Islamic University – Gaza
Deanery of Higher Studies
Faculty of Engineering
Computer Engineering Department

Interference Reduction in Mobile Ad Hoc and Sensor Networks

By
Maaly A. S. Hassan

Supervisor:

Prof. Ibrahim S. I. Abuhaiba

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

1431H (2010)

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Ibrahim Abuhaiba, for his guidance, encouragement, enthusiasm, professional advice, and profound understanding. He guided me all the way though my graduate studies, and his strong feedback has contributed greatly to my thesis. Therefore, I am deeply indebted to him for his time, help, and for providing a strong foundation of understanding from where I can undertake a research activity.

In the course of my study, I have had the benefit of interacting with my section-mates and fellow graduate students, and I would like to thank them for their advices and friendship. I am grateful to all my friends at Islamic University of Gaza for giving me enough strength during my ups and downs. In particular, I would like to thank my closest friends Huda Hubboub and Suhir Elkord, both of them of exceptional caliber. It was indeed my good fortune to collaborate with them, and be their friend. I also thank all friends who made my studying days at IUG more pleasurable, enjoyable, and memorable.

My greatest respects go to my sisters and brothers for their endless love, support, and for raising me. God gave me the best when he gave me you. I must particularly acknowledge my sister Manar; your support has been unwavering and I am indebted to you for your sacrifices. What do people do without sisters?

I cannot find words to express my earnest gratitude to my parents who not only brought me to this world but also served as a constant source of support, love, inspiration, and encouragement. Without their continued help I would have never been where I am academically today.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CBTC	Cone Based Topology Control
CDS	Connected Dominating Set
CDS-Rule K	Connected Dominating Set Rule K
COMPOW	COMmon POWer
CRM	Critical Transmission Range
d	The Euclidean distance
DGETRec	Dynamic Global Energy Topology Recreation
DGTRec	Dynamic Global Topology Recreation
DL-DSR	Dynamic Local DSR-based TM
DSR	Dynamic Source Routing
EECDS	Energy Efficient Connected Dominating Set
GG	Gabriel Graph
GRG	Geometric Random Graph
HEED	Hybrid Energy-Efficient Distributed clustering
HGTRotRec	Hybrid Global Topology Rotation and Recreation
HM	Hello Message
IA	Interference Amount
IACDS	Interference-Aware Connected Dominating Set
IARR	Interference Amount Reduction Ratio
INS	Interference Neighbors Set
IP	Internet Protocol
IR	Interference Range
LEACH	Low Energy Adaptive Clustering Hierarchy
LIFE	Low Interference Forest Establisher
LMST	Local Minimum Spanning Tree
MANETs	Mobile Ad hoc NETworks
MDS	Minimum Dominating Set
MIS	Maximal Independent Sets

PRM	Parent Recognition Message
RNG	Relative Neighborhood Graph
RSSI	Received Signal Strength Indicator
SGTRot	Static Global Topology Rotation
SINR	Signal to Noise Ratio
SM	Sleeping Message
TC	Topology Control
TM	Topology Maintenance
TR	Transmission Range
w.h.p.	with high probability
WiFi	Wireless Fidelity
WSN	Wireless Sensor Network
XTC	exotic, extreme, exceptional, exemplary, extravagant, extraterrestrial Topology Control
YG	Yao Graph

