a group. Both nodes have the same information without exchanging it, hence the advantage of our solution. Figure 5.1(b) shows the resulting topology after the group forming phase.

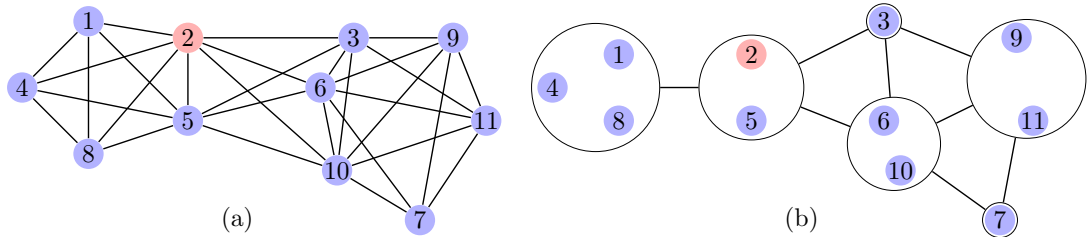


Figure 5.1: (a) The initial network topology represented by a communication graph. (b) The network topology after the group forming phase.

Table 5.1: 2's neighboring table and its EquivalentNeighborList.

| Node | NeighborList | NodeDegree |
|-----------------|-------------------------|------------|
| 2 | {1, 3, 4, 5, 6, 8, 10} | 7 |
| 1 | {2, 4, 5, 8} | 4 |
| 3 | {2, 5, 6, 9, 10, 11} | 6 |
| 4 | {1, 2, 5, 8} | 4 |
| 5 | {1, 2, 3, 4, 6, 8, 10} | 7 |
| 6 | {2, 3, 5, 7, 9, 10, 11} | 7 |
| 8 | {1, 2, 4, 5} | 4 |
| 10 | {2, 3, 5, 6, 7, 9, 11} | 7 |
| $SDNList(2)$ | {5, 6, 10} | |
| $SNList(2,5)$ | {1, 3, 4, 6, 8, 10} | |
| $SNList(2,6)$ | {3, 5, 10} | |
| $SNList(2,10)$ | {3, 5, 6} | |
| $EquivNList(2)$ | {5} | |

5.3.3 Leaders election phase

The goal of topology control is to exploit the nodes redundancy to conserve their energy while maintaining network connectivity. In fact, it determines the equivalent nodes and allows those that are redundant to go into sleep mode to conserve their energy. For this, a sleep-wakeup cycle or sleep scheduling is defined between nodes. Once the group forming phase is completed, each group starts the sleep/wake scheduling stage. In order to save energy, only one node in each group is required to be active while others turn off their radio (go to sleep mode). GECP uses the same scheduling strategy with priority as GTC which is based on nodes energy level. The priority is given to nodes with high

energy level to remain active. If there are nodes with the same energy, we use the node identity as selection criterion. In this case, the priority is given to the node having the highest identity.

Initially, each node broadcasts an election message. Each group member, receiving this message, compares its energy level with the other nodes in its group. If its energy is the highest, it becomes the leader of this group because it has the highest priority. A node, in active state, continues forwarding data and routing packets until it runs out of energy while the other nodes enter the energy conservation mode by turning off the radio and going to sleep mode for a period T_s . After this period (i.e., T_s period), the nodes can start the election process if the leader's energy level is less or equal to the Threshold. In this case, the priority is given to another node with high energy level to become the new leader of this group. So, a node in active state never returns to sleep state, which minimizes the number of transitions between states.

GECP state transitions

In GECP, nodes are in one of three states: *passive*, *active* and *sleep*. The state transitions diagram is given in Figure 5.2.

Passive state: Initially, all nodes start in *passive* state. In this state, each node sets a timer T_e and broadcasts *Election* message, containing the *NodeID*, *Energy Level* and *Node State*, to its group members. If a node receives an *Election* message from an active node, or from a node with a high energy level, before the expiration of T_e , it cancels the T_e timer and enters the sleeping state for T_s period. Otherwise, when T_e expires, a node enters the active state for T_a period and broadcasts an *Election* message, in order to inform the neighbors that it is the elected active node for T_a period.

Active state: In active state, a node remains active and forwards data until the depletion of its battery.

Sleeping state: A node, in a sleep state, wakes up after the expiration of T_s timer and moves into a *passive* state, in order to elect a new active node. There are two cases:

- The leader's remaining energy level is greater than the Threshold. In this case, the node returns to sleep mode for another T_s period.
- The leader's remaining energy level is less than the Threshold. In this case, the node does not return to sleep mode but sets a timer T_e and can be the new leader if it has a higher energy level than the others. In other words, a node can be the new active one if its energy level, is greater than that of the other nodes.

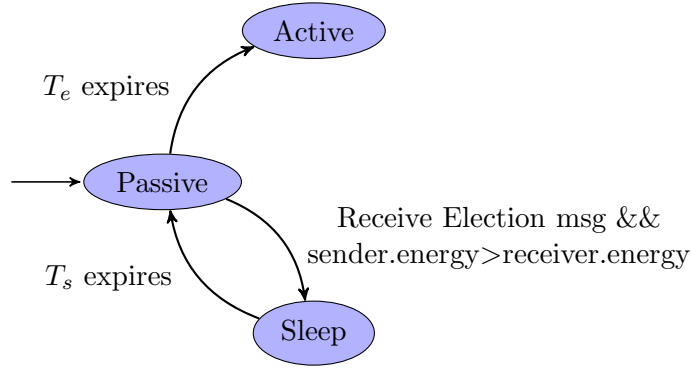


Figure 5.2: GECP state transitions.

5.3.4 GECP parameters tuning

In this section, we explain the parameter choices made in the current GECP algorithm but it should be noted that the choice of the parameters T_e , T_g , and Threshold is left to the application.

Node active time (T_a) represents the duration of activity of the leader. Initially, it sets to the expected node lifetime. The expected node lifetime is calculated according to the following assumption: if a node functions while using a maximum of resources, how much this latter will hold before the exhaustion of its battery? In reality, the real value of T_a depends on the network traffic.

Election message interval (T_e) determines the maximum time a node remains in the passive state. This parameter used to avoid node synchronization. It is chosen as a random value between two constants. We leave the choice of these constants to the application.

Threshold is the minimum time that allows the active node to go on ensuring the packet routing during the election of a new leader. This can also avoid the connectivity loss.

Node sleeping time (T_s) is equal to the duration of activity of the leader minus a certain grace period T_g . We note that if nodes wake up after T_s and the leader can ensures its activity (i.e., T_a is greater than the Threshold), nodes adjust their T_s to the remaining duration of activity of the leader minus grace period.

Grace period (T_g) represents the sufficient amount of time to complete the process of election of a new leader. This parameter ensures that nodes wake up before the remaining energy of the leader reaches the Threshold, in order to avoid the connectivity loss.

5.4 Complexity analysis

We analyze the complexity of our protocol and CPA in terms of time and communication. The analysis concerns only the group forming phase because it significantly affects the effectiveness of the protocol. Assume that n sensors are deployed into an area of interest, and the average connectivity degree of nodes is m .

5.4.1 Time complexity

In the centralized version, CPA encode the partition by a symmetric $n \times n$ matrix which gives the connectivity level of each node. The complexity of calculating this matrix is $O(n^2)$. In the process of calculating merging utility value, the maximum number of neighbors for each group is m . So, the complexity of the union and intersection operations for each two groups is $O(2m^2)$. Then, the level of equivalence between these two groups is measured and the complexity is $O(2m)$. Further, to measure the merging priority for all two completely adjacent groups is $O(nm)$. It is also required to calculate the ratio of energy in two groups to the total energy of the entire network. The complexity to calculate the total energy is $O(n)$. Therefore, the complexity to determine all candidate groups is $O(nm(2m^2 + 2m)) + O(n)$. In the process of sorting merging priority for all candidate groups, the number of groups is uncertain in each iteration and the maximum number is n , the complexity can be $O(n)$. Thus, the complexity of the merging phase in each iteration, is $O(nm(2m^2 + 2m)) + O(2n)$. Because the maximum number of iterations does not exceed n , the complexity becomes $O(n^2m(2m^2 + 2m)) + O(2n^2)$. Finally, the whole complexity of the centralized version of CPA is $O(n^2m(2m^2 + 2m)) + O(2n^2) + O(n^2)$.

In the distributed version of CPA, the complexity is calculated for each node. First, each node calculates the merging utility value which consists to determine the union, the intersection, the total energy. So, the complexity is $O(2m^2 + n)$. Thereafter, the level of equivalence between two groups is measured and the complexity is $O(2m)$. To measure the merging priority for all two completely adjacent groups is $O(m)$, and the complexity becomes $O(m(2m^2 + n + 2m))$. Further, because the number of groups is uncertain in each iteration and it does not exceed m , the complexity is $O(m)$. In order to determine all candidate groups, the process of group merging is repeated several iterations. The maximum number of iterations is m . Thus, the whole complexity becomes $O(m(m(2m^2 + n + 2m)) + O(m)) \simeq O(m^4)$.

In GECP, we only calculate the time complexity for each node because is a distributed scheme. First, the *SDNList* is calculated, and the complexity is $O(m)$ because the maximum number of neighbors for each node is m . Let's assume that k ($0 \leq k < m$), the number of nodes having the same degree. In the case of the *SDNList* is not empty, the node computes the *SNList* with each of its neighbors, in the *SDNList*, and deter-

mines the *EquivNList* with the complexity of $O(k^3)$. Thus, the whole complexity of the group forming phase of GECP is $O(k^3)$.

5.4.2 Communication complexity

For CPA, we only calculate the communication overhead of the distributed version. After sensors deployment, each node records information from its neighbors, and the complexity is $O(1)$. In order to calculate the merging utility value, the energy total is collected, and the communication overhead is $O(2n)$. When the candidate groups are selected, the head node send a MERGE_REQ message to the selected group. If the contention succeeds, it receives a MERGE_ACK message (otherwise, it receives a MERGE_NAK message). The head nodes of the two groups broadcast a HOLD_MSG to the neighboring groups. After the merge is finished, they broadcast an UPDATE_MSG. We note that, a neighbor node receiving an HOLD_MSG or an UPDATE_MSG, broadcasts its new state to the neighboring nodes. So, There are five type of messages exchanged during the merging process of each two groups, and the complexity is $O(4m + 7)$. The process of group merging is repeated several iterations. The maximum number of iterations is m , and the complexity is $O(m(2n + 4m + 7))$. Therefore, the communication complexity, for each node, is $O(2nm + 4m^2 + 7m + 1)$. Knowing that the number of nodes is n , the communication overhead of the distributed version of CPA becomes $O(2n^2m + 4nm^2 + 7nm + n)$.

Unlike CPA, in GECP, each node broadcasts one message to its neighbors. The communication overhead of GECP is $O(n)$ where n is the number of nodes in the networks.

5.5 Performance evaluation

In order to evaluate the performance of our protocol, we implemented it in order to perform simulations using Ns2 simulator. Our protocol is simulated using random topologies with different densities. When we examine node density, we vary the number of nodes from 20, 50, 100 and 200 while keeping a 20mx20m constant area. We define node density, noted D , as the number of nodes per radio range, given by the following equation [113]:

$$D = \frac{\pi * R^2 * n}{x * y} \quad (5.2)$$

where R is the radio range; n the number of nodes; and x and y , the simulation area width and length. Table 5.2 summarizes simulation parameters.

The performance evaluation of our solution consists of two parts: Group forming phase and Sleep/wake scheduling phase.

Table 5.2: Simulation parameters.

| | |
|--------------------------|------------------------|
| Topography | 20m*20 m |
| Number of nodes | 20, 50, 100, 200 nodes |
| Initial Energy | 5 joule |
| Propagation model | Two-Ray-Ground |
| Radio transmission range | 5m |
| Transmit power | 660mW |
| Receive power | 395mW |
| Idle power | 395mW |
| Sleep power | 100mW |

5.5.1 Group forming phase

In this phase, GECP is evaluated, in terms of number of groups formed and energy consumed, during this phase. To compare the performance of our algorithm with that of CPA, we implemented the distributed version of CPA and simulated it in the same environment as our protocol.

Number of groups formed

We notice that during the simulation of CPA, using different topologies, nodes die before ending the group forming phase. This is due to the high number of messages exchanged for the collection of total energy of all nodes and messages exchanged to merge nodes. Thus, the energy consumed during the grouping phase in CPA is not at all negligible as the authors of the CPA mention it and justify it by the fact that this phase runs once after network deployment.

For this reason and in order to compare the performance of our grouping nodes mechanism and that of CPA, in terms of number of groups, the messages exchanged for the total energy collection were neglected and only merging messages were considered to complete the nodes merging process in CPA. Table 5.3 summarizes the results.

From the results, we note that the number of groups formed is important for both the protocols because of nodes random deployment. We also observe that the number of groups formed increases with the increasing of nodes density. Moreover, for CPA, the number of groups increases with the connectivity degree *mindeg*. This is due to the need for a large number of active nodes, at a given time, to ensure high connectivity. On the other hand, we note that the number of groups formed in CPA is lower than our protocol for densities D3 and D4. This can be justified by (1) the strong constraint used by GECP to group similar nodes and requires that nodes must have exactly the same neighbors to form a group, (2) both nodes density and random deployment

Table 5.3: Number of groups formed in GECP and CPA under different node densities.

| Node Density | D1=3.92 | D2=9.81 | D3=19.63 | D4=39.67 |
|----------------|---------|---------|----------|----------|
| GECP | 13 | 39 | 89 | 187 |
| CPA (mindeg=1) | 13 | 26 | 55 | 170 |
| CPA (mindeg=2) | 16 | 33 | 62 | 171 |
| CPA (mindeg=3) | 20 | 39 | 63 | 171 |
| CPA (mindeg=4) | 20 | 45 | 76 | 171 |

affect the number of groups formed and their size. However, with CPA, nodes in the same group are not necessarily equivalent, which leads to network partitioning during the scheduling phase (see Section 5.5.2). We can explain this case with the following example: Consider the topology in Figure 5.3(a), each node calculates the utility value for each neighbor given by the equation (5.3):

$$f_{ij} = k_1 \left(1 - \frac{|N_2(A_i) \cap N_2(A_j)|}{|N_2(A_i) \cup N_2(A_j)|} \right) + k_2 \left(\frac{g(A_i) + g(A_j)}{E_{total}} \right) \quad (5.3)$$

where $N_2(A_i)$ and $N_2(A_j)$ are the set of neighboring groups completely adjacent with groups A_i and A_j , respectively, $g(A_i)$ and $g(A_j)$ denote the energy in group A_i and A_j , respectively, E_{total} is the total energy of all nodes in the network, and k_1 and k_2 are two coefficients. If we take the connectivity degree $\text{mindeg} = 1$, we find that nodes a and c are equivalent, even nodes f and k . If nodes f and k merge first, the resulting topology is that given in Figure 5.3(b). What can be observed in this topology is that both node b and group $\{f, k\}$ are arbitrarily adjacent. In the scheduling phase, the choice of an active node in each group affects connectivity. If the node f is selected as active node, connectivity is provided; but if the node k is chosen to represent its group, a partitioning will appear in the network.

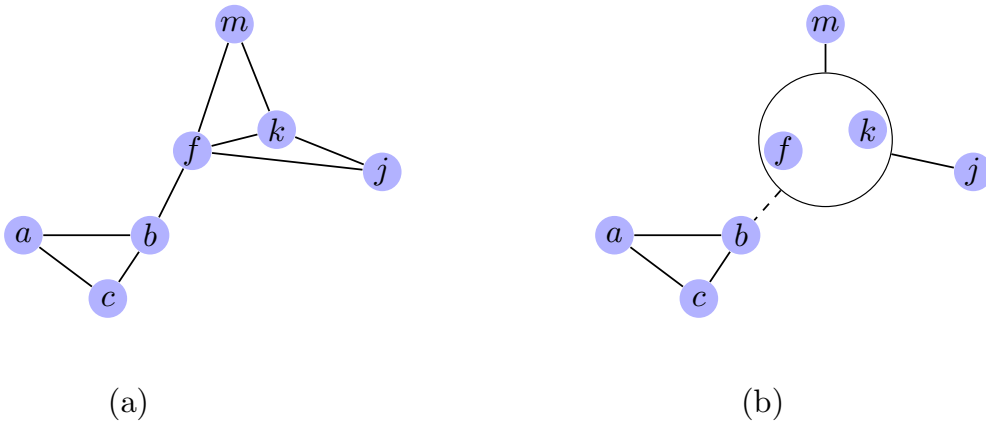


Figure 5.3: CPA group merging process. (a) Initial partition. (b) First iteration.

Energy consumed

Energy is an important factor for topology control protocol because it determines its effectiveness. A minimum energy consumption is desirable during the group forming so as not to influence the scheduling phase. Thus, the energy consumed during nodes grouping phase must be negligible. Figure 5.4 shows the energy consumed, during the group forming phase, by both GECP and CPA. It should be noted that the energy rate consumed by CPA and shown in this figure corresponds only to the energy consumed, after the exchange of merging messages between groups. In other words, we assume that the energy consumed in the total energy collection, which is an important parameter in the calculation of CPA utility values, is zero. The results show that energy consumption increases by increasing the number of nodes for the both protocols. However, for GECP, energy consumption is really negligible. Therefore, we can say that with our protocol, in the worst case, even though the number of groups is equal to the number of nodes, the energy consumed in the group forming phase does not affect the network lifetime. Unlike what precedes, when running CPA, the obtained results show that the increase of energy consumption rate is significant. This is what happens, especially, with densities D3 and D4 where all nodes lose their energy during the merging process. So, GECP is scalable with the network size while CPA is not.

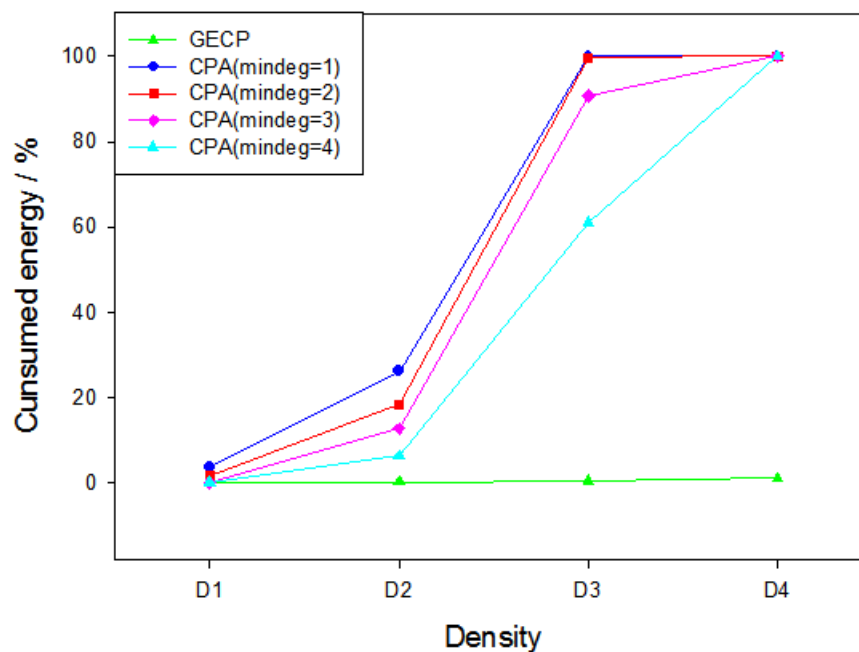


Figure 5.4: Consumed energy VS node density.

5.5.2 Sleep/wake scheduling phase

To see the effect of scheduling on the network lifetime, we first simulated the network without using any scheduling mechanism. If no sleeping scheduling scheme is used, all nodes keep active until death. We used the flooding as the routing protocol. In fact, nodes periodically (every second) send a message to the sink node.

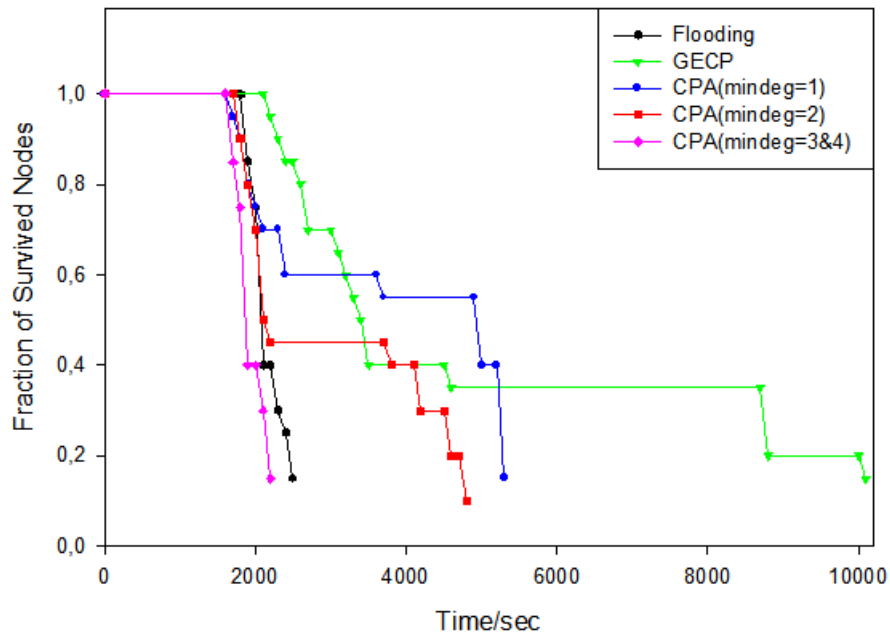
Network lifetime

The graphs in Figures 5.5, 5.6, 5.7 and 5.8 show the network lifetime, by monitoring the fraction of nodes having a positive energy and the fraction of remaining energy for the densities D1, D2, D3 and D4 respectively. The first thing we observe is that with Flooding, 80% of nodes run out of energy at about 2400s for D1, 1700s for D2, 1100s for D3 and 600s for D4. This is due to the fact that Flooding does nothing to conserve energy and all nodes take part in the routing. By cons, we find that the use of a topology control protocol, based on sleep-scheduling, with a routing protocol enhances significantly the sensor network lifetime. For instance, we can observe from Figure 5.5 that when 80% of nodes run out of energy, the network lifetime is extended by 4 times for GECP, 2.2 times for CPA(mindeg=2) and $1.95 \approx 2$ times for CPA(mindeg=1).

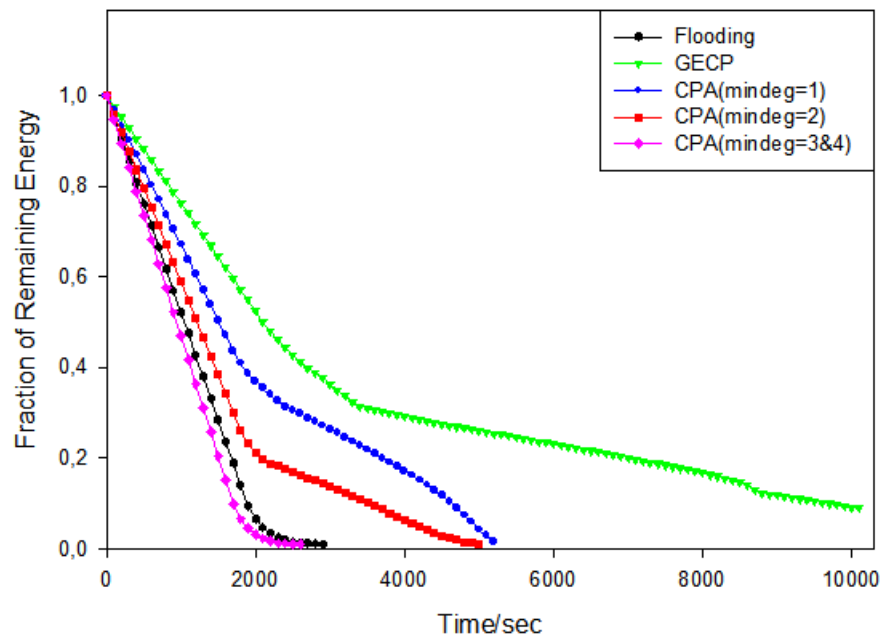
Furthermore, we can clearly observe that the results of GECP are better than those of CPA in terms of the number of survived nodes at each time, despite the high number of active nodes produced by running GECP. This improvement is due to the scheduling policy, used by our protocol, that minimizes the number of transitions.

Transition frequency

Figure 5.9 shows the number of transitions for both protocols GECP and CPA. We can see that the number of transitions in CPA is much higher, compared with GECP, especially for the density D2. This is due to important number of sleeping nodes and the scheduling policy used by CPA which tries to ensure a load balancing among nodes in same group. So, it changes the active node, periodically, to give a chance to other nodes within the same group to become active which increases the transition frequency. Figure 5.10 shows the number of sleeping nodes for densities D1 and D2 respectively. From the figures, we can see the frequent switching between active and sleep modes when the CPA approach is applied while GECP gives some stability in the network.

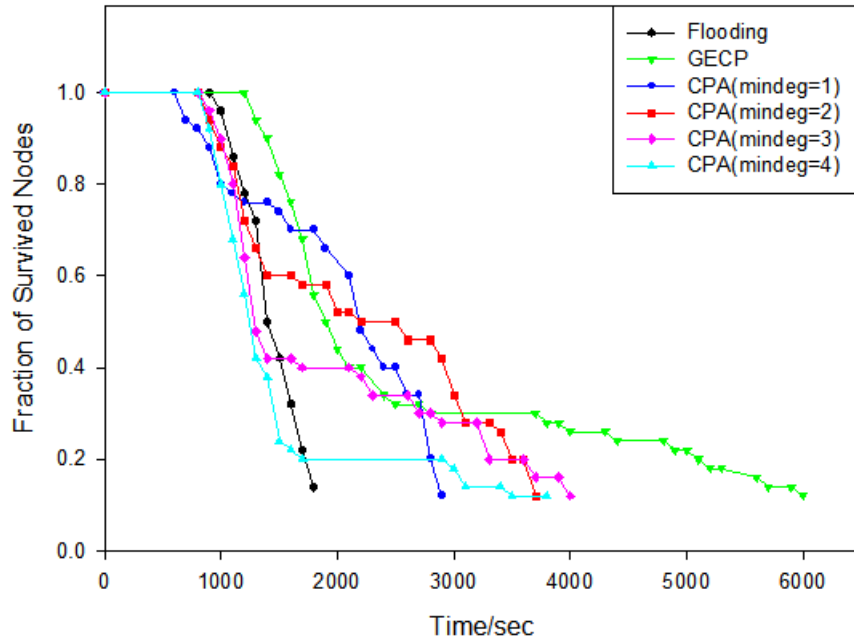


(a) Fraction of survived nodes VS Time

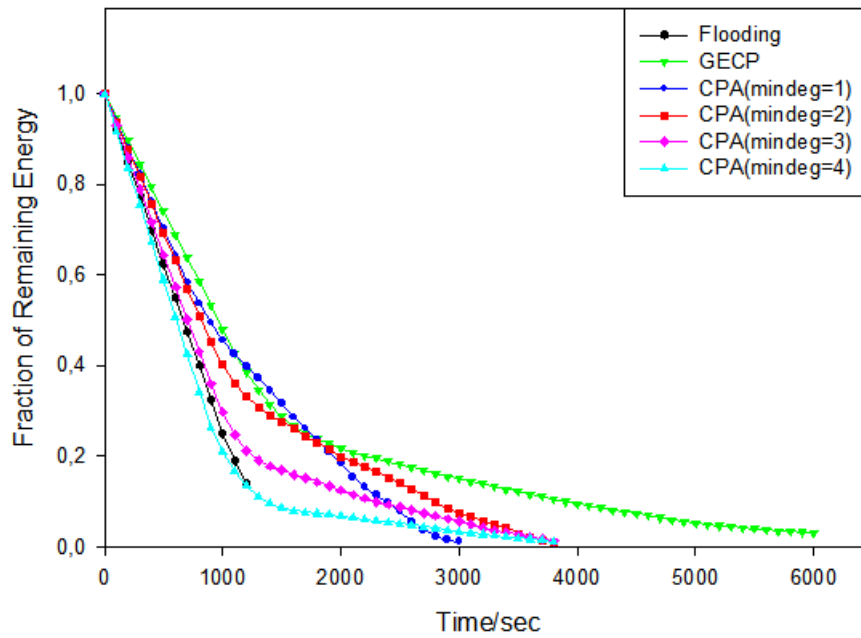


(b) Fraction of remaining energy VS Time

Figure 5.5: Network Lifetime for Density D1.

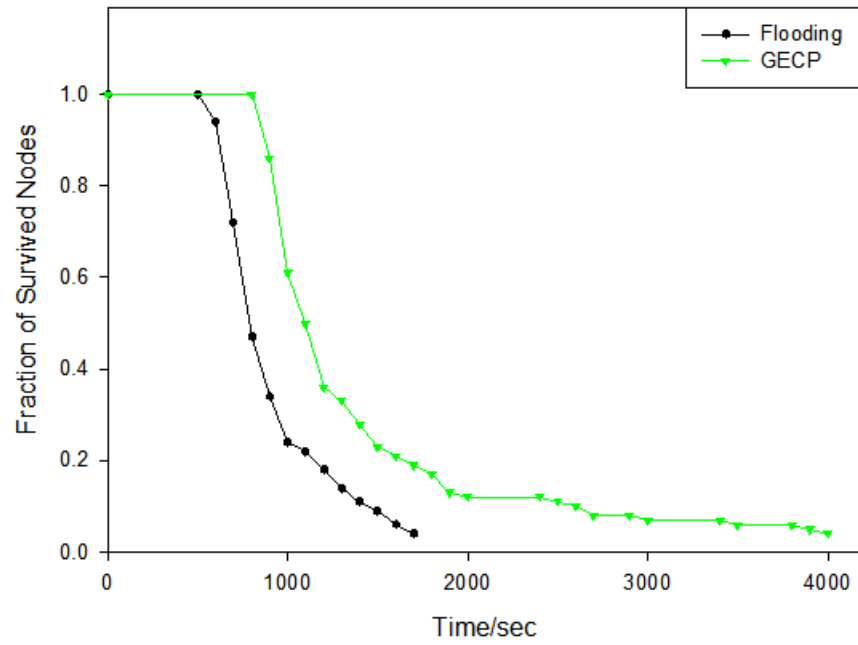


(a) Fraction of survived nodes VS Time

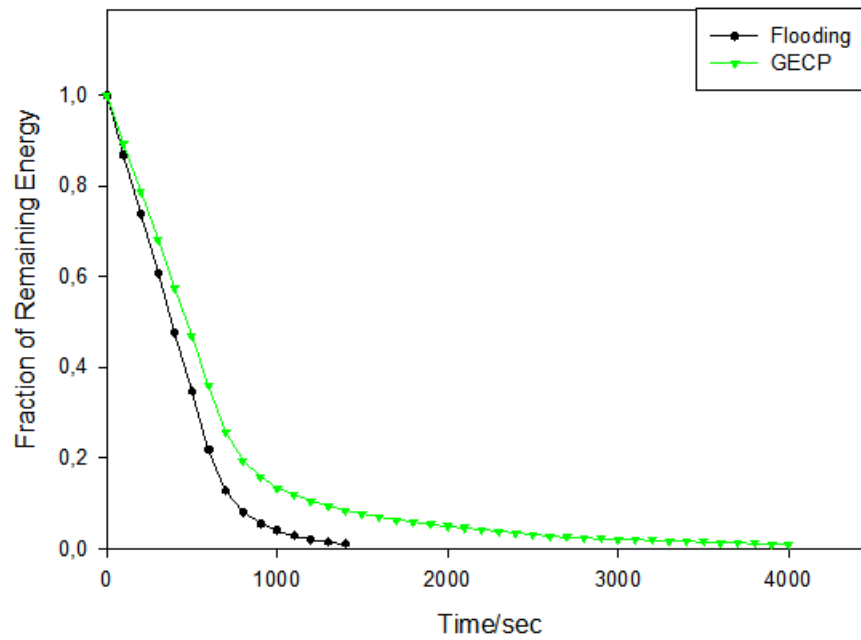


(b) Fraction of remaining energy VS Time

Figure 5.6: Network Lifetime for Density D2.

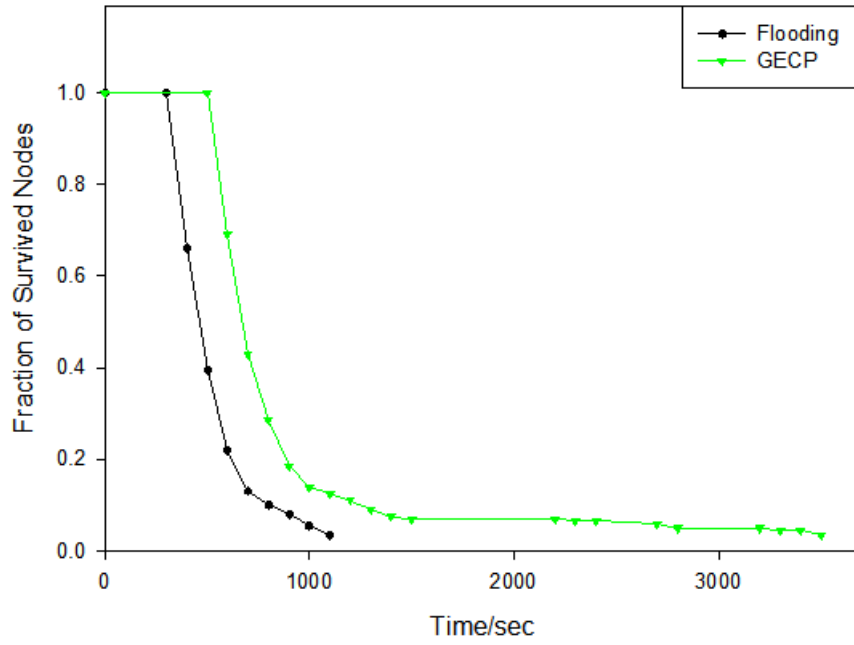


(a) Fraction of survived nodes VS Time

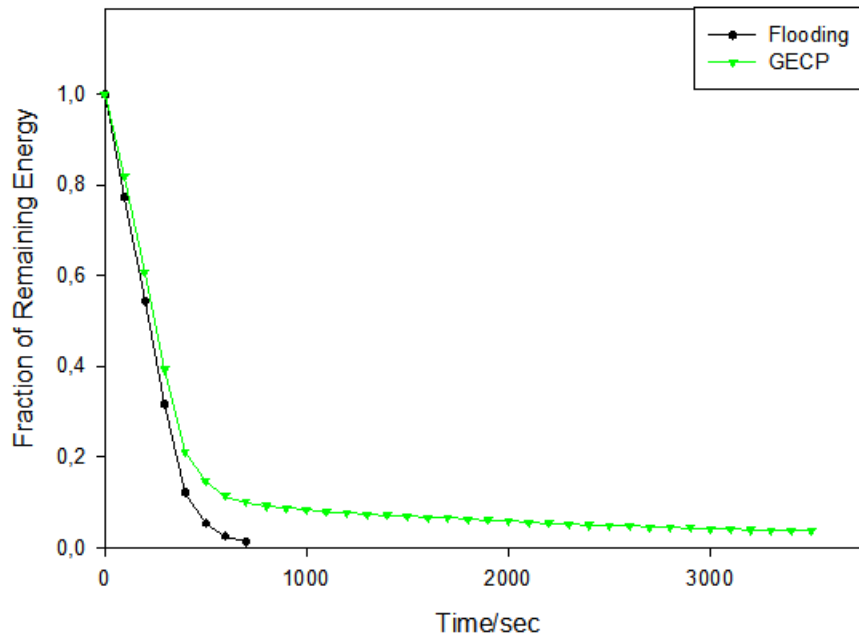


(b) Fraction of remaining energy VS Time

Figure 5.7: Network Lifetime for Density D3.



(a) Fraction of survived nodes VS Time



(b) Fraction of remaining energy VS Time

Figure 5.8: Network Lifetime for Density D4.

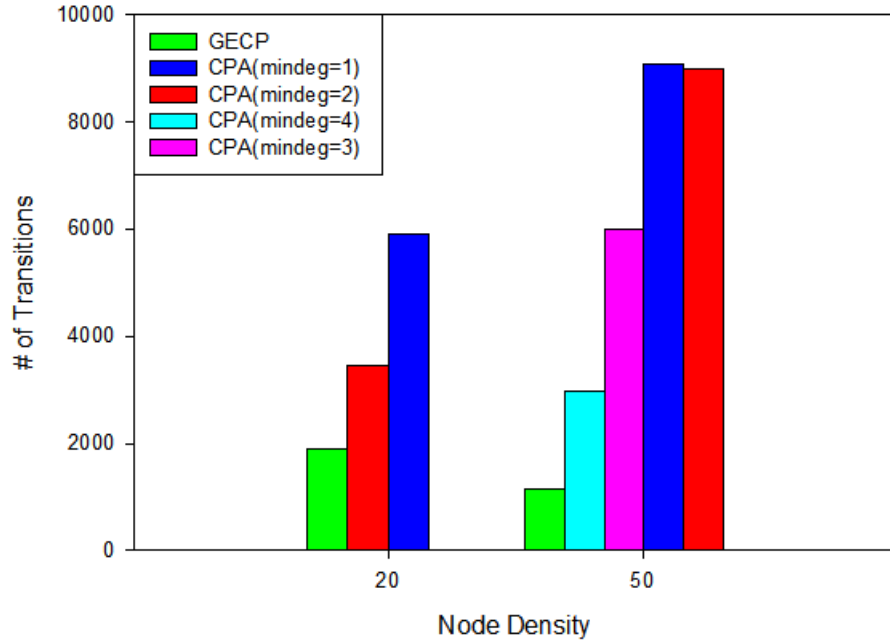
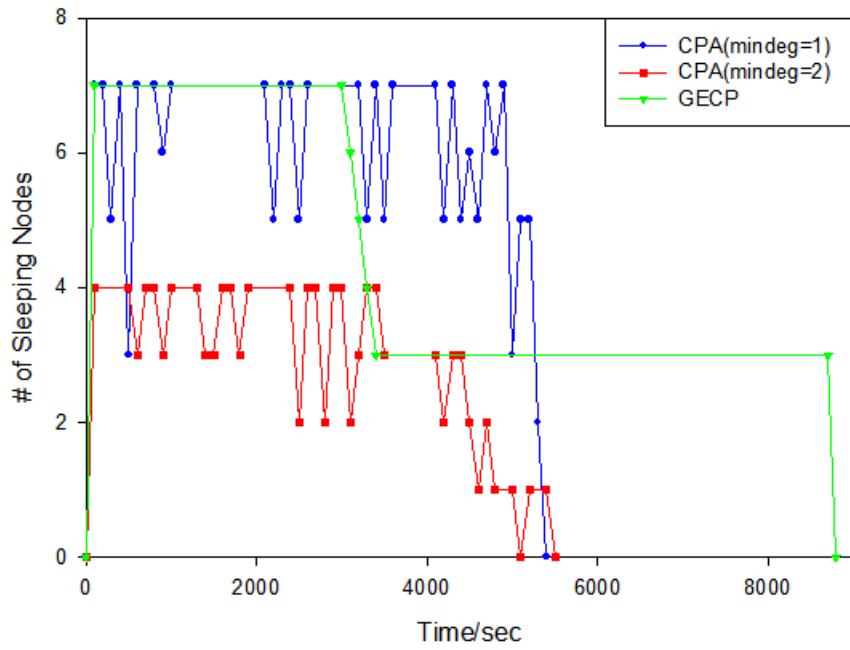


Figure 5.9: Number of transitions VS Node density.

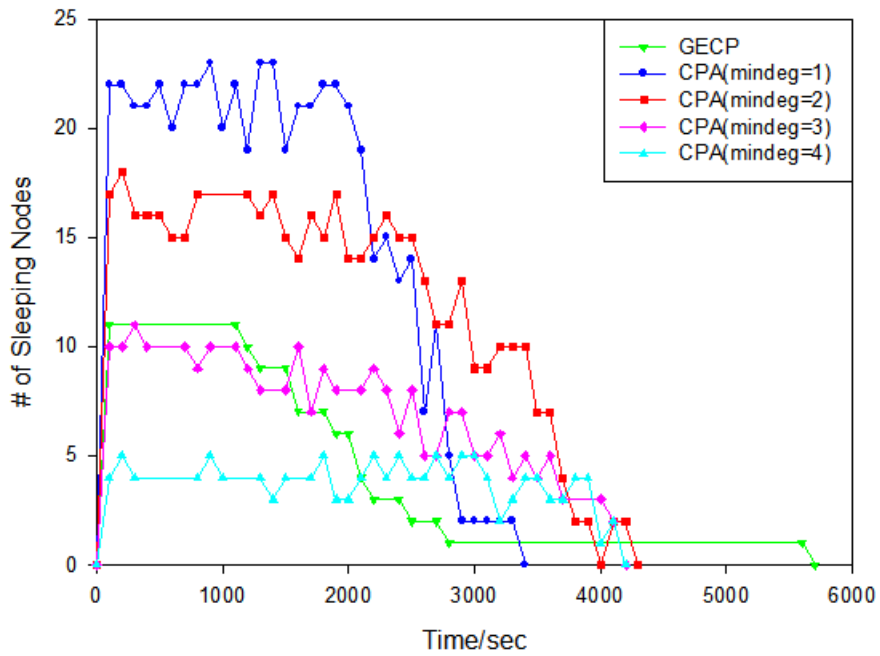
Network connectivity

In this section, we examine the topology graphs that result from using GECP and CPA in the case of D2 density. Figure 5.11(a) shows a topology graph when there is no topology control protocol is employed. Figure 5.11(b) shows the corresponding graph produced by GECP. We can see that there are more nodes selected to be active than necessary which strengthens the connectivity degree. Figures 5.11(c), 5.11(d), 5.11(e) and 5.11(f) show the graphs obtained by CPA(mindeg=1), CPA(mindeg=2), CPA(mindeg=3) and CPA(mindeg=4) respectively. From the graphs, we can see that CPA(mindeg=1) and CPA(mindeg=2) select fewer active nodes, compared with GECP, leading to more nodes that sleep and save energy. However, a loss of connectivity can occur with CPA(mindeg=1), leading to network partitioning because nodes in the same group are not equivalent as it is mentioned in Section 5.5.1.

Finally, from all these results, we can say that the effectiveness of a topology control protocol in terms of network lifetime is influenced by two parameters: (1) the technique used to nodes grouping, (2) the scheduling technique used for the active nodes election. In the nodes grouping phase, the more energy consumed is negligible, the longer lifetime is extended. In addition, the lower the number of active nodes, the more lifetime is maximized. On the other hand, in the scheduling phase, the election frequency of active nodes affects the energy consumed.



(a) Density D1



(b) Density D2

Figure 5.10: Number of Sleeping nodes VS time.

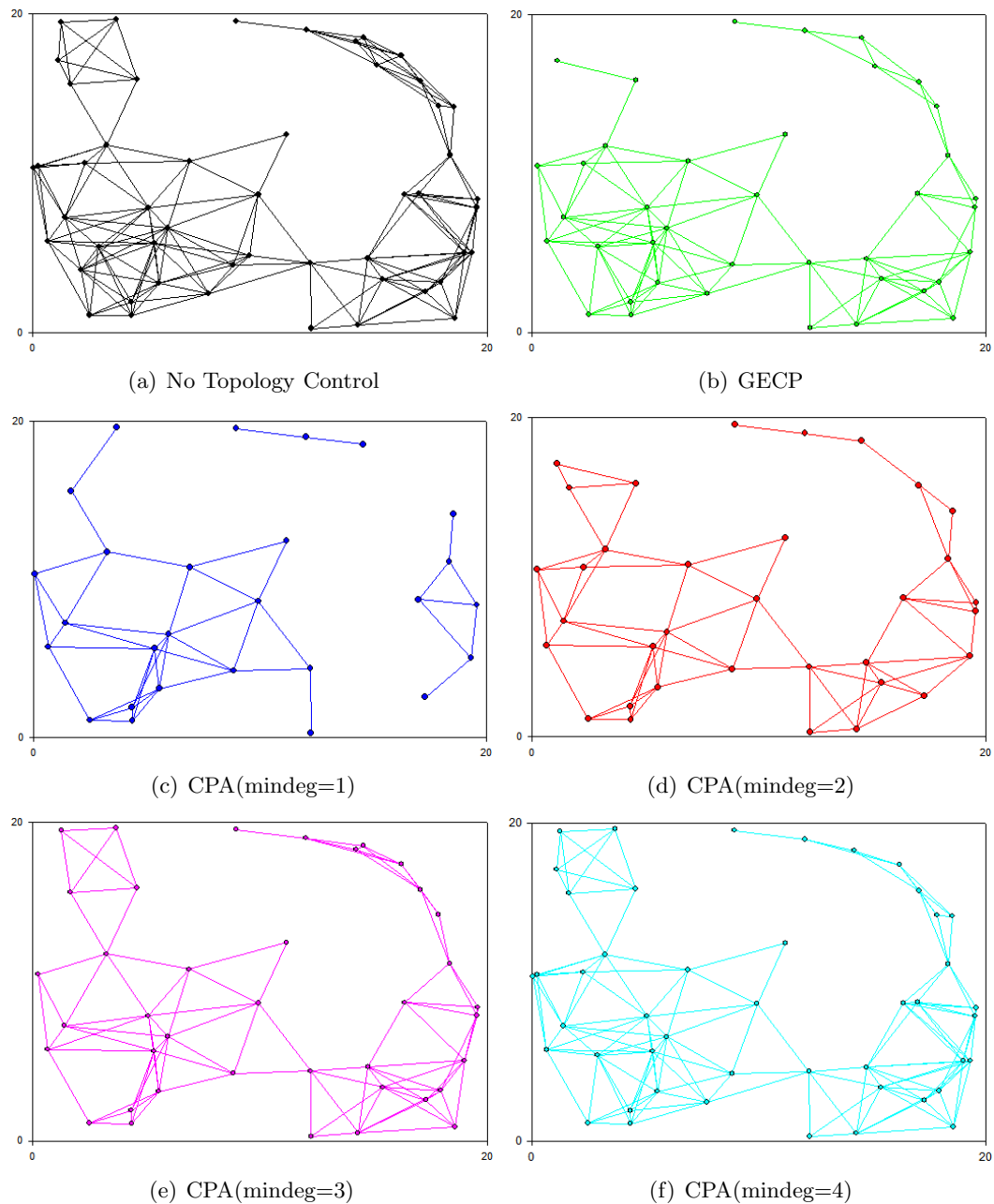


Figure 5.11: The network graphs of no Topology Control, GECP and CPA algorithms at 65s.

5.6 Conclusion

In this chapter, we proposed a protocol that guarantees energy saving while providing network connectivity. Our protocol called GECP exploits the sensor redundancy in the same region by dividing the network into groups so that a connected backbone can be maintained by keeping a necessary set of working nodes and turning off the redundant ones. GECP has three advantages: connectivity guarantee, energy saving, and algorithm efficiency. The simulation results show that : (1) the use of topology

control protocol, based on sleep-scheduling, with a routing protocol extends the network lifetime, (2) our solution outperforms existing group-based solutions in terms of energy saving, network lifetime and connectivity guarantee. Finally, we can say that the effectiveness of the protocol in terms of network lifetime is influenced by the technique used to nodes grouping and the scheduling technique used for the active nodes election.

Despite the scheduling policy used by GECP provides some stability in the network by minimizing the frequency of transitions between active and sleep modes, it does not provide a load balancing between nodes in terms of energy. In addition, this scheduling technique has the advantage of being the most conservative of energy but fails to be fault tolerant. On the other hand, in GECP, the criterion used to group equivalent nodes together is too strong which leads to high number of active nodes. So, it is necessary to reduce the number of groups (hence, the number of active nodes) using another constraint to identify node redundancy without increasing the communication overhead.

Chapter 6

RTCP: a Redundancy aware Topology Control Protocol

6.1 Introduction

In Chapter 5, we discussed the design of a group-based topology control protocol GECP that identify redundant nodes with a low communication overhead but it uses a strong constraint to group equivalent nodes together which leads to high number of active nodes. In addition, as we have stated in Chapter 3, several topology control algorithms, using the scheduling technique, have been proposed for Ad Hoc and sensor networks. They determine how many and which nodes should be allowed to sleep, with the purpose to ensure connectivity. However, the number of the active nodes is not always the minimum and sometimes the formed topology is not connected. Furthermore, a large communication overhead will be produced during the search of the redundant nodes which leads to large energy consumption. Consequently, it is desirable to design a new protocol that reduces the number of active nodes while ensuring connectivity on the one hand and without wasting the energy on the other hand. Nevertheless, there is a trade-off between power consumption and connectivity. When we try to ensure connectivity, nodes should exchange messages, which increase the communication overhead and, therefore, the energy consumption.

Motivated by this, a new distributed Redundancy aware Topology Control Protocol (RTCP) [121, 123, 126] is proposed in this chapter. RTCP shares a common purpose among the sleep scheduling-based topology control protocols which is the energy conserving while ensuring connectivity, by selecting a subset of active nodes. It uses the neighbor set as the communication redundancy metric, this allows RTCP to avoid any assumptions about communication regularity. In addition, it identifies redundant nodes and organizes them into groups with a low communication overhead. The redundant nodes that satisfy some eligibility rules, can form group according to their redundancy degrees, which leads to have a reduced number of groups without lose the network connectivity. RTCP defines a Threshold of connectivity level which allows applications to

parameter the desired connectivity degree in the reduced topology. On the other hand, RTCP tries to reduce the number of active nodes in the scheduling phase by selecting only a single node in each group.

In this chapter, we define, first, the communication redundancy metric and the eligibility rules used by RTCP to select active nodes while maintaining the network connectivity. Then, Section 6.3 gives a detailed description of the different phases of the proposed protocol. In order to evaluate RTCP in terms of number of active nodes, network lifetime and connectivity, a set of simulations are conducted and the results are presented in Section 6.4.

6.2 Network model

The Network model we use is similar to that described in Chapter 5. We consider a homogeneous wireless sensor network, consisting of one (or more) sink(s) and a high number of sensor nodes, deployed over a large geographic area. Both the sink and the sensor nodes are assumed to be static; and all nodes have the same radio communication range. We assume that each node is assigned a unique identifier, *NodeID*. Before describing the proposed protocol, we define additional important concepts.

Definition 6.1. (*Union Neighbor List*). The union neighbor list of two nodes u and v , noted $UNList(u,v)$, is the set of nodes that are neighbors to u or v , i.e., $UNList(u,v) = NList(u) - \{v\} \cup NList(v) - \{u\}$.

Definition 6.2. (*Non-Common Neighbor List*). The non-common neighbor list of two nodes u and v , noted $Non-CNList(u,v)$, is the set of nodes that are not neighbors to u or v , i.e., $Non-CNList(u,v) = UNList(u,v) - SNList(u,v)$.

Definition 6.3. (*Redundancy Degree*). The redundancy degree of two nodes u and v , noted $RedDegree(u,v)$, describes quantitatively to what degree, neighbors of node v can be reached by node u . Formally, $RedDegree(u,v) = \frac{|SNList(u,v)|}{|UNList(u,v)|}$.

Definition 6.4. (*Threshold of Connectivity Level*). The Threshold of connectivity level, noted T_{cl} , is the neighbors set that a node needs to cover before it can be eligible to Sleep. This value is related to the probability of having a connected network; the higher the value of T_{cl} , the harder to enter the Sleep mode, and the higher the probability that the network is connected.

Definition 6.5. (*Equivalent Neighbor List*). The equivalent neighbor list of node u , noted $EquivNList(u)$, is the set of equivalent neighbors providing required communication redundancy with u . Formally, $v \in EquivNList(u)$, if the following formula is satisfied:

$$RedDegree(u,v) \neq 0 \text{ and } RedDegree(u,v) * 100 \geq T_{cl}\% \quad (6.1)$$

Definition 6.6. (Group). The group of node u , noted $\text{Group}(u)$, is the set of nodes that are equivalent to u satisfying the following eligibility rules.

Eligibility Rule 1(ER1): An equivalent node v should be a neighbor to all u 's group members, i.e.,
 $\text{Group}(u) \subseteq \text{NList}(v)$.

Eligibility Rule 2(ER2): There exist at least two shared neighbors with different redundancy degrees between all nodes in $\text{Group}(u)$ and v . Formally,
 $\exists w, w' \in \text{NList}(v) \cap \text{NList}(\text{Group}(u)) / \text{RedDegree}(u,w) \neq \text{RedDegree}(u,w')$.

Eligibility Rule 3(ER3): All non-common neighbors between all nodes in $\text{Group}(u)$ and v are reachable by at least two shared neighbors. Formally,
 $\forall w \in \text{Non-CNList}(\text{Group}(u),v) / |\text{NList}(w) \cap \text{NList}(v) \cap \text{NList}(\text{Group}(u))| \geq 2$.

Definition 6.7. (Non-Common Member List). The non-common member list of two nodes u and v , noted $\text{Non-CMList}(u,v)$, is the set of nodes that are members in u 's group and not members in v 's group.

6.3 RTCP approach

In this section, a distributed and localized topology control protocol, Redundancy aware Topology Control Protocol (RTCP), is presented. Like other topology control protocols, RTCP works in three phases: Neighbor Discovery Phase, Group Forming Phase and Leaders Election Phase. In the first phase, each node will discover its neighbors. In the second one, the network is divided into groups so that nodes within the same group are equivalent for routing, no matter which node is selected from each group. In the third phase, only one node in each group is required to be active to ensure data forwarding, while other nodes turn off their radio.

6.3.1 Neighbor Discovery phase

Initially, each node will discover its neighbors, by exploiting beacon messages, exchanged between sensor nodes that contain the transmitting node identifier *NodeID*. In this way, each node builds its neighbor list *NList*.

6.3.2 Group Forming phase

In this phase, the energy dissipated during the group forming must be negligible in order not to affect the leaders election phase and network lifetime. This can be achieved by minimizing the number of exchanged messages, i.e., communication overhead. That is

the reason why we use, only, the *Neighbor List* as the basis in defining and measuring communication redundancy and equivalence, which is the key concept in our solution.

Once the neighbor lists *NList* are determined, the process of group forming can begin and takes place in two rounds.

First Round

In this round, each node determines the redundant neighbors and tries to form a group with them so that the connectivity is still ensured.

At the beginning, each node broadcasts a discovery message, containing its identifier *NodeID* and its neighbor list *NList*. As RTCP is a distributed scheme, this process enables each node to acquire the 2-hop neighbors' information which would be enough to perform subsequent steps.

Each node, receiving this message, locally computes the Redundancy Degree with each of its neighbors.

Thereafter, each node independently calculates its equivalent neighbor list *EquivNList*. Two nodes are equivalent and they can be in the same group, if their Redundancy Degree is greater than a certain Threshold of connectivity level T_{cl} (the value of T_{cl} will be discussed in Section 6.3.2). However, this condition is necessary to determine the equivalence between nodes but is not sufficient to group nodes together so that the network connectivity is still preserved.

Let's take the example of nodes 6 and 9, in the topology shown in Figure 6.1(a). Their neighbor lists, redundancy degrees, lists of equivalent neighbors are given in Table 6.1. If we assume that the T_{cl} value is 80%, nodes 3, 6 and 10 are equivalent and form a group. For their side, nodes 9 and 11 form a group too. The resulting topology is given in Figure 6.1(b), from which we note that the connectivity is still ensured. By cons, if we suppose that the T_{cl} value is 70%, node 9 finds that node 7 is equivalent and can be in the same group. Figure 6.1(c) shows the resulting topology, in this case, with a connectivity loss (considering the adjacency concept).

In order to ensure the network connectivity in the final topology, we define additional eligibility conditions. Two nodes u and v are equivalent if they satisfy the formula (6.1) and the three eligibility rules, ER1, ER2 and ER3. Formally,

$$Group(u) = \{v \in EquivNList(u) / v \in ER1 \cap ER2 \cap ER3\} \quad (6.2)$$

The first eligibility rule allows coordination between members of the same group, during the election of an active node. The others eligibility rules avoid a connectivity loss in the topology, formed by active nodes, selected in the leaders election phase.

Indeed, each node tries to find those which cover a set of its neighbors when it enters sleep mode so that this set of common neighbors is still reachable by a member of its

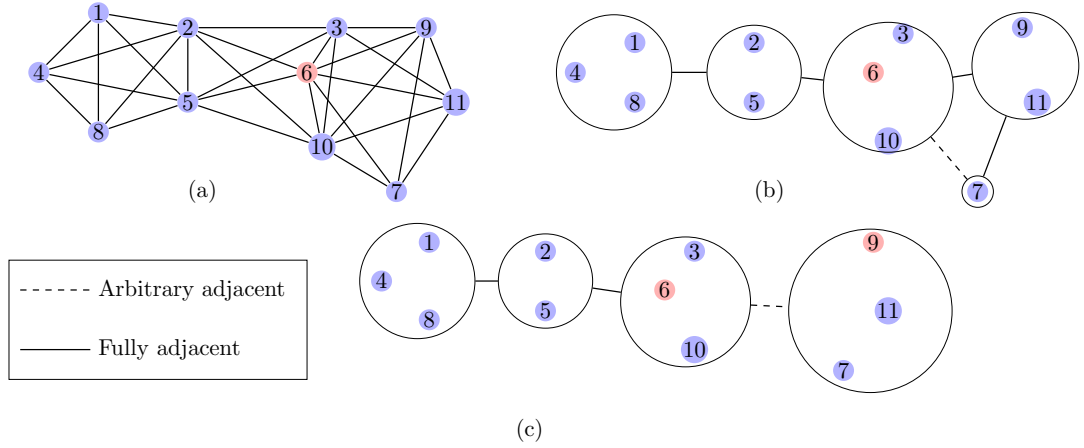


Figure 6.1: (a) The initial network topology (Topo1) represented by a communication graph. (b) The resulting topology when $T_{cl} = 80\%$ with connectivity guarantee. (c) The resulting topology when $T_{cl} = 70\%$ with connectivity loss.

Table 6.1: 6's and 9's neighboring tables and their equivalent neighbor lists for Topo1.

| Node6 | | |
|---|------------------------|-------------------------|
| Node | Neighbor List | Redundancy Degree |
| 2 | {1, 3, 4, 5, 6, 8, 10} | $\frac{3}{9} = 33.33\%$ |
| 3 | {2, 5, 6, 9, 10, 11} | $\frac{5}{6} = 83.33\%$ |
| 5 | {1, 2, 3, 4, 6, 8, 10} | $\frac{3}{9} = 33.33\%$ |
| 7 | {6, 9, 10, 11} | $\frac{3}{6} = 50\%$ |
| 9 | {3, 6, 7, 10, 11} | $\frac{4}{6} = 66.66\%$ |
| 10 | {2, 3, 5, 6, 7, 9, 11} | 1 = 100% |
| 11 | {3, 6, 7, 9, 10} | $\frac{4}{6} = 66.66\%$ |
| <i>EquivNList</i> (6) when $T_{cl} = 80\%$ or $T_{cl} = 70\%$ | | {3, 10} |

| Node9 | | |
|--|-------------------------|-------------------------|
| Node | Neighbor List | Redundancy Degree |
| 3 | {2, 5, 6, 9, 10, 11} | $\frac{3}{6} = 50\%$ |
| 6 | {2, 3, 5, 7, 9, 10, 11} | $\frac{4}{6} = 66.66\%$ |
| 7 | {6, 9, 10, 11} | $\frac{3}{4} = 75\%$ |
| 10 | {2, 3, 5, 6, 7, 9, 11} | $\frac{4}{6} = 66.66\%$ |
| 11 | {3, 6, 7, 9, 10} | 1 = 100% |
| <i>EquivNList</i> (9) when $T_{cl} = 80\%$ | | {11} |
| <i>EquivNList</i> (9) when $T_{cl} = 70\%$ | | {11, 7} |

group. However, the other neighbors which are not covered by some members (non-common neighbors) must be always reachable by some common neighbors. The third eligibility rule ensures that non-common neighbors are reachable by at least one active shared neighbor when the others go to sleep mode. Since our protocol is distributed and localized, each node can form a group with its neighbors (even the shared neighbors) that have no knowledge of other groups, formed in the neighborhood. If the shared neighbors have the same redundancy degree, the probability that these nodes form a

group is much higher and they can be seen as one node. When these nodes enter a sleep mode, a loss of connectivity can occur, hence the usefulness of the second eligibility rule.

Once the list of equivalent neighbors is established, each node should filter this list to determine its group members, in order to keep only equivalent neighbors, satisfying the eligibility rules. To select group members, each node sorts all equivalent neighbors in descending order, according to their redundancy degrees. Thereafter, it takes node by node in sorted order, it checks whether adding this node, as group member, does not violate the three eligibility rules. An equivalent node could be a group member if (1) it is neighbor to all group members, (2) there exist at least two shared neighbors, with different redundancy degrees between this node and all group members, and (3) all non-common neighbors between this node and all group members are reachable by at least two of their shared neighbors.

The first round of the group forming process is given by Algorithm 6.3.2.

To better illustrate the first round of the group forming process, let's back to the previous example in which we discuss the two cases: $T_{cl} = 80\%$ and $T_{cl} = 70\%$.

Case 1: $T_{cl} = 80\%$

First, the redundancy degrees and the equivalent lists are calculated by each node. The equivalent neighbor list of node 6 is formed by nodes 10 and 3, sorted according to their redundancy degrees. Node 6 adds node 10, first, to its group as a member because they have an equivalence of 100%. However, for node 3, it must verify the eligibility rules because they have a redundancy degree of 83.33%. It finds that (1) node 3 is neighbor to both 6 and 10, (2) the shared neighbor list, between node 3 and the group {6, 10}, which is {2, 5, 9, 11}, satisfies the second eligibility rule, and (3) all non-common neighbors between node 3 and the group {6, 10}, which are {7}, are reachable by the two common neighbors 9 and 11. So, node 3 can be added, to the group of node 6, as new member. By cons, the equivalent neighbor list of node 9 is formed by only node 11 which becomes a member in the group of node 9.

Algorithm 2 : Round 1 of Group Forming Phase

```

1:  $Group(u) := \{u\}$ ;
2: // Find equivalent neighbors
3:  $EquivNList(u) := \{\}$ ;
4: for all  $v \in NList(u)$  do
5:   Find shared neighbor list  $SNList(u, v)$ ;
6:   Find union neighbor list  $UNList(u, v)$ ;
7:    $RedDegree(u, v) := \frac{|SNList(u, v)|}{|UNList(u, v)|}$ ;

8:   if ( $RedDegree(u, v) \neq 0$ ) and ( $RedDegree(u, v) \geq T_{cl}$ ) then
9:      $EquivNList(u) := EquivNList(u) \cup \{v\}$ ;
10:  end if;
11: end for;
12: Sort the  $EquivNList(u)$  in descending order according to the  $RedDegree$ ;
13: // Find the group members
14: for all  $v \in EquivNList(u)$  do
15:   if ( $RedDegree(u, v) == 1$ ) then
16:     // They have the same neighbors
17:      $Group(u) := Group(u) \cup \{v\}$ ;
18:   else
19:     if  $v$  is neighbor to all  $w \in Group(u)$  then
20:       Find the  $SNList(v, Group(u))$ ;
21:       Find the number NB of shared neighbors in  $SNList(v, Group(u))$  with
         different degrees;
22:       if  $NB \geq 2$  then
23:         Find the  $Non - CNList(v, Group(u))$ ;
24:         if (all  $k \in Non - CNList(v, Group(u))$  are reachable by at least 2
         nodes  $(l, m) \in SNList(v, Group(u))$ ) then
25:            $Group(u) := Group(u) \cup \{v\}$ ;
26:         end if;
27:       end if;
28:     end if;
29:   end if;
30: end for;

```

Case 2: $T_{cl} = 70\%$

In this case, unlike to node 9, no change can occur for node 6. The equivalent neighbor list of node 9 is formed by nodes 11 and 7, sorted according to their redundancy degrees. Node 9 adds node 11, first, to its group as a member because they have an equivalence of 100%. By cons, for node 7, it checks the eligibility rules. It finds that (1) node 7 is neighbor to both 9 and 11, (2) the shared neighbor list between node 7 and the group $\{9, 11\}$, which is $\{6, 10\}$, does not satisfy the second eligibility rule, because nodes 6 and 10 have the same degree. So, node 7 can not be added, to the group of node 9, as new member because the eligibility rules are not satisfied. While node 7, on the other side, can add node 9 as member to its group since the three eligibility rules are satisfied.

Table 6.2: List of groups formed after the first round for Topo1.

| $T_{cl} = 80\%$ | | $T_{cl} = 70\%$ | |
|-----------------|---------------|-----------------|---------------|
| Node | Group members | Node | Group members |
| 1 | {1, 4, 8} | 1 | {1, 4, 8} |
| 2 | {2, 5} | 2 | {2, 5} |
| 3 | {3, 6, 10} | 3 | {3, 6, 10} |
| 4 | {1, 4, 8} | 4 | {1, 4, 8} |
| 5 | {2, 5} | 5 | {2, 5} |
| 6 | {3, 6, 10} | 6 | {3, 6, 10} |
| 7 | {7} | 7 | {7, 9} |
| 8 | {1, 4, 8} | 8 | {1, 4, 8} |
| 9 | {9, 11} | 9 | {9, 11} |
| 10 | {3, 6, 10} | 10 | {3, 6, 10} |
| 11 | {9, 11} | 11 | {9, 11} |

From this example, we can see clearly the importance of the second eligibility rule. Indeed, without this latter, node 9 adds node 7 to its group since the third eligibility rule is satisfied, which leads to a connectivity loss as shown in Figure 6.1(c).

Table 6.2 shows the list of groups, formed by each node, after the first round for the two cases. From the table, we can see that node 9 is member of two groups which is not allowed in the final grouping because only one active node is elected in each group. Hence, the necessity to exchange the group lists for an eventual (possible) update in the second round.

In order to show the importance of the third eligibility rule, let's take the example, illustrated in Figure 6.2(a). If we consider node 5, for example, its neighbor list, redundancy degrees, and a list of equivalent neighbors are given in Table 6.3. In the case of a T_{cl} value of 60%, node 5 adds node 6, first, to its group as a member because they have an equivalence of 100%. By cons, for node 7, it checks the eligibility rules. It results that (1) node 7 is neighbor to both 5 and 6, (2) the shared neighbor list between node 7 and the group {5,6}, which is {3,8}, satisfies the second eligibility rule, (3) all non-common neighbors between node 7 and the group {5,6} which are {4,9}, are not reachable by two common neighbors. Since the third eligibility rule is not satisfied, node 5 can not add node 7 to its group as a member. However, without this eligibility rule, node 7 can be added to the group {5,6}, which leads to a connectivity loss in the final grouping. Table 6.4 shows the list of groups formed, by each node, after the first round for Topo2 when $T_{cl} = 60\%$.

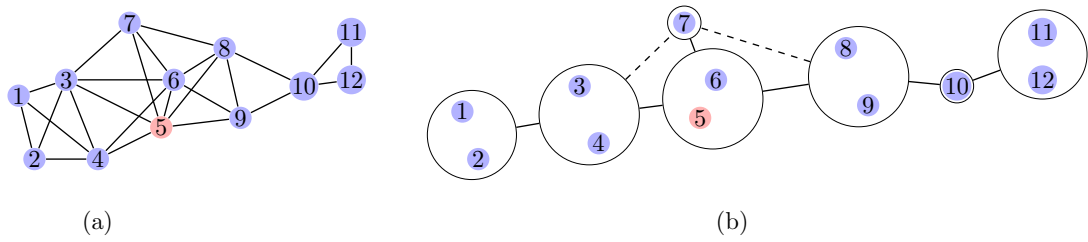


Figure 6.2: (a) The initial network topology (Topo2). (b) The network topology after the group forming phase when $T_{cl} = 60\%$ with connectivity guarantee.

Table 6.3: 5's neighboring table and its equivalent neighbor list for Topo2.

| Node5 | | |
|--|--------------------|-------------------------|
| Node | Neighbor List | Redundancy Degree |
| 3 | {1, 2, 4, 5, 6, 7} | $\frac{3}{7} = 42.85\%$ |
| 4 | {1, 2, 3, 5, 6} | $\frac{2}{7} = 28.57\%$ |
| 6 | {3, 4, 5, 7, 8, 9} | 1 = 100% |
| 7 | {3, 5, 6, 8} | $\frac{3}{5} = 60\%$ |
| 8 | {5, 6, 7, 9, 10} | $\frac{3}{6} = 50\%$ |
| 9 | {5, 6, 8, 10} | $\frac{2}{6} = 33.33\%$ |
| <i>EquivNList</i> (5) when $T_{cl} = 60\%$ | | {6, 7} |

Table 6.4: List of groups formed after the first round for Topo2.

| $T_{cl} = 60\%$ | |
|-----------------|---------------|
| Node | Group members |
| 1 | {1, 2} |
| 2 | {1, 2} |
| 3 | {3, 4} |
| 4 | {3, 4} |
| 5 | {5, 6} |
| 6 | {5, 6} |
| 7 | {7, 5} |
| 8 | {8, 9} |
| 9 | {8, 9} |
| 10 | {10} |

Second Round

In the second round of the group forming phase, nodes exchange the information about their group in order to get the final grouping. Indeed, at the end of the first round, a node can be member of different groups. This is not allowed in the final grouping. So, a node must be member of only one group. Group members should be mutually selected and all members should have the same group members. This is the purpose of the second round. This later takes place as follows:

- Each node broadcasts an update message, containing its *NodeID* and the list of its group members noted *Group*.
- Once all update messages are received by each node, nodes begin updating their group lists based on the group lists received in update messages. The updating process is performed only if the group members do not have the same view about their groups, i.e., they do not have the same members in their groups. In case they have, some members must be deleted.

In order to implement such an updating process, we should specify two points: firstly, which member could be removed from the group?, secondly, who performs this deletion? The updating process is given in Algorithm 6.3.2 and it consists of two steps:

Step 1: Each node first checks, if it is listed, as a member, in its members' group lists. If not, it proceeds to remove members of which it is not member of their group lists. At the end of this step, all group members are mutually chosen but their group lists may still be different.

Step 2: Each node checks, then, if there are members who are not common to all members of its group. In this case, it determines the non-common member list *Non-CMList* with each group member. If the *Non-CMList* is not empty, the member is added to deleted member list *DMList*. The later is sorted in ascending order, according to the number of non-common members (i.e., $-Non-CMList$). Thereafter, if *DMList* is not empty, it removes the member having the most non-common members (the last member in *DMList*). If there are several members with the same number of non-common members, the node removes the member with the lowest *NodeID*. Step 2 is repeated until all members obtain the same group lists.

- At the end of groups updating process, the nodes having modified their group lists send an update message to their group members. Exchanging update messages is necessary to update the group lists after deleting some members.
- Upon receiving the update message, each node only checks if it is still member in its members' group lists. If not, it removes the members of which it is not

member of their group lists.

Algorithm 3 : Round 2 of Group Forming Phase: Updating process

```

1: for all  $v \in Group(u)$  do
2:   if ( $u \notin Group(v)$ ) then
3:      $Group(u) := Group(u) - \{v\}$ ;
4:   end if;
5: end for;
6: //Find the member list having non-common members  $DMList$ ;
7:  $DMList(u) := \{\}$ ;
8:  $Finish := False$ ;
9: repeat
10:  for all  $v \in Group(u)$  do
11:    if ( $v \neq u$ ) then
12:      Find the non-common member list  $Non-CMList(Group(u), Group(v))$ ;
13:      if ( $Non-CMList(Group(u), Group(v)) \neq \emptyset$ ) then
14:         $DMList(u) := DMList(u) \cup \{v\}$ ;
15:         $NB - NCM := |Non-CMList(Group(u), Group(v))|$ ;
16:        Sort the  $DMList(u)$  in ascending order according to the number of
        non-common members  $NB - NCM$ ;
17:      end if;
18:    end if;
19:  end for;
20:  if ( $DMList(u) \neq \emptyset$ ) then
21:    Find the last element  $k \in DMList(u)$ 
22:    if ( $\exists l \in DMList(u)$  with the same  $NB - NCM$  as  $k$ ) then
23:      Choose  $l$  having the lowest  $NodeID$ 
24:       $k := l$ 
25:    end if;
26:     $Group(u) := Group(u) - \{k\}$ ;
27:  else
28:     $Finish := True$ ;
29:  end if;
30: until ( $Finish == True$ );

```

Let's return to the previous examples. At the end of the first round, each node broadcasts an update message. For Topo1 (Figure 6.1), in the case of $T_{cl} = 80\%$, no node executes the updating process. The fact that, all nodes have the same group views as their group members. However, with $T_{cl} = 70\%$, some nodes are members of several groups; it is the case of node 9. Once node 7, receives the different update messages from its neighbors, it finds that it has not been chosen, as a member, by node 9. In this case, node 7 removes node 9 from its group list; and on this action, the group forming phase ends. Figure 6.1 (b) shows the final grouping in the two cases $T_{cl} = 80\%$ and $T_{cl} = 70\%$. For Topo2 (Figure 6.2), node 7 removes node 5 from its group because it is not listed in node 5's group. The resulting topology is given in Figure 6.2(b) from which we can see that the connectivity is still ensured.

Threshold of connectivity level T_{cl}

In this section, we discuss the choice of the T_{cl} value. The application can fix the value of T_{cl} . The higher the value of T_{cl} gets, the higher the number of active nodes which provide some redundancy in transmitted data. This is known as spatial redundancy. Spatial redundancy means the possibility to obtain information for a specific location, from different sources [25]. This type of redundancy is commonly used to provide fault tolerance, to improve the reliability of the measurement data and to increase the level of information security. For example, when a disaster happens, sensor nodes, located near survivors, may detect them and forward the detected survivors' information to the sink(s) for analysis. The duplicated information increases accuracy and helps making the right decisions in these critical situations. However, for some applications, like environmental monitoring and disaster prevention by monitoring disaster prone areas, this redundancy leads to an energy waste. So, with the purpose of monitoring these critical areas, as long as possible, the number of active nodes must be minimized, in order to conserve sensors energy without sacrificing connectivity. This can be achieved by selecting a small value of T_{cl} .

6.3.3 Leaders Election phase

The goal of topology control is to exploit the nodes redundancy to conserve their energy while maintaining network connectivity. In fact, it determines the equivalent nodes and allows those that are redundant to go into sleep mode to conserve their energy. To this end, a sleep-wakeup cycle or sleep scheduling is defined between nodes.

Once the group forming phase is completed, every group starts the sleep/wake scheduling stage. In order to save energy, only one node in each group is required to be active while others turn off their radio. RTCP uses a load balancing strategy for selecting active nodes which we call "Round Robin Scheduling". This strategy allows nodes to periodically represent their group. In other words, the active nodes must be changed periodically to let other nodes become active and represent the group. The objective behind this technique is to keep all nodes alive as long as possible.

The scheduling technique uses the energy level of nodes as an election criterion. Priority is given to the node with a higher energy level to represent the group and become the leader. In case where multiple nodes have the same energy level, the node identity is used as a second criterion. A leader node remains active for a period of activity T_a , calculated according to its estimated lifetime TTL (Time To Live). The TTL value is an estimate of the node lifetime if it works using a maximum of resources (computing, transmitting and receiving). The other group members go in energy conservation mode (sleep mode) during a sleep period, noted T_s . The value of T_s is calculated according to the leader's activity period so that $T_s = T_a$. After the expiry of T_s , nodes wake up and trigger the election process of a new leader. In

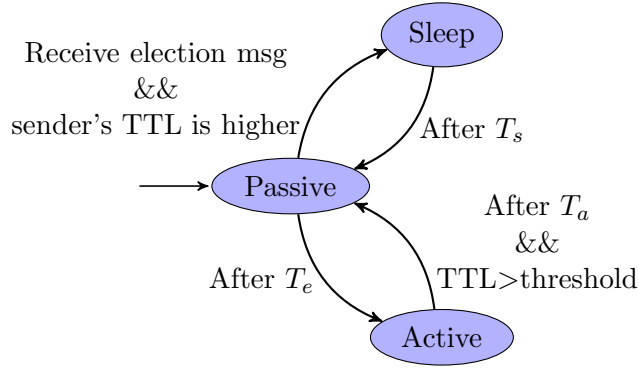


Figure 6.3: RTCP state diagram transitions

order to give nodes the possibility to work periodically and ensure routing data for several periods, the value of T_a must be less than or equal to half of the TTL . Formally, $T_a = TTL/p$ so that $p > 1$ and p represents the number of periods of a node activity.

In RTCP, the nodes are in one of the three following states: *passive*, *active* and *sleep*. The state transitions diagram is given in Figure 6.3. Initially, all nodes start in *passive* state. In this state, each node sets a timer T_e and broadcasts *Election* message, containing the *NodeID* and TTL , to its group members. On receipt of this message, each member compares its energy level with the transmitters' energy level (expressed as TTL). In case where at least one transmitter has a higher priority, a node goes directly to the *sleep* state and sets a timer T_s , according to the maximum value of the received TTL . Otherwise, the node remains in the *passive* state for a period T_e . After the expiration of this period, the node enters the *active* state and provides routing and/or data collection for an activity period T_a . After the expiry of T_s , nodes enter the *passive* state again and trigger the election process of the new leader. The same process is repeated for each round until the energy of nodes reaches a certain threshold. In this case, the node remains active until the exhaustion of its battery.

6.3.4 RTCP complexity analysis

We analyze the complexity of our protocol in terms of communication and time. The analysis concerns only the group forming phase because it significantly affects the protocol effectiveness. Let's assume that n sensors are deployed into an area of interest, and the average connectivity degree of nodes is m .

6.3.5 Communication complexity

In RTCP, each node broadcasts a *discovery message* to acquire the 2-hop neighbors' information. After determining the group members, each node broadcasts an *update message* to exchange information about their groups, and to perform the updating process. At the end of groups updating process, the nodes having modified their group

lists, send *an update message* to their group members. Let's assume that k ($0 \leq k \leq n$), the number of nodes having modified their groups. Knowing that the number of nodes is n , the communication overhead of RTCP is $O(2n + k) \simeq O(n)$.

6.3.6 Time complexity

In RTCP, any computation is, mainly, based on calculating intersection and union. Before presenting the RTCP complexity, we first present, in general, the complexity of the intersection and union of two sets.

Let's assume S_1 and S_2 , two sets of sizes n_1 and n_2 , respectively. The complexity of $S_1 \cap S_2$ or $S_1 \cup S_2$, depends on the initial representation of S_1 and S_2 . Two representations are possible:

- If they are represented by vectors, where each cell contains an element of the set, the complexity is $O(n_1 n_2)$.
- If they are represented as bit vectors, the complexity is therefore $O(n_1)$ or $O(n_2)$.

Lest's assume that node v is running RTCP algorithms.

In the first round of the group forming phase, node v calculates its equivalent neighbor list *EquivNList* with a complexity of $O(m^3)$ if we use the first representation or $O(nm)$ with the second representation. The *EquivNList* size is not known because it depends on the T_{cl} 's value but it does not exceed m . The complexity of sorting this list is $\Theta(m \log m)$, using a merge sort.

Once the *EquivNList* is sorted, node v should filter this list to determine its group members, in order to keep only equivalent neighbors satisfying the eligibility rules. If we assume that all predecessor nodes, of the last node m in the *EquivNList*, satisfy the three eligibility rules (they become group members), the complexity of checking for the m^{th} node, the eligibility rule:

- R1 is $O(m^2)$ with the first representation or is $O(n)$ with the second.
- R2 is $O(km^2)$ with the first representation or is $O(kn)$ with the second.
- R3 is $O(km^2)$ with the first representation or is $O(kn)$ with the second.

where $k = |NList(v) \cap NList(1) \cap \dots \cap NList(m)|$; i.e., $0 \leq k < m$.

So, the complexity to check the three eligibility rules, for all nodes in *EquivNList*, under the assumption that all nodes in *EquivNList* become v 's group members, is $O(km^3)$ with the first representation or is $O(knm)$, with the second one which represents the time complexity of Algorithm 6.3.2.

In the second round, node v checks if there are members who are not common to all members of its group in order to determine *DMList*. Let's assume that l is the average cardinality of v 's group, i.e., $l = |Group(v)|$. So, the complexity, in this case,

is $O(l^3)$ with the first representation or is $O(ln)$ with the second one, where $1 < l < m$. The *DMList* size is not known but it does not exceeds $(l - 1)$. Once the *DMList* is established, and if the *DMList* is not empty, node v removes the member, having the most non-common members. The complexity to remove the adequate member is $O((l - 1))$. The process is repeated until the *DMList* becomes empty. The maximum number of iterations is r where $0 \leq r < l$. So, the time complexity of Algorithm 2 is $O(rl^3)$ with the first representation or is $O(rln)$ with the second one.

Finally, the time complexity of RTCP is $O(km^3)$ with the first representation or is $O(knm)$ with the second one.

6.4 Performance evaluation

In this Section, we evaluate the performance of the proposed protocol RTCP, using Ns2 simulator. RTCP is simulated, using random topologies with different densities. When we examine node density, we vary the number of nodes from 20, 50, 100 and 200 while keeping a 20mx20m constant area.

Table 6.5 summarizes the simulation parameters. The performance evaluation of our solution consists of two parts: Group forming phase and Leaders election phase.

To show the effectiveness of RTCP, we first evaluated it, in terms of number of groups formed and energy consumed, during the groups forming phase. After that, it is evaluated in terms of network lifetime and connectivity when running the sleep/wake-up scheduling phase. To compare the performance of our algorithm with that of GECP, GAF and CPA, we implemented GAF and the distributed version of CPA and simulated them in the same environment.

Table 6.5: Simulation parameters.

| | |
|--------------------------|-------------------------|
| Topography | 20m*20 m |
| Number of nodes | 20, 50, 100, 200 nodes |
| Initial Energy | 5 joule |
| Propagation model | Two-Ray-Ground |
| Radio transmission range | 5m |
| Transmit power | 660mW |
| Receive power | 395mW |
| Idle power | 395mW |
| Sleep power | 100mW |
| T_{cl} | 90%, 80%, 70%, 60%, 50% |
| p | 4 |

Table 6.6: Number of groups formed in RTCP, GAF and CPA under different node densities.

| Node Density | D1=3.92 | D2=9.81 | D3=19.63 | D4=39.67 |
|-------------------------|---------|---------|----------|----------|
| RTCP($T_{cl} = 90\%$) | 13 | 39 | 89 | 186 |
| RTCP($T_{cl} = 80\%$) | 13 | 39 | 86 | 174 |
| RTCP($T_{cl} = 70\%$) | 13 | 38 | 81 | 128 |
| RTCP($T_{cl} = 60\%$) | 13 | 37 | 65 | 96 |
| RTCP($T_{cl} = 50\%$) | 13 | 31 | 52 | 78 |
| GECP | 13 | 39 | 89 | 187 |
| CPA(mindeg=1) | 13 | 26 | 55 | 170 |
| CPA(mindeg=2) | 16 | 33 | 62 | 171 |
| CPA(mindeg=3) | 20 | 39 | 63 | 171 |
| CPA(mindeg=4) | 20 | 45 | 76 | 171 |
| GAF | 81 | 81 | 81 | 81 |

6.4.1 Number of groups formed

Table 6.6 summarizes the results after running the group forming phase for different node densities.

From the results, the first thing we can see is the number of groups (grids), formed by GAF which is the same for all densities; i.e., some grids are empty. Indeed, GAF divides the network into static grids with the same size, using the location information provided by GPS. GECP gives the same number of groups as RTCP when the value of T_{cl} is 90%. On the other side, RTCP and CPA form groups dynamically, using connectivity information. We note that the number of groups formed for both the protocols increases with the connectivity degree *mindeg* for CPA; and with the Threshold of connectivity level T_{cl} for RTCP. This is due to the need for a large number of active nodes, at a given time, to ensure high connectivity. Moreover, with CPA, the number of groups increases with increasing node density because nodes lose their energy during the merging process which leads to stop it. By cons, with RTCP, the number of groups formed is greater than that of CPA for some values of T_{cl} but it is fewer than CPA's for small values of T_{cl} . This is due to the need for more active nodes to provide some data redundancy, required by applications.

So, we can conclude that (1) GAF works better for high node densities, (2) GECP and CPA is suitable for low densities, and (3) RTCP gives acceptable results (the best results for some values of T_{cl}) regardless of node density.

6.4.2 Energy consumed in groups forming phase

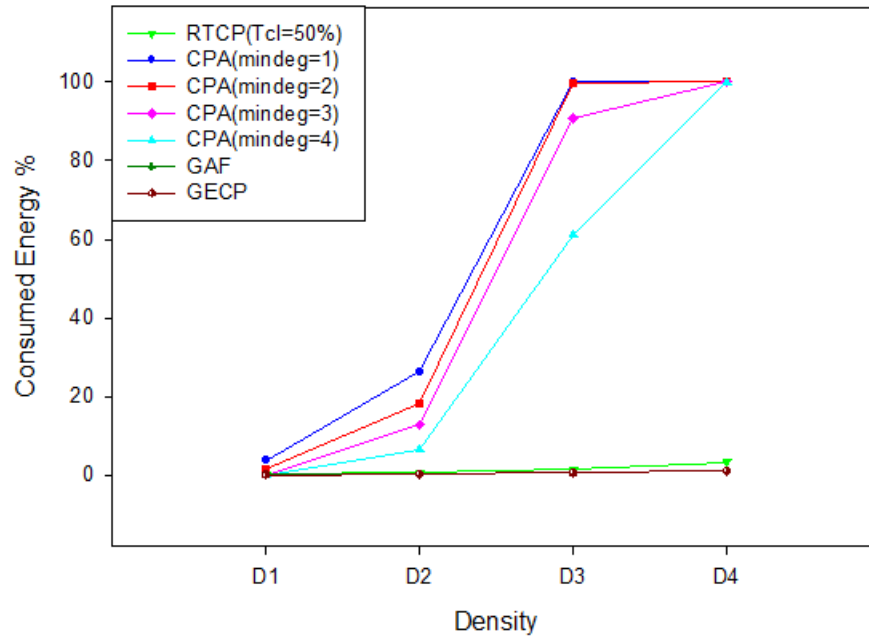
Figure 6.4 shows the energy consumed, during the group forming phase, by RTCP, GAF and CPA. The results show that energy consumption increases by increasing the

number of nodes for each protocol. However, GAF and GECP consume less energy than the other protocols because GAF uses the node position to associate each node to a grid while in GECP node need to exchange only their neighboring lists. By cons, RTCP and CPA consume more energy than GAF and GECP; they exchange messages to ensure connectivity in the final topology. Although, the consumed energy is really negligible for RTCP, compared with that of CPA. Therefore, we can say that with GAF, GECP and RTCP, in the worst case, even though the number of groups is equal to the number of nodes, the energy consumed in the group forming phase does not affect the network lifetime. Unlike what precedes, when running CPA, the obtained results, show that the increase of energy consumption rate is significant. This is what happens, especially, with densities D3 and D4 where all nodes lose their energy without ending the merging process. So, GAF, GECP and RTCP are scalable with the network size while CPA is not.

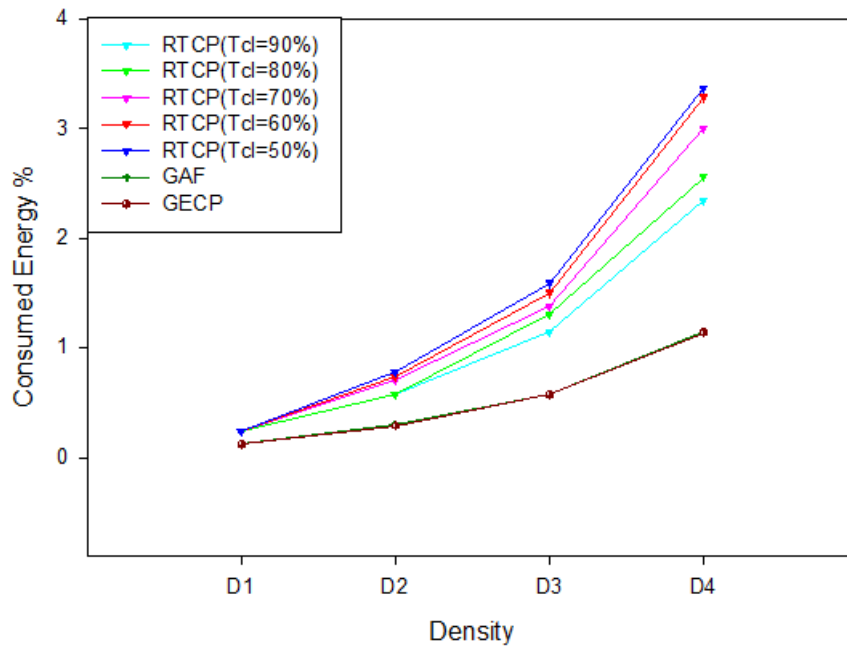
6.4.3 Network lifetime

Figures 6.5, 6.10 and 6.11 show the fraction of survived nodes versus time for node densities D1, D3 and D4 respectively. For better readability of the results concerning the density D2, we preferred to present the results of RTCP, with the different values of T_{cl} , in separate figures. So, Figures 6.6, 6.7, 6.8 and 6.9 show the fraction of survived nodes versus time for node densities D2 when, the T_{cl} 's value is 50%, 60%, 70% and 80%&90%, respectively.

The first thing we observe is that with Flooding, all nodes run out of energy, at about the same time. This is due to the fact that Flooding does nothing to conserve energy and all nodes take part in the routing. By cons, we find that the use of a topology control protocol, based on sleep-scheduling, jointly with a routing protocol enhances significantly the sensor network lifetime; especially when the number of active nodes is reduced. Indeed, the network lifetime is decreased when the value of T_{cl} is increased for RTCP as well as that of *mindeg* for CPA. This is due to the large number of active nodes participating in the collected data transmission, which increases the number of exchanged messages, influencing thereby on the network lifetime. Therefore, GAF gives better results than CPA and so does RTCP for some values of T_{cl} . This is due to the fact that RTCP and CPA allow applications to parameterize the connectivity degree while GAF does not; the degree is static. In addition, despite the high number of active nodes selected by GECP, it outperforms the other protocols, except RTCP($T_{cl} = 50\%$), when the density is low (D1 and D2). On the other hand, when the density is high (D3 and D4), GAF and RTCP gives better results than GECP. Furthermore, we can clearly observe that the RTCP results are better than those of CPA, in terms of the number of survived nodes at each time. This improvement is due to (1) the reduced number of active nodes, and (2) the really negligible energy consumed in the first phase.



(a)



(b)

Figure 6.4: Consumed energy VS Node density.

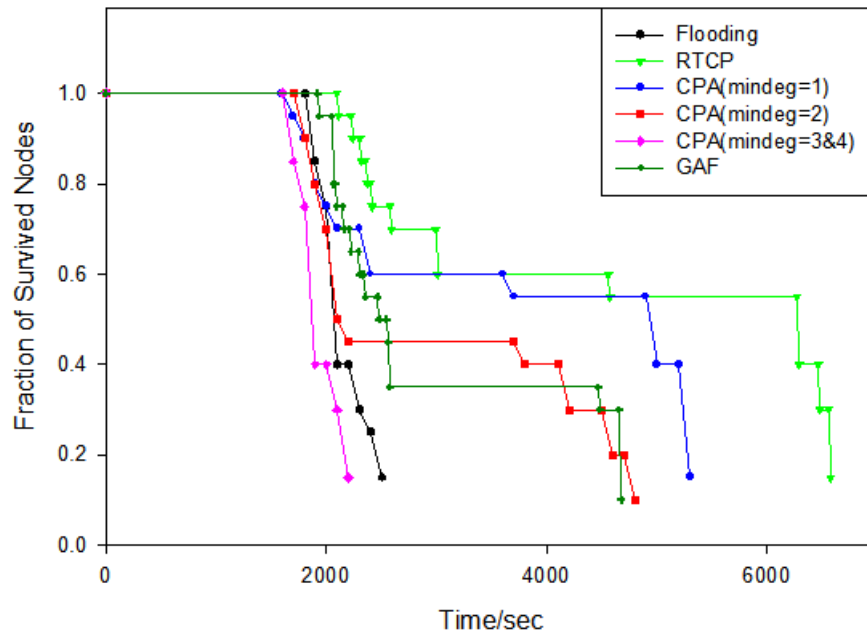
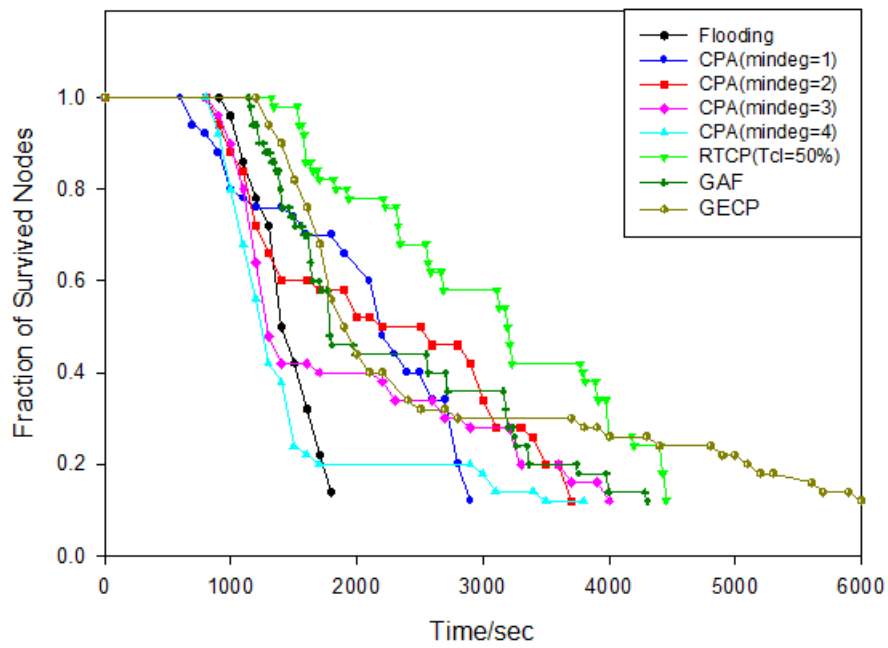


Figure 6.5: Fraction of survived nodes VS time for Node density D1.

Figure 6.6: Fraction of survived nodes VS time for Node density D2 when $T_{cl} = 50\%$.

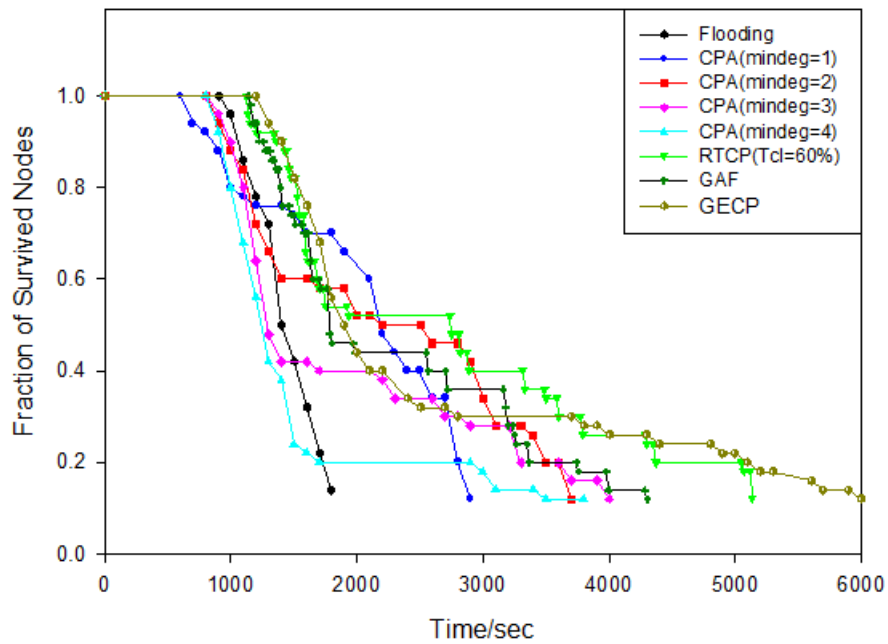


Figure 6.7: Fraction of survived nodes VS time for Node density D2 when $T_{cl} = 60\%$.

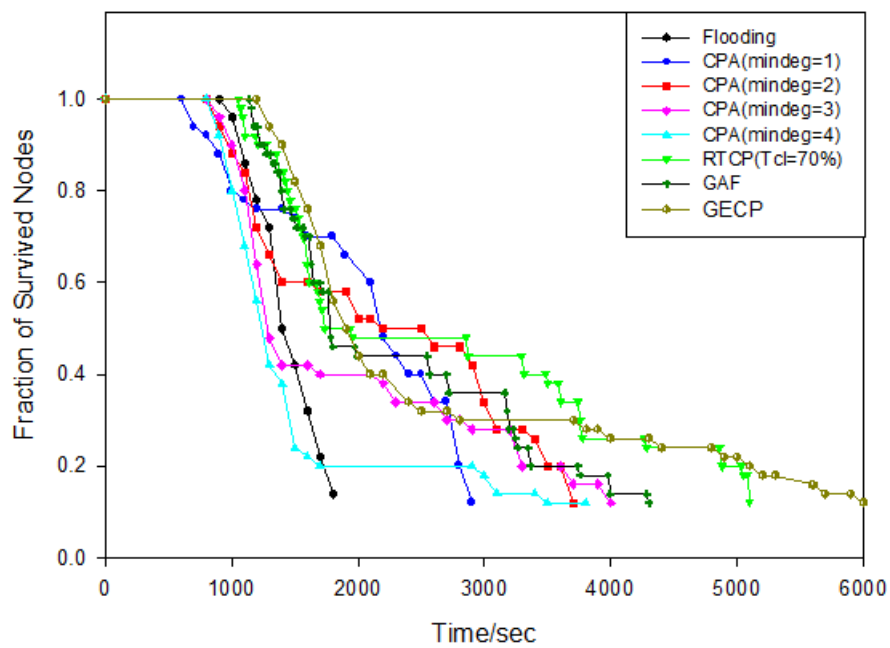


Figure 6.8: Fraction of survived nodes VS time for Node density D2 when $T_{cl} = 70\%$.

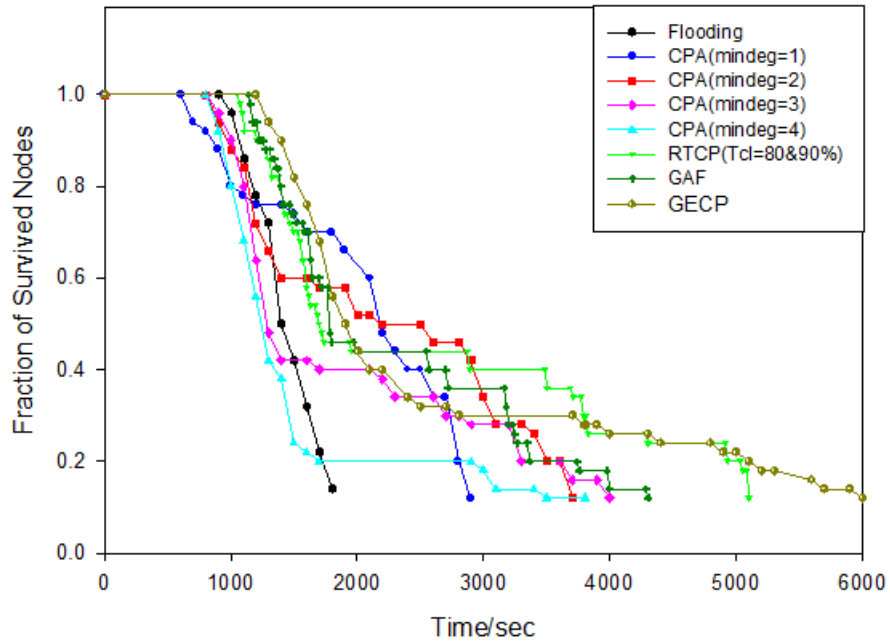


Figure 6.9: Fraction of survived nodes VS time for Node density D2 when $T_{cl} = 80\%$ and $T_{cl} = 90\%$.

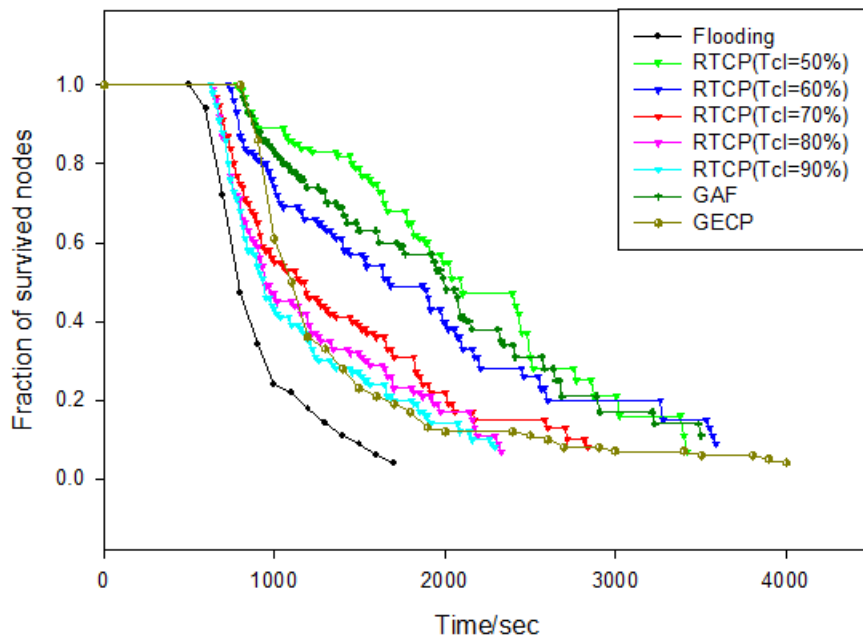


Figure 6.10: Fraction of survived nodes VS time for Node density D3.

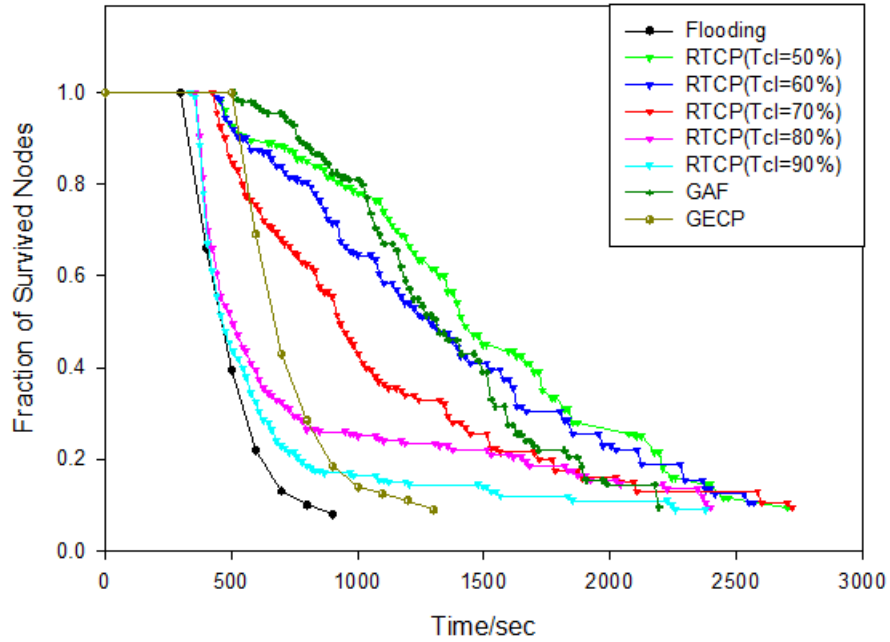


Figure 6.11: Fraction of survived nodes VS time for Node density D4.

6.4.4 Network connectivity

In this section, we examine the topology graphs that result from using RTCP, GECP, GAF and CPA in the case of D2 density. Figure 6.12, 6.13, 6.15 and 6.14 show the corresponding graph, produced by RTCP, GECP, GAF and CPA, respectively. From the graphs, we can see that there are more nodes, selected to be active with GECP and when the value of T_{cl} and $mindeg$ are high for RTCP and CPA, respectively. In addition, we can see that the connectivity is ensured when running GECP, GAF and RTCP with different values of T_{cl} , while a connectivity loss can occur with CPA($mindeg=1$), leading to network partitioning.

6.5 Conclusion

In this chapter, we proposed a protocol, called RTCP, that ensures energy saving, while providing network connectivity. The group forming approach, used by RTCP, organizes nodes into groups with a low communication overhead. The redundant nodes, that satisfy some eligibility rules, can form groups according to their redundancy degrees. This leads to having a reduced number of groups without losing network connectivity. RTCP defines a Threshold of connectivity level which allows applications to parameterize the desired connectivity degree in the reduced topology. Moreover, it uses a load balancing technique that ensures fairness among nodes of the same group. The simulation results show that our solution outperforms some existing solutions, in terms of energy saving, network lifetime and connectivity guarantee. However, RTCP uses the node redun-

dancy degree as criterion to group redundant nodes together with respect to Threshold of connectivity level T_{cl} . Finding the exact value of T_{cl} which gives the minimum number of groups, is difficult when nodes do not have the same redundancy degree. To deal with this, we propose an enhancement to RTCP in the next chapter.

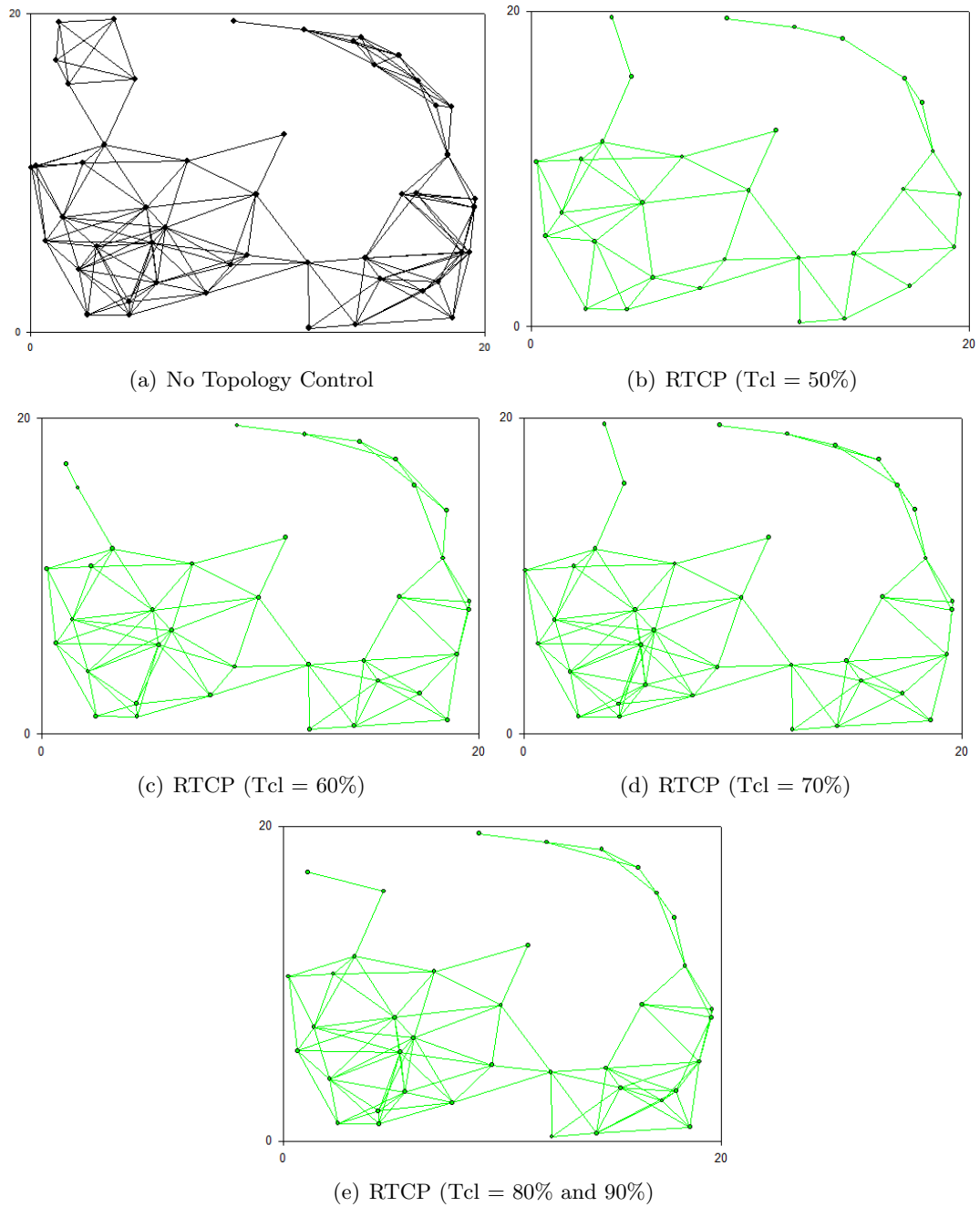


Figure 6.12: The network graphs of no Topology Control and RTCP algorithm at 65s for node density D2.

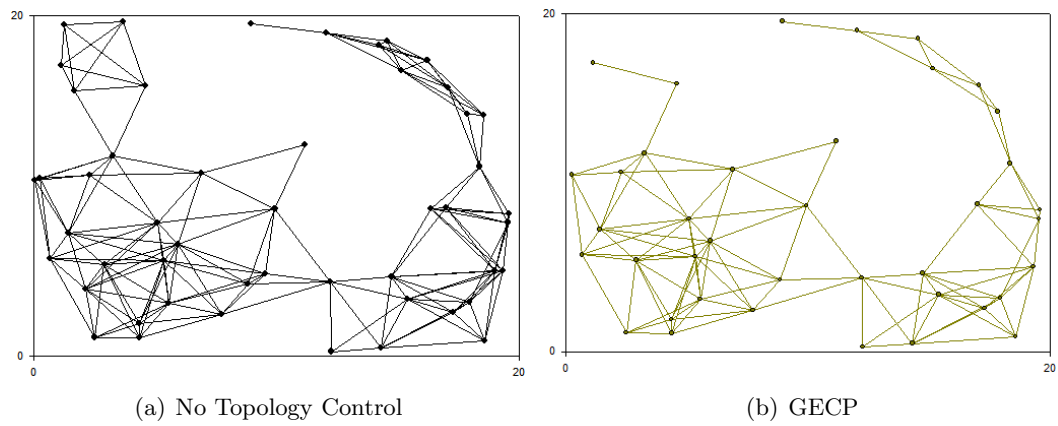


Figure 6.13: The network graphs of no Topology Control and GECP algorithm at 65s for node density D2.

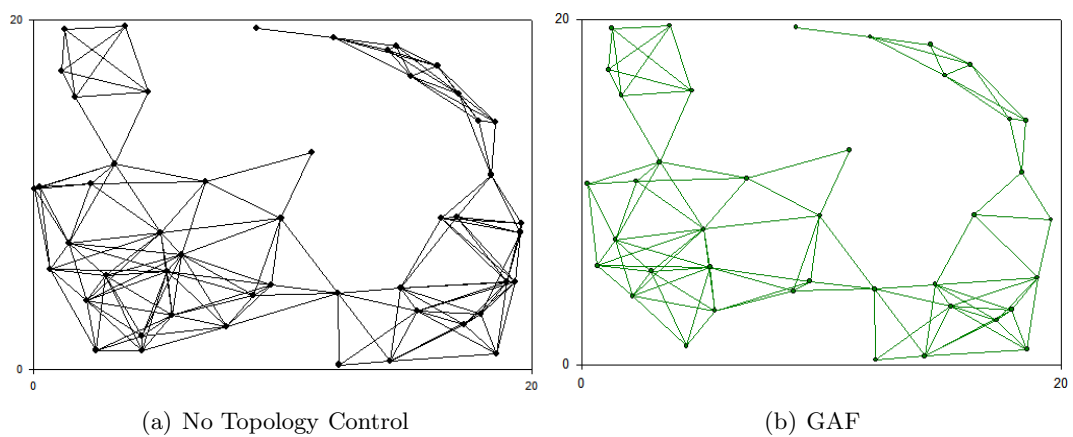


Figure 6.14: The network graphs of no Topology Control and GAF algorithm at 65s for node density D2.

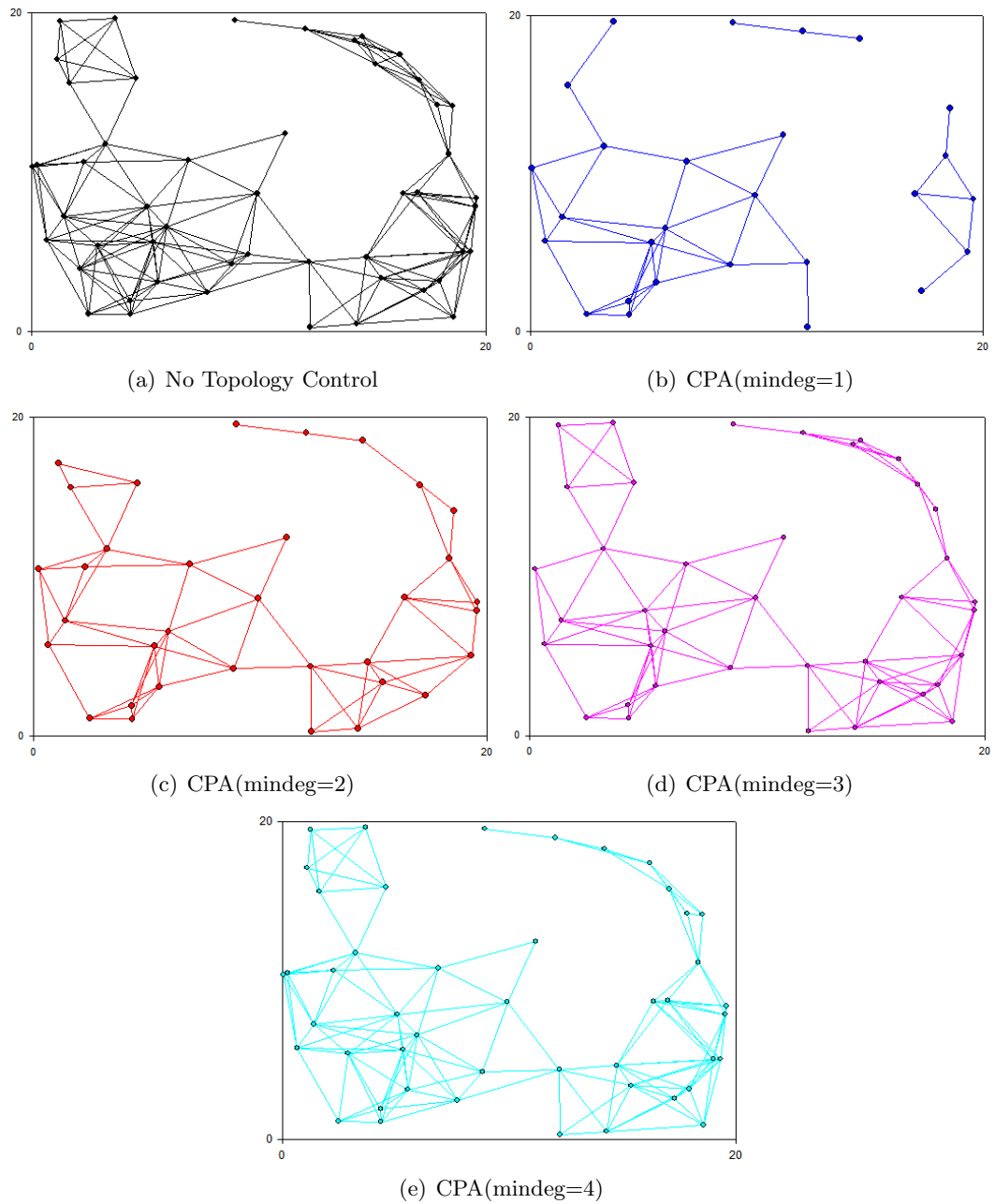


Figure 6.15: The network graphs of no Topology Control and CPA algorithm at 65s for node density D2.

Chapter 7

Enhanced Redundancy aware Topology Control Protocol (ERTCP)

7.1 Introduction

In the previous chapter, a Redundancy aware Topology Control Protocol (RTCP) was presented. As we have seen, RTCP outperforms some other existing algorithms, in terms of energy conservation, network lifetime and connectivity guarantee. This is due to the fact that, in RTCP, nodes use only the 2-hop neighbors' information to locally compute the Redundancy Degree with each of their neighbors. In order to avoid a loss of connectivity, RTCP controls a node to belong to a group, applying three eligibility rules. Moreover, it defines a Threshold of connectivity level which allows applications to parameterize the desired connectivity degree in the reduced topology. Nevertheless, the value of T_{cl} , fixed by the application, affects the number of groups formed. The smallest the value of T_{cl} , the fewer the groups formed. Finding the exact value of T_{cl} which gives the minimum number of groups, is difficult when nodes do not have the same redundancy degree. To overcome this limitation, we propose, in this chapter, ERTCP: an Enhanced Redundancy aware Topology Control Protocol [125].

Thus, this chapter is devoted to present the improvement of RTCP that allows to each sensor node to determine dynamically the value of the Threshold of connectivity level T_{cl} . Indeed, each node should find the minimum value of T_{cl} that provides a connected network. So, ERTCP works the same way as RTCP, except it uses a different metric to determine the equivalent nodes.

7.2 Definition

The Network model and the most important concepts we use are similar to those described in Chapter 6, except the following definition of Equivalent Neighbor List. Indeed, ERTCP did not need to use the Threshold of connectivity level T_{cl} to select

the equivalent neighbors.

Definition 7.1. (*Equivalent Neighbor List*). The equivalent neighbor list of node u , noted $\text{EquivNList}(u)$, is the set of equivalent neighbors providing required communication redundancy with u . Formally, $v \in \text{EquivNList}(u)$, if the following formula is satisfied:

$$\text{RedDegree}(u, v) > 0 \quad (7.1)$$

7.3 Enhanced Redundancy aware Topology Control Protocol

In this section, we give the design details of ERTCP. Like other topology control protocols, ERTCP is divided into two phases, the *Redundant nodes identification* phase and the *Active nodes election* phase.

7.3.1 Redundant nodes identification phase

Once the neighbor lists $NList$ are determined, the identification process of redundant nodes can start. The redundant nodes identification phase of ERTCP is similar to that of RTCP, except in the process of determining the equivalent neighbors. It works in two rounds which are described as follows:

First Round

At the beginning, each node broadcasts a discovery message, containing its identifier $NodeID$ and its neighbor list $NList$. As ERTCP is a distributed scheme, this process enables each node to acquire the 2-hop neighbors' information which would be enough to perform subsequent steps.

Each node, receiving this message, locally computes the Redundancy Degree with each of its neighbors. Thereafter, each node independently calculates its equivalent neighbor list EquivNList . Two nodes are equivalent and can be in the same group, if their Redundancy Degree is greater than zero; i.e., they have at least one shared neighbor. However, this condition is necessary to determine the equivalence between nodes but is not sufficient to group nodes together so that the network connectivity is still preserved. In order to ensure the network connectivity in the final topology, we define additional eligibility conditions. Two nodes u and v are equivalent if they satisfy the equation (7.1) and the three eligibility rules, ER1, ER2 and ER3 given previously in Definition 6.6.

The first round of the group forming process is given by Algorithm 7.3.1.

Algorithm 4 : Round 1 of Redundant nodes identification phase

```

1: Group(u):={u};
2: // Find equivalent neighbors
3: EquivNList(u):={};
4: for all v ∈ NList(u) do
5:   Find shared neighbor list SNList(u,v);
6:   Find union neighbor list UNList(u,v);
7:   RedDegree(u,v):=  $\frac{|SNList(u,v)|}{|UNList(u,v)|}$ ;

8:   if (RedDegree(u,v) > 0) then
9:     EquivNList(u):= EquivNList(u) ∪ {v};
10:  end if;
11: end for;
12: Sort the EquivNList(u) in descending order according to the RedDegree;
13: // Find the group members
14: for all v ∈ EquivNList(u) do
15:   if (RedDegree(u,v)== 1) then
16:     // They have the same neighbors
17:     Group(u) := Group(u) ∪ {v};
18:   else
19:     if v is neighbor to all w ∈ Group(u) then
20:       Find the SNList(v,Group(u));
21:       Find the number NB of shared neighbors in SNList(v,Group(u)) with different
degrees;
22:       if NB ≥ 2 then
23:         Find the Non-CNList(v,Group(u));
24:         for all k ∈ Non-CNList(v,Group(u)) do
25:           if k is reachable by l and m ∈ SNList(v,Group(u)) then
26:             Group(u) := Group(u) ∪ {v};
27:           end if;
28:         end for;
29:       end if;
30:     end if;
31:   end if;
32: end for;

```

Illustration

To better illustrate the first round, we present an example scenario depicted in Figure 7.1. Let's take the example of nodes 6, 7, and 9 in the topology shown in Figure 7.1(a). Their neighbor lists, redundancy degrees, lists of equivalent neighbors are given in Table 7.1.

The equivalent neighbor list of node 9 is formed by nodes $\{11,7,6,10,3\}$, sorted according to their redundancy degrees. Node 9 adds node 11, first, to its group, as a member, because they have an equivalence of 100%, i.e., they have the same set of neighbors. By cons, for the other equivalent nodes, it checks the eligibility rules. It takes node 7, first, it finds that (1) node 7 is neighbor to both 9 and 11, (2) the shared neighbor list between node 7 and the group $\{9,11\}$ which is $\{6,10\}$ does not satisfy the second eligibility rule, because nodes 6 and 10 have the same degree. So, node 7 can not be added to 9's group as new member. While node 7, for its side, can add node 9 as member to its group since the three eligibility rules are satisfied. From this example, we can see clearly the importance of the second eligibility rule. Indeed, without this letter, node 9 adds node 7 to its group since the third eligibility rule is satisfied, which leads to a loss of connectivity. After that, node 9 checks if it can add node 6 as member. In this case, the three eligibility rules are satisfied (1) node 6 is neighbor to both nodes 9 and 11, (2) nodes in the shared neighbor list between node 6 and the group $\{9,11\}$ which is $\{7,3,10\}$, have different redundancy degrees, and (3) all non-common neighbors between node 6 and the group $\{9,11\}$ which are $\{2,5\}$, are reachable by two common neighbors. So, node 6 can be added as member to 9's group. It is not the case for node 10 because the third eligibility rule is not satisfied. The same process is repeated with all nodes in the equivalent list. The group lists of nodes 6, 7 and 9 are given in Table 7.1. From the table, we can see that a node can be member of several groups; it is the case of nodes 6 and 9. This grouping is not allowed because, in the sleep/wake scheduling phase, only one active node is elected in each group and a node can not be, at the same time, in active and sleep mode. Hence the necessity to exchange the group lists for a possible update in the second round.

Second Round

The second round of the redundant nodes identification phase, in ERTCP, works in the same manner of that of RTCP, already described in Section 6.3.2. In this round, nodes exchange the information about their group in order to get the final grouping. Indeed, at the end of the first round, a node can be member of different groups. This is not allowed in the final grouping. So, a node must be member of only one group. Group members should be mutually selected and all members should have the same group members.

Let's go back to the previous example. At the end of the first round, each node

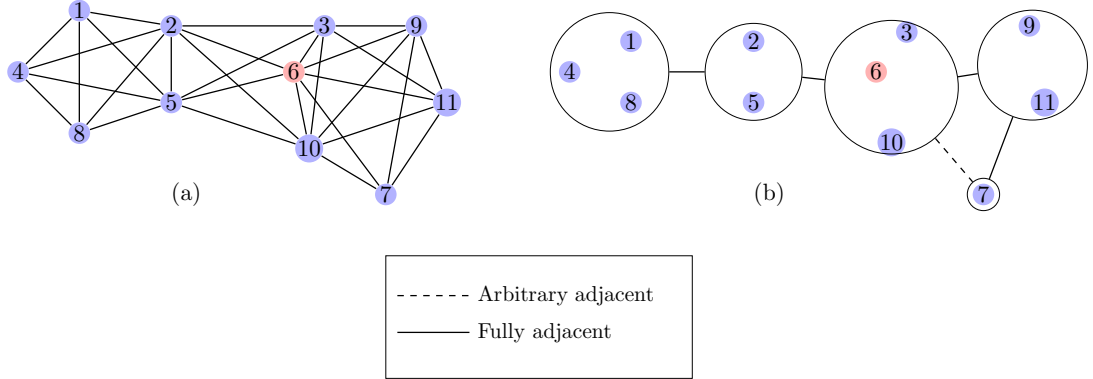


Figure 7.1: (a) The initial network topology represented by a communication graph. (b) The resulting topology, after round 2, with connectivity guarantee.

Table 7.1: 6's, 7's and 9's, neighboring tables and their equivalent neighbor lists.

| Node6 | | |
|-----------------------|------------------------|-------------------------|
| Node | Neighbor List | Redundancy Degree |
| 2 | {1, 3, 4, 5, 6, 8, 10} | $\frac{3}{9} = 33.33\%$ |
| 3 | {2, 5, 6, 9, 10, 11} | $\frac{5}{6} = 83.33\%$ |
| 5 | {1, 2, 3, 4, 6, 8, 10} | $\frac{3}{9} = 33.33\%$ |
| 7 | {6, 9, 10,11} | $\frac{3}{6} = 50\%$ |
| 9 | {3, 6, 7, 10, 11} | $\frac{4}{6} = 66.66\%$ |
| 10 | {2, 3, 5, 6, 7, 9, 11} | 1 = 100% |
| 11 | {3, 6, 7, 9, 10} | $\frac{4}{6} = 66.66\%$ |
| <i>EquivNList</i> (6) | | {10,3,9,11,7,2,5} |
| <i>Group</i> (6) | | {6,10,3} |
| Node7 | | |
| Node | Neighbor List | Redundancy Degree |
| 6 | {2, 3, 5, 7, 9, 10,11} | $\frac{3}{6} = 50\%$ |
| 9 | {3, 6, 7, 10, 11} | $\frac{3}{4} = 75\%$ |
| 10 | {2, 3, 5, 6, 7, 9, 11} | $\frac{3}{6} = 50\%$ |
| 11 | {3, 6, 7, 9, 10} | $\frac{3}{4} = 75\%$ |
| <i>EquivNList</i> (7) | | {9,11,6,10} |
| <i>Group</i> (7) | | {7,9} |
| Node9 | | |
| Node | Neighbor List | Redundancy Degree |
| 3 | {2, 5, 6, 9, 10, 11} | $\frac{3}{6} = 50\%$ |
| 6 | {2, 3, 5, 7, 9, 10,11} | $\frac{4}{6} = 66.66\%$ |
| 7 | {6, 9, 10,11} | $\frac{3}{4} = 75\%$ |
| 10 | {2, 3, 5, 6, 7, 9, 11} | $\frac{4}{6} = 66.66\%$ |
| 11 | {3, 6, 7, 9, 10} | 1 = 100% |
| <i>EquivNList</i> (9) | | {11,6,10,3,7} |
| <i>Group</i> (9) | | {9,11,6} |

broadcasts an update message. Once nodes 7 and 9, receive the different update messages from their neighbors, they find that they have not been chosen, as a member, by node 9 and 6 respectively. In this case, node 7 removes node 9 from its group list and node 9 do the same thing with node 6. The resulting topology is given in Figure 7.1(b) from which we can see that the connectivity is still ensured.

7.3.2 Active nodes election phase

Once the identification of equivalent nodes is completed and the groups are formed, every group starts the stage of sleep/wake scheduling to elect its leader which remain active for a period of time. In order to save energy, only one node in each group is required to be active while others go to sleep mode. ERTCP uses the same load balancing strategy, used by RTCP, for selecting active nodes, that ensures fairness among nodes of the same group.

Figure 7.2 shows the application of the load balancing strategy, used in ERTCP, for three equivalent nodes. Initially, each node sets a timer T_e and broadcasts an election message. In this example, $Node_3$ remains active for T_s period because it has the highest priority while $Node_1$ and $Node_2$ enters a sleep state for a T_s period. After T_s timers of $Node_1$ or/and $Node_2$ expire, they wake up, enter the *passive* state again and trigger the election process of the new leader. In addition, $Node_3$ enters also the *passive* state, after T_a period, and participates to the election of the new leader. The same process is repeated for each round until the nodes' batteries depletion. We note that if the energy of an active node reaches a certain threshold, it does not participate in the election process and remains active until the exhausting of its battery.

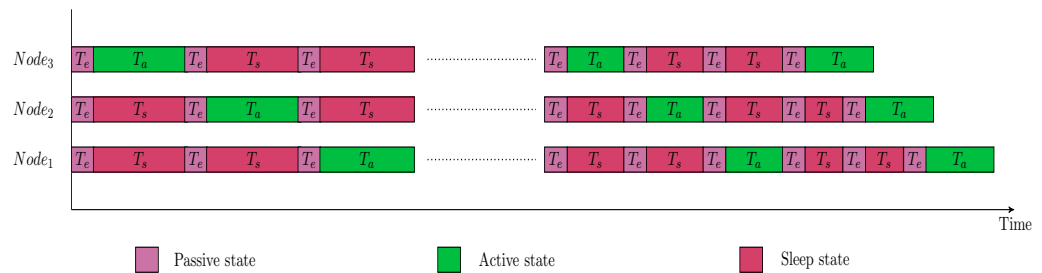


Figure 7.2: The round robin strategy used by ERTCP in the active node election phase.

7.4 Performance evaluation

The performance of ERTCP is evaluated using Ns2 simulator. Our protocol is simulated, using the same random topologies with different densities as previously defined in Chapter 5 and Chapter 6. Table 7.2 summarizes the simulation parameters.

Table 7.2: Simulation parameters.

| | |
|--------------------------|------------------------|
| Topography | 20m*20 m |
| Number of nodes | 20, 50, 100, 200 nodes |
| Initial Energy | 5 joule |
| Propagation model | Two-Ray-Ground |
| Radio transmission range | 5m |
| Transmit power | 660mW |
| Receive power | 395mW |
| Idle power | 395mW |
| Sleep power | 100mW |

7.4.1 Number of groups formed

Table 7.3 summarizes the results after running the redundant node identification phase for the protocols

From the results, the first thing we can see is the number of groups (grids), formed by GAF, which is the same for all densities; i.e., some grids are empty. Indeed, GAF divides the network into static grids with the same size, using the location information provided by GPS. By cons, ERTCP, RTCP, CPA and GECP form groups dynamically using connectivity information. On the other hand, we note that the number of groups formed by ERTCP is less than that of RCTP, CPA, GAF and GECP. GECP gives more groups than the other protocols which is due to the strong constraint used to group nodes together. The number of groups formed for RTCP and CPA increases, respectively, with the Threshold of connectivity level T_{cl} and the connectivity degree $mindeg$. This is due to the need for a large number of active nodes, at a given time, to ensure high connectivity. Moreover, with CPA, the number of groups increases with increasing nodes density because nodes lose their energy during the merging process which leads to stop it. By cons, with RTCP, the number of groups formed is greater than that of CPA. This is due to the need for more active nodes to provide some data redundancy required by applications when the value of T_{cl} is high.

So, we can conclude that (1) GAF works better for high node densities, (2) CPA and GECP are suitable for low densities, (3) RTCP gives acceptable results (the results depend on the values of T_{cl}), and (4) ERTCP gives the best results regardless of node density.

7.4.2 Energy consumed in the redundant nodes identification phase

Figure 7.3 shows the energy consumed, during the group forming phase, by the implemented protocols ERTCP, RTCP, GECP, CPA and GAF. The results show that energy

Table 7.3: Number of groups formed in ERTCP, RTCP, CPA, GAF and GECP under different node densities.

| Node Density | D1=3.92 | D2=9.81 | D3=19.63 | D4=39.67 |
|-------------------------|---------|---------|----------|----------|
| ERTCP | 13 | 29 | 43 | 76 |
| RTCP($T_{cl} = 80\%$) | 13 | 39 | 86 | 174 |
| RTCP($T_{cl} = 50\%$) | 13 | 31 | 52 | 78 |
| GECP | 13 | 39 | 89 | 187 |
| CPA (mindeg=1) | 13 | 26 | 55 | 170 |
| CPA (mindeg=2) | 16 | 33 | 62 | 171 |
| CPA (mindeg=3) | 20 | 39 | 63 | 171 |
| CPA (mindeg=4) | 20 | 45 | 76 | 171 |
| GAF | 81 | 81 | 81 | 81 |

consumption increases by increasing the number of nodes for all protocols. However, GAF and GECP consume less energy than the other protocols. Indeed, GAF uses the node position to associate each node to a grid and GECP uses only the neighbor lists exchanged initially. By cons, ERTCP, RTCP and CPA consume more energy than GAF and GECP; they exchange messages to ensure connectivity in the final topology. Although, the consumed energy is really negligible for ERTCP and RTCP, compared with that of CPA. Therefore, we can say that with ERTCP, RTCP, GAF and GECP, in the worst case, even though the number of groups is equal to the number of nodes, the energy consumed in the group forming phase does not affect the network lifetime. Unlike what precedes, when running CPA, the obtained results, show that the increase of energy consumption rate is significant. This is what happens, especially, with densities D3 and D4 where all nodes lose their energy without ending the merging process. So, ERTCP, RTCP, GECP and GAF are scalable with the network size while CPA is not.

7.4.3 Network lifetime

Figures 7.4, 7.5, 7.6, and 7.7 show the fraction of survived nodes versus time for the densities D1, D2, D3, and D4 respectively.

The first thing we observe is that with Flooding, all nodes run out of energy at about the same time. This is due to the fact that Flooding does nothing to conserve energy and all nodes take part in the routing. By cons, we find that the use of a topology control protocol, based on sleep-scheduling, jointly with a routing protocol enhances significantly the sensor network lifetime; especially when the number of active nodes is reduced. Indeed, the network lifetime is decreased when the value of T_{cl} is increased for RTCP and when the value of *mindeg* is increased for CPA. This is due to the large number of active nodes participating in collected data transmission, which increases the

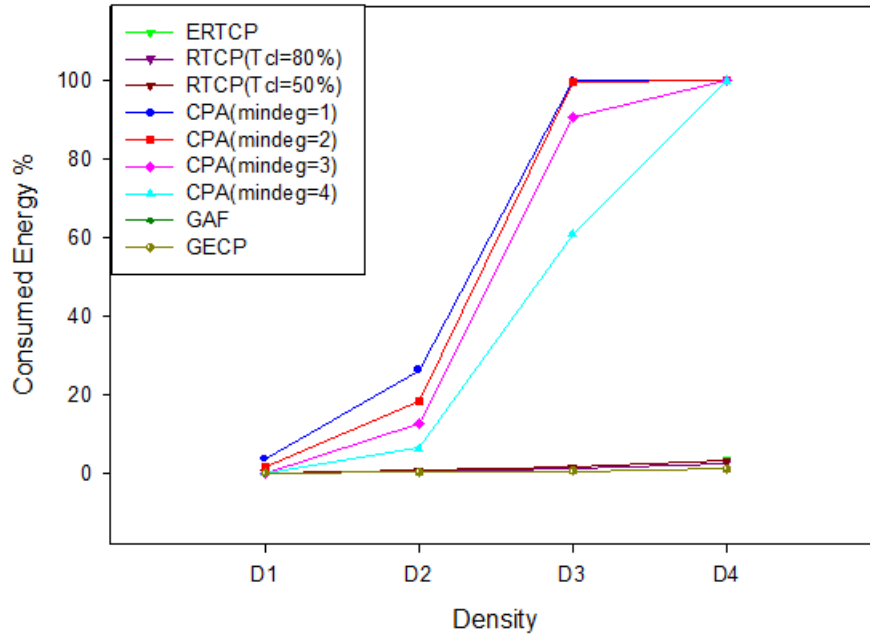


Figure 7.3: Consumed energy VS node density.

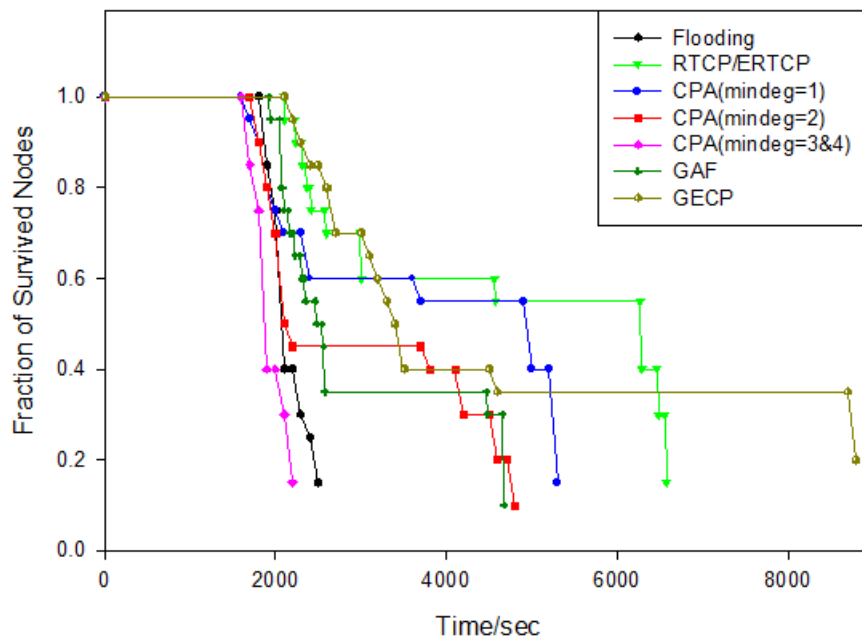


Figure 7.4: Fraction of survived nodes VS time for Density D1.

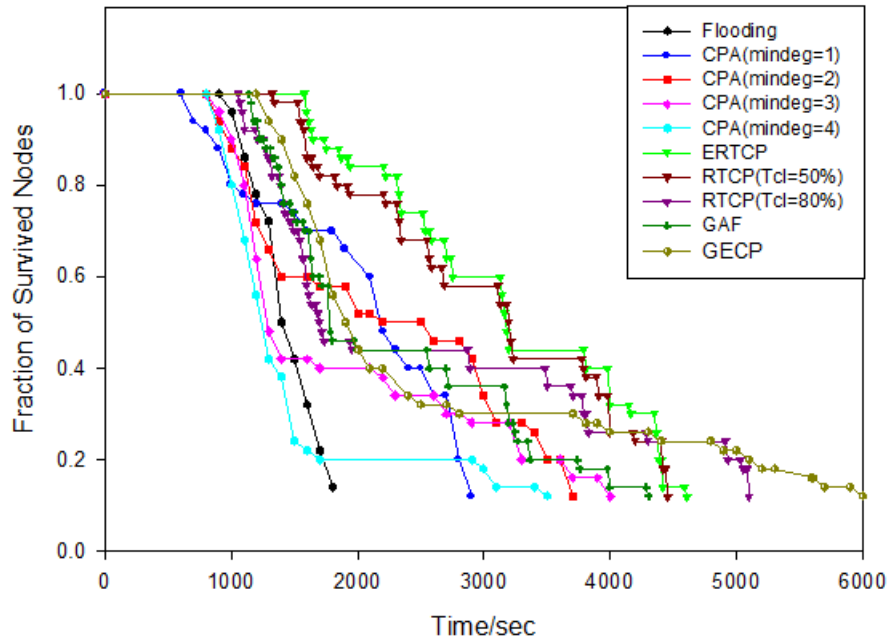


Figure 7.5: Fraction of survived nodes VS time for Density D2.

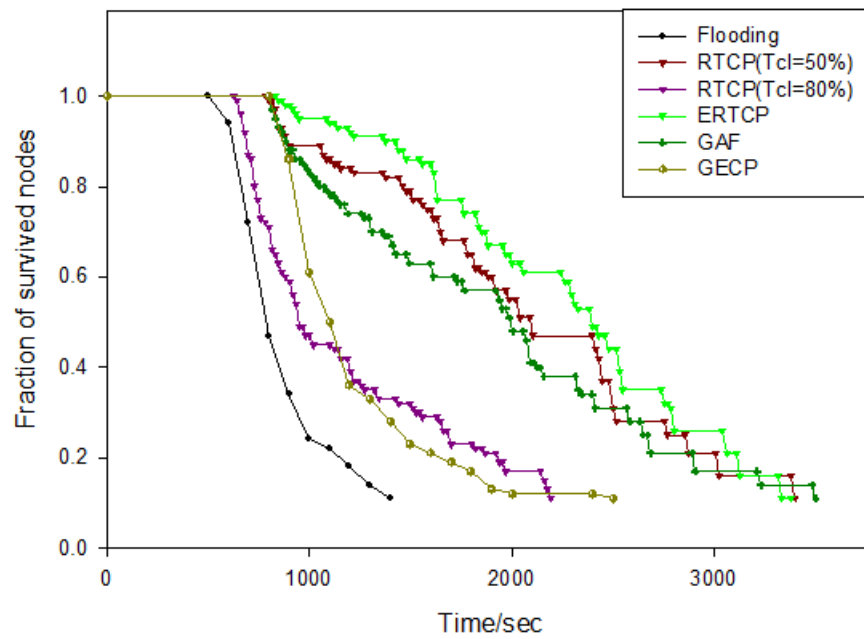


Figure 7.6: Fraction of survived nodes VS time for Density D3.

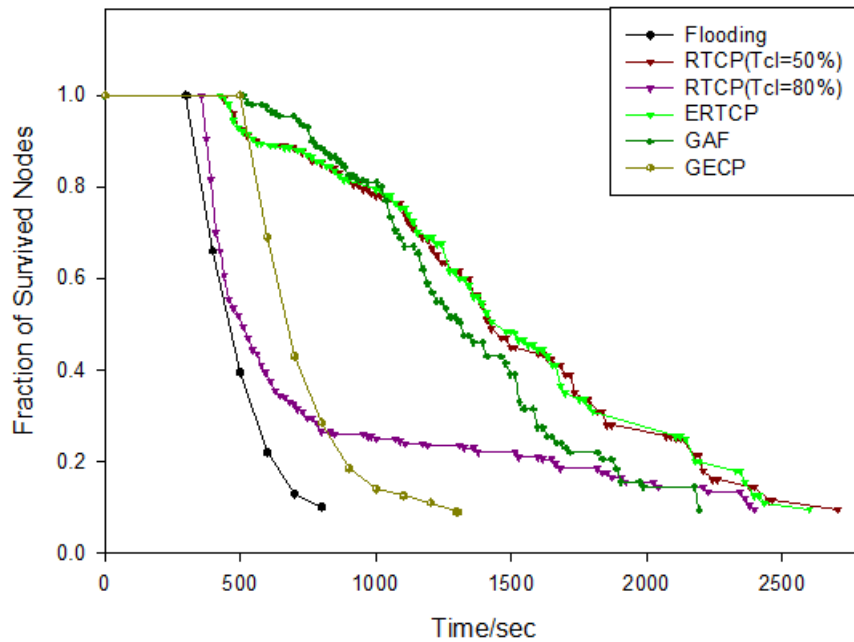


Figure 7.7: Fraction of survived nodes VS time for Density D4.

number of exchanged messages, influencing thereby on the network lifetime. Therefore, GAF gives better results than CPA and RTCP. This is due to the fact that RTCP and CPA allow applications to parameterize the connectivity degree while GAF does not; the degree is static. On the other hand, the results show that, GECP performs better than RTCP, CPA and GAF (for D1 and D2) despite the high number of active nodes produced by running GECP. This is due to the scheduling policy, used by our protocol, that minimizes the number of transitions.

Furthermore, we can clearly observe that the results of ERTCP are better than those of RTCP, CPA, GAF and GECP in terms of the number of survived nodes at each time. This improvement is due to (1) the reduced number of active nodes, and (2) the energy consumed in the groups forming phase which is really negligible.

7.4.4 Network connectivity

In this section, we examine the topology graphs that result from using ERTCP and RTCP in the case of D2 density. Figure 7.8 shows the corresponding graphs. From the graphs, we can see that there are more nodes selected to be active when the value of T_{cl} is high for RTCP. In addition, we can note that the connectivity is ensured when running ERTCP and RTCP (with the different values of T_{cl} .)

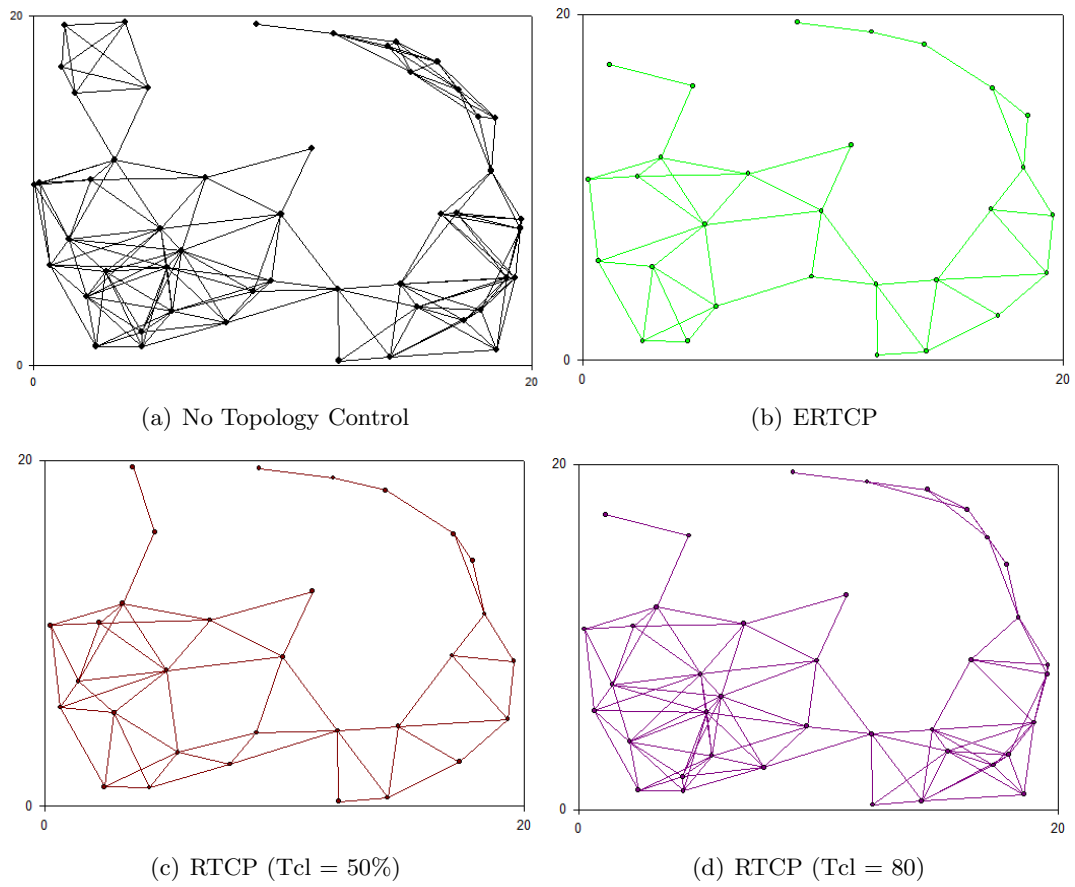


Figure 7.8: The network graphs of no Topology Control, ERTCP and RTCP algorithms at 65s for D2.

7.5 Conclusion

In this chapter, we addressed the energy saving problem using topology control based on duty cycling. A distributed topology control protocol is proposed that guarantees energy saving while providing network connectivity. Our protocol, called ERTCP exploits the sensor redundancy in the same region to reduce the number of active nodes while ensuring a connected backbone. The network connectivity is preserved using a new mechanism to identify the communication node redundancy controlled by some eligibility rules. The group forming approach used by ERTCP organizes redundant nodes into groups with a low communication overhead. Moreover, it uses a load balancing technique that ensures fairness among nodes of the same group. The simulation results confirm the superior performance of the proposed protocol with respect to connectivity and energy efficiency. Furthermore, the network lifetime is extended when ERTCP is used jointly with routing protocol.

Chapter 8

Conclusion

In this thesis, we have dealt with one of the critical issues in wireless sensor networks, which is Energy Saving. After giving some important concepts, we reviewed the main existing energy management approaches for WSN, classified them and discussed their objectives. Besides energy saving, other requirements and QoS metrics can be ensured when using energy conservation approaches. Duty cycling approach, especially *Topology Control*, is the most important energy saving mechanism which meets several objectives. Indeed, topology control plays an important role in saving energy (and hence prolonging network lifetime), improving spatial reuse (and hence increase network capacity), reducing radio interference (and hence mitigating MAC contention), ensuring network coverage and connectivity, among others [129].

Among the existing topology control mechanisms, we focused on topology control based on duty cycling, which allows some redundant nodes to sleep and conserve their energy. Since most energy is spent for radio communications, an effective approach for energy conservation is scheduling sleep intervals for redundant nodes, while the other nodes stay active to provide continuous service. Thus, connectivity can be maintained with much fewer routing nodes and more energy saving.

Several topology control algorithms, using the scheduling technique, have been proposed for wireless ad hoc and sensor networks. We presented and described the most representative existing sleep-scheduling algorithms and classified them into four categories: Flat protocols, Grid-based protocols, Cluster-based protocols and Group-based protocols. Thereafter, we discussed and summarized the key features, objectives, major assumptions of the reviewed sleep scheduling-based topology control protocols. Moreover, the category of each protocol was identified and advantages and limitations have been included. Among the existing categories, we focused on Grid-based and Group-based topology control protocols.

We found that grid-based topology control protocols have an attractive feature that is related to the communication cost and the energy efficiency of determining the redundant nodes. Indeed, they do not incur a high communication overhead during the

grid forming phase, as grids can be built statically once the communication range is known. For this, we proposed a new grid-based algorithm, called GTC: a Geographical Topology Control protocol, that benefits from the advantages of the existing grid-based algorithms to identify redundant nodes. GTC uses a new scheduling strategy which reduces the number of transitions between active and sleep states in order to minimize the transition energy and ensure the network longevity. It also minimizes the frequency of active nodes election and allows some stability in the topology that avoids a loss of connectivity. The proposed protocol is characterized by: (1) a reduced number of messages, (2) a low frequency of leaders' election, (3) a minimum number of state transitions, and (4) a connectivity gain by introducing the grace period. Simulation results showed that GTC displays better performance in terms of network lifetime, data delivery, scalability, and message overhead.

However, the proposed protocol requires the availability of location information provided by GPS, which is sometimes impossible to acquire. This limitation motivates us to define a new mechanism to group nodes dynamically without using any localization system. Thus, our second contribution investigates the group-based topology control category, since it uses only neighborhood information to group redundant nodes together. In this direction, a Group-based Energy-Conserving Protocol (GECP) is proposed and presented in this thesis. GECP exploits the sensor redundancy in the same region by dividing the network into groups so that a connected backbone can be maintained by keeping only one active node in each group and turning off the redundant ones. It identifies redundant nodes, organizes them into groups with a low communication overhead. It uses the same scheduling strategy as GTC which aims to minimize the number of transitions between active and sleep states and, hence, extend the network lifetime. The simulation results showed that our solution outperforms existing group-based solutions in terms of energy saving, network lifetime and connectivity guarantee. Nevertheless, the proposed group-based protocol uses too strong criterion to group equivalent nodes together which leads to high number of active nodes. So, it is necessary to reduce the number of groups (hence, the number of active nodes) using another constraint to identify node redundancy without increasing the communication overhead.

Our third contribution consists on the proposition of another group-based protocol, called RTCP: Redundancy aware Topology Control Protocol, that ensures energy saving, while providing network connectivity. The group forming approach, used by RTCP, organizes nodes into groups with a low communication overhead. The redundant nodes, that satisfy some eligibility rules, can form groups according to their redundancy degrees. This leads to having a reduced number of groups without a network connectivity loss. RTCP defines a Threshold of connectivity level which allows applications to parameterize the desired connectivity degree in the reduced topology. Moreover, it

uses a load balancing technique that ensures fairness among nodes of the same group. The simulation results showed that our solution outperforms some existing solutions, in terms of energy saving, network lifetime and connectivity guarantee. However, RTCP uses the node redundancy degree as criterion to group redundant nodes together with respect to Threshold of connectivity level T_{cl} . Finding the exact value of T_{cl} which gives the minimum number of groups, is difficult when nodes do not have the same redundancy degree. To deal with this, we proposed ERTCP, an Enhanced Redundancy aware Topology Control Protocol. In ERTCP, the network connectivity is preserved using a new mechanism to identify the communication node redundancy controlled by some eligibility rules. The simulation results confirmed the superior performance of the proposed protocol with respect to connectivity and energy efficiency. Furthermore, the network lifetime is extended due to the low number of active nodes compared to other protocols.

We plan to extend our work in two directions. First, in order to achieve high energy-efficiency, it is necessary to take power control and sleep scheduling jointly into account. The use of the power control mechanism on the reducing topology can enhance considerably the network lifetime which, hence, leads to increase the network capacity. Second, as the deployment of nodes is randomly generated, the groups formed by any topology control protocol are heterogeneous in terms of the number of redundant nodes (group size). In other words, these protocols cannot ensure that every group consists of the same number of nodes and they can generate singletons groups with a single node. Thus, during data forwarding, nodes in groups with fewer nodes, especially singleton groups, are lost first due to the early depletion of their batteries which results in a loss of connectivity or a network partitioning. So, routing protocols used for data dissemination, should take into consideration, the heterogeneity of the groups forming the topology. In this direction, we plan to propose a cross layer solution to balance the energy consumption of nodes, especially singleton nodes, in order to prolong the network lifetime and prevent loss of connectivity.

Chapter 9

List of Publications

1. Manel Chenait, [Bahia Zebbane](#), Houda Zeghilet and Nadjib Badache. BSCP: Backup Scheduling Mecanism for Coverage Perserving in WSNs. In Proc. of the Third International Conference on Sensor Technologies and Applications (SENSORCOMM), pages 473-475, June 2009.
2. [Bahia Zebbane](#), Manel Chenait, Houda Zeghilet and Nadjib Badache. Topology Control Protocol for Conserving Energy in Wireless Sensor Networks. In Proc. of the 14th IEEE Symposium on Computers and Communications (ISCC), pages 717-720, July 2009.
3. [Bahia Zebbane](#), Manel Chenait and Nadjib Badache. Towards an Energy-Efficient Algorithm based Sleep-Scheduling for Wireless Sensor Networks. In Proc. of the International Conference on New Technologies, Mobility and Security (NTMS), pages 1-4, May 2012.
4. [Bahia Zebbane](#), Manel Chenait and Nadjib Badache. Energy-Efficient Protocol based Sleep-Scheduling for Wireless Sensor Networks. In Proc. of the International Conference on Complex Systems (ICCS), pages 407-412, November 2012.
5. [Bahia Zebbane](#), Manel Chenait and Nadjib Badache. Enhancing the Sensor Network Lifetime by Topology Control and Sleep-Scheduling. In Proc. of the 4th International Conference on SmArt COmmunications in Network Technologies (SaCoNeT), pages 1-5, June 2013.
6. Manel Chenait, [Bahia Zebbane](#), Hamza Belbezza, Hakim Balli and Nadjib Badache. Distributed and Stable Energy-Efficient Scheduling Algorithm For Coverage in Wireless Sensor Networks. In Proc. of the 9th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), pages 418-423, July 2013.
7. [Bahia Zebbane](#), Manel Chenait and Nadjib Badache. GTC: a Geographical Topology Control Protocol to Conserve Energy in Wireless Sensor Networks. In-

- ternational Journal of Trust Management in Computing and Communications (IJTMCC), Inderscience Publishers, Vol. 1, No. 3-4, pages 320-340, September 2013.
8. Bahia Zebbane, Manel Chenait and Nadjib Badache. Exploiting Node Redundancy for Maximizing Wireless Sensor Network Lifetime. In Proc. of the IEEE/IFIP Wireless Days (WD), pages 1-3, November 2013.
 9. Bahia Zebbane, Manel Chenait and Nadjib Badache. RTCP: Redundancy aware Topology Control Protocol for Wireless Sensor Network. In Proc. of the 1st IEEE International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), pages 118-123, March 2014.
 10. Manel Chenait, Bahia Zebbane, Chafika Benzaid and Nadjib Badache. Sleep scheduling with predictive coverage redundancy check in wireless sensor networks. In Proc. of the 12th International Symposium on Wireless Communication Systems (ISWCS), pages 366-370, August 2015.
 11. Bahia Zebbane, Manel Chenait and Nadjib Badache. A Group-Based Energy-Saving Algorithm for Sleep/Wake Scheduling and Topology Control in Wireless Sensor Networks. International journal of Wireless Personal Communications, Springer, Vol. 84, No. 2, pages 959-983, September 2015.
 12. Manel Chenait, Bahia Zebbane, Lyes Boufennara, Lotfi Haouaya, Chafika Benzaid and Nadjib Badache. Light Weight Sector Eligibility Algorithm for k-Coverage in Wireless Sensor Networks. In Proc. of the 9th International Conference on Next Generation Mobile Applications, Services and Technologies, pages 294-299, September 2015.
 13. Manel Chenait, Bahia Zebbane, Sihem Filali, Ibtissem Belhocine and Chafika Benzaid and Nadjib Badache. A Low-Complex Coverage Eligibility Algorithm For Wireless Sensor Networks. In Proc. of the International Conference on Intelligent Information Processing, Security and Advanced Communication, pages 1-6, November 2015.
 14. Bahia Zebbane, Manel Chenait, Chafika Benzaid and Nadjib Badache. RTCP: a Redundancy aware Topology Control Protocol for Wireless Sensor Networks. International journal of Information and Communication Technology, Inderscience Publishers, to appear.
 15. Bahia Zebbane, Manel Chenait and Nadjib Badache. A Distributed Lightweight Redundancy aware Topology Control Protocol for wireless sensor networks. International journal of Wireless Networks, Springer, DOI: 10.1007/s11276-016-1248-5, 2016.

Bibliography

- [1] Cc2420 data sheet. <<https://inst.eecs.berkeley.edu/cs150/Documents/CC2420.pdf>>, pages 1–87, 2004.
- [2] N. Ababneh. Performance evaluation of a topology control algorithm for wireless sensor networks. *International Journal of Distributed Sensor Networks, Hindawi*, 2010:1–16, July 2010.
- [3] N. Ababneh, A. Viglas, H. Labiod, and N. Boukhatem. Ectc: Energy efficient topology control algorithm for wireless sensor networks. *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks and Workshops (WOWMOM)*, pages 1–9, 2009.
- [4] N. Abanneh and S. Selvadurai. Topology control algorithms for wireless sensor networks: an overview. *International Journal on Wireless & Optical Communications, World Scientific*, 3(1):49–68, April 2006.
- [5] A. A. Abbasi and M. Younis. A survey on clustering algorithms for wireless sensor networks. *Computer Communications, Elsevier*, 30(14-15):2826–2841, October 2007.
- [6] J. N. AI-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11(6):6–28, December 2004.
- [7] H. D. Akaydin, N. Elvin, and Y. Andreopoulos. Energy harvesting from highly unsteady fluid flows using piezoelectric materials. *Journal of Intelligent Material Systems and Structures*, 21(13):1263–1278, March 2010.
- [8] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks, Elsevier*, 3(3):325–349, May 2005.
- [9] I Akyildiz, W Su, Y Sankarasubramaniam, and E. Cayirci. Wireless sensor networks : a survey. *Computer Networks*, 38(4):393–422, March 2002.
- [10] I. F. Akyildiz and M. C. Vuran. Factors influencing wsn design. *Book Chapter: Sensor Wireless Networks, John Wiley&Sons, Ltd*, pages 37–51, June 2010.

-
- [11] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: a survey. *Ad Hoc Networks, Elsevier*, 7(3):537–568, May 2009.
- [12] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. Wang. Controlled sink mobility for prolonging wireless sensor networks lifetime. *Wireless Networks, Springer*, 14(6):831–58, December 2008.
- [13] S. Baydere, Y. Safkan, and O. Durmaz. Lifetime analysis of reliable wireless sensor networks. *IEICE Transactions on Communications E88-B*, 6:2465–2472, December 2005.
- [14] M. Bhardwaj and A. Chandrakasan. Bounding the lifetime of sensor networks via optimal role assignment. In *Proc. of the 21st IEEE Conference on Computer Communications (INFOCOM)*, 3:1587–1596, June 2002.
- [15] M. Busse, T. Haenselmann, and W. Effelsberg. Teca: A topology and energy control algorithm for wireless sensor networks. *Proc. 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems (MSWiM)*, pages 317–321, October 2006.
- [16] B. Carburnar, A. Grama, J. Vitek, and O. Carburnar. Redundancy and coverage detection in sensor networks. *ACM Transactions on Sensor Networks*, 2(1):94–128, February 2006.
- [17] M. Cardei, M. T. Thai, Y. Li, and W. Wu. Energy-efficient target coverage in wireless sensor networks. In *Proc. of the 24th IEEE Conference on Computer Communications INFOCOM*, 3:1976–1984, March 2005.
- [18] A. Cerpa and D. Estrin. Ascent : Adaptive self-configuring sensor networks topologies. *IEEE Transactions on Mobile Computing*, 3(3):272–285, July-August 2004.
- [19] Y. Challal, A. Ouadjaout, N. Lasla, M. Baga, and A. Hadjidj. Secure and efficient disjoint multipath construction for fault tolerant routing in wireless sensor networks. *Network and Computer Applications, Elsevier*, 34(4):1380–1397, July 2011.
- [20] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordinator algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks, Springer*, 8(5):481–494, September 2002.
- [21] D. Chen and P. K. Varshney. Qos support in wireless sensor networks: A survey. *Proc. the International Conference on Wireless Networks (ICWN)*, pages 227–233, June 2004.

- [22] J. Chen, M. Diaz, L. Llopis, B. Rubio, and J. M. Troya. A survey on quality of service support in wireless sensor and actor networks: Requirements and challenges in the context of critical infrastructure protection. *Network and Computer Applications, Elsevier*, 34(4):1225–1239, July 2011.
- [23] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong. Approximate data collection in sensor networks using probabilistic models. *In Proc. of the 22nd International Conference on Data Engineering (ICDE)*, pages 48–59, April 2006.
- [24] L. H. A. Correia, D. F. Macedo, A. L. dos Santos, A. A. F. Loureiro, and J. M. S. Nogueira. Transmission power control techniques for wireless sensor networks. *Computer Networks, Elsevier*, 51(17):4765–4779, December 2007.
- [25] D. I. Curiac, C. Volosencu, D. Pescaru, L. Jurca, and A. Doboli. Redundancy and its applications in wireless sensor networks: A survey. *WSEAS Transactions on Computers*, 8(4):705–714, April 2009.
- [26] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney. Scheduling sleeping nodes in high density cluster-based sensor networks. *Mobile Network and Applications, Springer*, 10(6):825–835, December 2005.
- [27] I. Dietrich and F. Dressler. On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks*, 5(1):1–39, February 2009.
- [28] Y. Ding, C. Wang, and L. Xiao. An adaptive partitioning scheme for sleep scheduling and topology control in wireless sensor networks. *IEEE Transaction on Parallel and Distributed Systems*, 20(9):1352–1365, September 2009.
- [29] B. Doherty, K. S. J. Pister, and L. E. Ghaoui. Convex position estimation in wireless sensor networks. *Proc. 20th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 3:1655–16663, April 2001.
- [30] D. Dondi, A. Bertacchini, D. Brunelli, L. Larcher, and L. Benini. Modeling and optimization of a solar energy harvester system for selfpowered wireless sensor networks. *IEEE Transactions on Industrial Electronics*, 55(7):2759–2766, July 2008.
- [31] P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler. A building block approach to sensor net systems. *Proc. the 6th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pages 267–280, November 2008.
- [32] Bulut E. and I. Korpeoglu. Sleep scheduling with expected common coverage in wireless sensor networks. *Wireless Networks, Springer*, 17(1):19–40, January 2011.

-
- [33] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, 14(2):70–87, April 2007.
- [34] L. Frye and L. Cheng. Topology management for wireless sensor networks. *Book Chapter: Guide to Wireless Sensor Networks, Computer Communications and Networks*, Springer, pages 27–45, 2009.
- [35] R. K. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278, September 1969.
- [36] A. Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-delay trade-off in wireless networks. *Proc. the 23rd IEEE Conference on Computer Communications (INFOCOM)*, pages 464–475, March 2004.
- [37] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin. Networking issues in wireless sensor networks. *Parallel and Distributed Computing, Elsevier*, 64(7):799–814, July 2004.
- [38] L. Gu and J. Stankovic. Radio-triggered wake-up for wireless sensor networks. *Real-Time Systems, Springer*, 29(2):157–182, March 2005.
- [39] Q. Han, S. Mehrotra, and N. Venkatasubramanian. Energy efficient data collection in distributed sensor environments. *In Proc. of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 590–597, March 2004.
- [40] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. Application specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, October 2002.
- [41] K. Hellman and M. Colagrosso. Investigating a wireless sensor network optimal lifetime solution for linear topologies. *Journal of Interconnection Networks*, 7(1):91–100, March 2006.
- [42] D. Incebacak, R. Zilan, B. Tavli, J. M. Barcelo-Ordinas, and J. Garcia-Vidal. Optimal data compression for lifetime maximization in wireless sensor networks operating in stealth mode. *Ad Hoc Networks, Elsevier*, 24(PartB):134–147, January 2015.
- [43] S. Jain, R. Shah, W. Brunette, G. Borriello, and S. Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. *Mobile Networks and Applications, Springer*, 11(3):327–339, June 2006.

- [44] T. Johansson and L. Carr-Motyckova. Reducing interference in ad hoc networks through topology control. *Proc. the Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, pages 17–23, September 2005.
- [45] H. Karl and A. Willig. Protocols and architectures for wireless sensor networks. *Book, Wiley, United Kingdom*, 2005.
- [46] O. Kasten. Energy consumption. *ETH-Zurich, Swiss Federal Institute of Technology. Available at http://www.inf.ethz.ch/kasten/research/bathtub/energy_consumption.html*, 2001.
- [47] A. Keshavarzian, H. Lee, and L. Venkatraman. Wakeup scheduling in wireless sensor networks. *In Proc. of the 7th ACM International symposium on Mobile ad Hoc networking and computing(MobiHoc)*, pages 322–333, May 2006.
- [48] J. A. Khan, H. K. Qureshi, and A. Iqbal. Energy management in wireless sensor networks: a survey. *Computers and Electrical Engineering, Elsevier*, 41:159–176, January 2015.
- [49] H. S. Kim, T. F. Abdelzaher, and W. H. Kwon. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. *Proc. the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 193–204, 2003.
- [50] A. Kurs, A. Karalis, R. Moffatt, J. Joannopoulos, P. Fisher, and M. Soljacic. Wireless power transfer via strongly coupled magnetic resonances. *Science*, 317(5834):83–86, July 2007.
- [51] M. A. Labrador and P. M. Wightman. Topology control. *Book chapter: Topology Control in Wireless Sensor Networks with a companion simulation tool for teaching and research, Springer*, pages 61–70, 2009.
- [52] Y. A. Le Borgne, S. Santini, and G. Bontempi. Adaptive model selection for time series prediction in wireless sensor networks. *Signal Processing, Elsevier*, 87(12):3010–3020, December 2007.
- [53] W. L. Lee, A. Datta, and R. Cardell-Oliver. Flexitp, a flexible-schedule-based tdma protocol for fault-tolerant and energy-efficient wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(6):851–864, June 2008.
- [54] J. Li and P. Mohapatra. Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. *Pervasive Mobile Computing, Elsevier*, 3(3):233–254, June 2007.

- [55] M. Li, Z. Li, and A.V. Vasilakos. A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues. *Proceedings of the IEEE*, 101(12):2538–2557, November 2013.
- [56] N. Li, J. Hou, and L. Sha. Design and analysis of an mst-based topology control algorithm. *IEEE Transactions on Wireless Communications*, 4(3):1195–1206, May 2005.
- [57] Y.J. Li, C.S. Chen, Y. Q. Song, and Z. Wang. Real-time qos support in wireless sensor networks: a survey. *Proc. the 7th IFAC International Conference on Field-buses & Networks in Industrial & Embedded Systems*, pages 373–380, November 2007.
- [58] S. Lin, J. Zhang, G. Zhou, L. Gu, J. Stankovic, and T. He. Atpc: Adaptive transmission power control for wireless sensor networks. *Proc. the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 223–236, October–November 2006.
- [59] A. Liu, J. Ren, X. Li, Z. Chen, and X. S. Shen. Design principles and improvement of cost function based energy aware routing algorithms for wireless sensor networks. *Computer Networks, Elsevier*, 56(7):1951–1967, May 2012.
- [60] H. Liu, P. Wan, C. W. Yi, X. Jia, S. Makki, and P. Niki. Maximal lifetime scheduling in sensor surveillance networks. *In Proc. of the 24th IEEE Conference on Computer Communications (INFOCOM)*, 4:2482–2491, March 2005.
- [61] X-Y Liu, Y. Zhu, L. Kong, C. Liu, Y. Gu, A. V. Vasilakos, and M-Y. Wu. Cdc: Compressive data collection for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(8):2188–2197, August 2015.
- [62] Y. M. Lu and V. W. S. Wong. An energy-efficient multipath routing protocol for wireless sensor networks: Research articles. *Communication Systems, Wiley*, 20(7):747–766, July 2007.
- [63] R. Madan, S. Cui, S. Lall, and A. Goldsmith. Cross-layer design for lifetime maximization in interference-limited wireless sensor networks. *IEEE Transactions on Wireless Communications*, 5(11):3142–3152, November 2006.
- [64] S. Mahfoudh and P. Minet. Survey of energy efficient strategies in wireless ad hoc and sensor networks. *In Proc. of the 7th International Conference on Networking(ICN)*, pages 1–7, April 2008.
- [65] A. Manjeshwar and D. P. Agarwal. Teen: a routing protocol for enhanced efficiency in wireless sensor networks. *In Proc. of the 15th International Parallel and Distributed Processing Symposium*, pages 2009–2015, April 2001.

- [66] F. Marcelloni and M. Vecchio. Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization. *Information sciences, Elsevier*, 180(10):1924–1941, May 2010.
- [67] V. Mhatre and C. Rosenberg. Design guidelines for wireless sensor networks: communication, clustering and aggregation. *Ad Hoc Networks, Elsevier*, 2(1):45–63, January 2004.
- [68] V. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff. A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Transactions on Mobile Computing*, 4(1):4–15, January-February 2005.
- [69] P. Minet. Energy efficient routing. *Book chapter: Ad Hoc and Sensor Wireless Networks: Architectures: Algorithms and Protocols, Bentham Science*, pages 49–68, 2009.
- [70] P. D. Mitcheson, E. M. Yeatman, G. K. Rao, A. S. Holmes, and T. C. Green. Energy harvesting from human and machine motion for wireless electronic devices. *Proceedings of the IEEE*, 96(9):1457–1486, September 2008.
- [71] W. Mo, D. Qiao, and Z. Wang. Mostly-sleeping wireless sensor networks: connectivity, k-coverage, and alpha-lifetime. *In Proc. of the 43rd Annual Allerton Conference on Communication, Control, and Computing*, 2005.
- [72] S. Mueller, R. P. Tsang, and D. Ghosal. Multipath routing in mobile ad hoc networks: Issues and challenges. *Book chapter: Performance Tools and Applications to Networked Systems, Springer*, 2965:209–234, 2004.
- [73] Le N-T. and Y. M. Jang. Energy-efficient coverage guarantees scheduling and routing strategy for wireless sensor networks. *International Journal of Distributed Sensor Networks, Hindawi*, 2015:1–12, January 2015.
- [74] G. Nan, G. Shi, Z. Mao, and M. Li. Cdsws: Coverage-guaranteed distributed sleep/wake scheduling for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 44(1):1–14, December 2012.
- [75] C. S. Ok, S. Lee, P. Mitra, and S. Kumara. Distributed energy balanced routing for wireless sensor networks. *Computers & Industrial Engineering, Elsevier*, 57(1):125–135, August 2009.
- [76] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for sensor networks. *In Proc. of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 95–107, November 2004.

-
- [77] R. V. Prasad, S. Devasenapathy, V. S. Rao, and J. Vazifehdan. Reincarnation in the ambiance: Devices and networks with energy harvesting. *Communications Surveys & Tutorials, IEEE*, 16(1):195–213, February 2014.
- [78] J. Rahme, A. Carneiro Viana, and K. Al Agha. Looking for network functionalities extension by avoiding energy-compromised hotspots in wireless sensor networks. *Annals of Telecommunications, Springer*, 63(9):487–500, October 2008.
- [79] R. Rajagopalan and P. K. Varshney. Data-aggregation techniques in sensor networks: a survey. *IEEE Communications Survey & Tutorials*, 8(4):48–63, January 2006.
- [80] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks, Springer*, 12(1):63–78, February 2006.
- [81] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. *Proc. the 19th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies*, 2:404–413, March 2000.
- [82] T. Rault, A. Bouabdallah, and Y. Challal. Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks, Elsevier*, 67:104–122, July 2014.
- [83] A.V. Renold and S. Chandrakala. Survey on state scheduling-based topology control in unattended wireless sensor networks. *Computers and Electrical Engineering, Elsevier, In press, doi:10.1016/j.compeleceng.2015.12.024*, 2016.
- [84] I. Rhee, A. Warrier, M. Aia, J. Min, and M. Sichitiu. Z-mac: a hybrid mac for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 16(3):511–524, June 2008.
- [85] K. Romer and F. Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, December 2004.
- [86] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Computing Surveys*, 37(2):164–194, June 2005.
- [87] W. K. G. Seah, Y. K. Tan, and A. T. S. Chan. Research in energy harvesting wireless sensor networks and the challenges ahead. *Book Chapter: Autonomous Sensor Networks, Collective Sensing Strategies for Analytical Purposes, Springer*, 13:73–93, 2013.

- [88] C. Sengul, A. Harris, and R. Kravets. Reconsidering power management. *In Proc. of the 4th International Conference on Broadband Communications, Networks and Systems (BROADNETS)*, pages 799–808, September 2007.
- [89] K. Sha, J. Gehlot, and R. Greve. Multipath routing techniques in wireless sensor networks: A survey. *Wireless Personal Communications, Springer*, 70(2):807–829, May 2013.
- [90] K. Sha and W. Shi. Modeling the lifetime of wireless sensor networks. *Sensor Letters*, 3(2):126–135, June 2005.
- [91] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks, Elsevier*, 1(2-3):215–233, September 2003.
- [92] G. Shi, G. Nan, J. Kou, and R. Rong. Comprehensive review of sleep/wake scheduling in wireless sensor networks. *Chapter in: series Communications in Computer and Information Science, High Performance Networking, Computing, and Communication Systems, Second International Conference ICHCC 2011, Springer*, 139:492–499, May 2011.
- [93] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. *Proc. 7th ACM/IEEE Conference on Mobile Computing and Networks (MOBICOM)*, pages 272–287, July 2001.
- [94] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki. Practical data compression in wireless sensor networks: A survey. *Network and Computer Applications, Elsevier*, 35(1):37–59, January 2012.
- [95] S. Sudevalayam and P. Kulkarni. Energy harvesting sensor nodes: survey and implications. *Communications Surveys & Tutorials, IEEE*, 13(3):443–461, September 2011.
- [96] C. Tang and C. S. Raghavendra. Compression techniques for wireless sensor networks. *Book chapter: Wireless Sensor Networks, Kluwer Academic Publishers*, pages 207–231, 2004.
- [97] M. Tarique, K. E. Tepe, and S. Adibi, S. Erfani. Survey of multipath routing protocols for mobile ad hoc networks. *Network and Computer Applications, Elsevier*, 32(6):1125–1143, November 2009.
- [98] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. *In Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 32–41, 2002.

-
- [99] D. Tian and N. D. Georganas. Connectivity maintenance and coverage preservation in wireless sensor networks. *Ad Hoc Networks, Elsevier*, 3(6):744–761, November 2005.
- [100] G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition, Elsevier*, 12(4):261–268, 1980.
- [101] M. C. Vuran, O. B. Akan, and I. F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks, Elsevier*, 45(3):245–259, June 2004.
- [102] A. Y. Wang and C. G. Sodini. A simple energy model for wireless microsensor transceivers. *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, 5:3205–3209, November-December 2004.
- [103] W. Wang, V. Srinivasan, and K.C. Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. *In Proc. of the 11th ACM International Conference on Mobile Computing and Networking (MobiCom), Germany*, pages 270–283, August-September 2005.
- [104] X Wang, G Xing, Y Zhang, C Lu, R Pless, and C Gill. Integrated coverage and connectivity configuration in wireless sensor networks. *ACM Transactions on Sensor Networks*, 1(1):36–72, August 2005.
- [105] Y. C. Wang, Y. Y. Hsieh, and Y. C. Tseng. Multiresolution spatial and temporal coding in a wireless sensor network for long-term monitoring applications. *IEEE Transactions on Computers*, 58(6):827–838, June 2009.
- [106] A. Warriier, S. Park, J. Mina, and I. Rhee. How much energy saving does topology control offer for wireless sensor networks? a practical study. *Computer Communications, Elsevier*, 30(14-15):2867–2879, October 2007.
- [107] G. Wei, Y. Ling, B. Guo, B. Xiao, and A. V. Vasilakos. Prediction-based data aggregation in wireless sensor networks: Combining grey model and kalman filter. *Computer Communications, Elsevier*, 34(6):793–802, May 2011.
- [108] X. Wei, Z. Wang, and H. Dai. A critical review of wireless power transfer via strongly coupled magnetic resonances. *Energies*, 7:4316–4341, July 2014.
- [109] Lu X. and S. H. Yang. Thermal energy harvesting for wsns. *In Proc. of the IEEE International Conference on Systems Man and Cybernetics (SMC)*, pages 3045–3052, October 2010.
- [110] F. Xia. Qos challenges and opportunities in wireless sensor/actuator networks. *Sensors*, 8(2):1099–1110, 2008.

-
- [111] L. Xie, Y. Shi, Y. T. Hou, W. Lou, H. D. Sherali, and S. F. Midkiff. On renewable sensor networks with wireless energy transfer: the multi-node case. *In Proc. of the 9th Annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks (SECON)*, pages 10–18, June 2012.
- [112] L. Xie, Y. Shi, Y. T. Hou, and H. D. Sherali. Making sensor networks immortal: an energy-renewal approach with wireless power transfer. *IEEE/ACM Transactions on Networking*, 20(6):1748–1761, December 2012.
- [113] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. *In Proc. of the 7th annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, pages 70–84, July 2001.
- [114] F. Ye, Zhong G., J. Cheng, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. *Proc. 23rd IEEE International Conference on Distributed Computing Systems*, pages 28–37, May 2003.
- [115] W. Ye, J. Heidemann, and Estrin D. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, June 2004.
- [116] B. Yin, H. Shi, and Y. Shang. A two-level strategy for topology control in wireless sensor networks. *Proc. 11th International Conference on Parallel and Distributed Systems, (ICPADS)*, pages 358–362, 2005.
- [117] M. Younis and T. Nadeem. Energy efficient mac protocol for wireless sensor networks. *Book Chapter: Wireless Ad Hoc and Sensor Networks, Ed. Ahmed Safwat, Kluwer Academic Publishers*, 2005.
- [118] M. Younis, I. F. Senturk, K. Akkaya, S. Lee, and F. Senel. Topology management techniques for tolerating node failures in wireless sensor networks: A survey. *Computer Networks, Elsevier*, 58:254–283, January 2014.
- [119] B. Zebbane, M. Chenait, and N. Badache. Energy-efficient protocol based sleep-scheduling for wireless sensor networks. *Proc. 1st IEEE International Conference on Complex Systems (ICCS)*, pages 407–412, November 2012.
- [120] B. Zebbane, M. Chenait, and N. Badache. Towards an energy-efficient algorithm based sleep-scheduling for wireless sensor networks. *Proc. 5th IFIP/IEEE International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–4, May 2012.
- [121] B. Zebbane, M. Chenait, and N. Badache. Exploiting node redundancy for maximizing wireless sensor network lifetime. *Proc. IEEE/IFIP Wireless Days (WD)*, pages 1–3, November 2013.

- [122] B. Zebbane, M. Chenait, and N. Badache. Gtc: a geographical topology control protocol to conserve energy in wireless sensor networks. *International Journal of Trust Management in Computing and Communications (IJTMCC)*, Inderscience Publishers, 3/4(1):320–340, September 2013.
- [123] B. Zebbane, M. Chenait, and N. Badache. Rtcp: Redundancy aware topology control protocol for wireless sensor networks. *Proc. 1st IEEE International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pages 118–123, March 2014.
- [124] B. Zebbane, M. Chenait, and N. Badache. A group-based energy-saving algorithm for sleep/wake scheduling and topology control in wireless sensor networks. *Wireless Personal Communications*, Springer, 84(2):959–983, September 2015.
- [125] B. Zebbane, M. Chenait, and N. Badache. A distributed lightweight redundancy aware topology control protocol for wireless sensor networks. *Wireless Networks*, Springer, DOI: 10.1007/s11276-016-1248-5, 2016.
- [126] B. Zebbane, M. Chenait, C. Benzaid, and N. Badache. Rtcp: a redundancy aware topology control protocol for wireless sensor networks. *International Journal of Information and Communication Technology (IJICT)*, Inderscience Publishers, To appear.
- [127] B. Zebbane, M. Chenait, H. Zeghilet, and N. Badache. Topology control protocol for conserving energy in wireless sensor networks. *Proc. 14th IEEE Symposium on Computers and Communications (ISCC)*, pages 717–720, July 2009.
- [128] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Wireless Ad Hoc and Sensor Networks*, 1(1-2):89–124, November 2005.
- [129] X. Zhang, X. Y. Ding, S. Lu, and G. Chen. Principles for energy-efficient topology control in wireless sensor networks. *Proc. the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom)*, pages 1–3, September 2009.
- [130] X. M Zhang, Y Zhang, F Yan, and A. V Vasilakos. Interference-based topology control algorithm for delay-constrained mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 14(4):742–754, April 2015.
- [131] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. *Proc. the 5th ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc)*, pages 187–198, May 2004.