تقليص التداخل في الشبكات المتنقلة العشوائية والاستشعارية

مقدم من:
معالي عوض صابر حسن

الملخص

لاقى حقل الشبكات المتنقلة العشوائية وشبكات الاستشعار اللاسلكية في العقد الماضي اهتماما متزايداً نتيجة للتوسع الملحوظ في مجالاته العملية وتطوراته التقنية، كما ساعد الاتصال اللاسلكي على سهولة انتشار المعلومات وتواصلها بما يفوق قدرة مجال الإنترنت السلكي، غير أنه لا يزال هناك تحديات جديرة بالاهتمام تواجه نمو هذا المجال، فمثلاً طوبولوجيا هذا النوع من الشبكات تجعلها ضعيفة لعدة إعتبارات منها تعرض مثل هذه الشبكات للتداخل ومدى تأثيره على كفاءة و فاعلية هذا النوع الخاص من الشبكات. فإذا كانت طبولوجيا الشبكة تعاني من التداخل، فإن هذا سيؤدي إلى تصادم إشارات الإتصال التي ترسلها العقد المتراسة فيما بينها، كما سيسبب تأخيرا جديا في تسليم البيانات للعقد المعنية، ومن جهة أخرى سينتهي كل هذا باستهلاك المزيد من الطاقة والتي هي نقطة ضعف هذه الشبكات. لذلك، نصل إلى استنتاج مفاده أن التداخل يشكل تأثير سلبي ملحوظ على أداء الشبكات المتنقلة العشوائية والاستشعارية، ففي مجال البحث توصل الباحثون إلى استخدام آليات التحكم في طوبولوجيا هذه الشبكات في حل الكثير من المشاكل التي تواجهها. الحافز من وراء استخدام تقنية التحكم بالطوبولوجيا هو الحفاظ على الارتباط بين العقد في الشبكة، وكذلك للحد من درجة العقدة والذي يعني الحد من التداخل، ومن جهة أخرى الحد من استهلاك الطاقة في العقد الاستشعارية. كما تشير بعض الدراسات أن العقدة يمكن أن تتداخل مع عقدة أخرى حتى لو كانت خارج نطاق اتصالها، لذا فإن تصميم خوارزميات تستغل آليات التحكم في الطبولوجيا وتأخذ في الحسبان التداخل أمر ضروري لتحسين أداء الشبكة، فهو من جهة يقلل من تصادم وإعادة إرسال حزم (عند فقد حزم البيانات نتيجة التداخل) والذي - بشكل غير مباشر - يقلل من استهلاك الطاقة ويزيد من فترة عمل الشبكة. في هذه الأطروحة، أقترح خوارزمية جديدة لبناء طبولوجيا تأخذ في عين الاعتبار التداخل وتعتمد على انشاء مجموعة مهيمنة مرتبطة من العقد وهي خوارزمية IACDS: خوارزمية بسيطة، موزعة، تراعي التداخل، وذات كفاءة في استخدام الطاقة، فهي تجد مجموعة مهيمنة مرتبطة من العقد وذلك بإيقاف العقد التي لا لزوم لها مع الحفاظ على الشبكة مرتبطة، وتوفير تغطية اتصال كاملة لجميع النقاط مع الحد الأدنى من التداخل. خوارزمية IACDS تستخدم مقياس يعتمد على (المسافة والطاقة والتداخل) والذي يسمح بالمفاضلة بين أطوال فروع الطوبولوجيا (المسافة) وبين قوة ومتانة الطوبولوجيا (الطاقة والتداخل).

كلمات مفتاحية- التداخل، الشبكات المتنقلة العشوائية، التحكم في الطوبولوجيا، المجموعة المهيمنة المرتبطة، الشبكات الاستشعارية اللاسلكية، خوارزمية IACDS

Interference Reduction in Mobile Ad Hoc and Sensor Networks

By

Maaly A. S. Hassan

ABSTRACT

There are still a lot of open questions in the field of MANETs and sensor networks. If a topology incurs a large interference, either many communication signals sent by nodes will collide, or the network may experience a serious delay at delivering the data for some nodes, and even consume more energy. So, we reach to the conclusion that interference imposes a potential negative impact on the performance of wireless networks. In the last few years, researchers actively explored topology control approaches for such networks. The motivation of topology control (TC) is to maintain the connectivity of the network, reduce the node degree and thereby reduce the interference, and reduce power consumption in the sensor nodes. Some literatures have pointed out that a node can interfere with another node even if it is beyond its communication range. To improve the network performance, designing topology control algorithms with consideration of interference is imminent and necessary. Since, it leads to fewer collisions and packet retransmissions, which indirectly reduces the power consumption and extends the lifetime of the network. In this thesis, we propose a new interference-aware connected dominating set-based topology construction algorithm, namely, IACDS algorithm, a simple, distributed, interference-aware and energy-efficient topology construction mechanism that finds a sub-optimal Connected Dominating Set (CDS) to turn unnecessary nodes off while keeping the network connected and providing complete communication coverage with minimum interference. IACDS algorithm utilizes a weighted (distance-energy-interference)-based metric that permits the network operator to trade off the lengths of the branches (distance) for the robustness and durability of the topology (energy and interference).

Keywords- Interference, MANETs, Topology control, Connected Dominating Set (CDS), WSN, IACDS algorithm

CHAPTER 1

Introduction

1.1 Mobile Ad hoc NETworks (MANETs)

1.1.1 Background

A Mobile Ad hoc NETwork (MANET) is a temporary self-organizing multi-hop system of wireless mobile nodes which rely on each other to keep the network connected without the help of any preexisting infrastructure, pre-defined topology, or central administrator. These networks are generally formed in environments where it is difficult to find or settle down a network infrastructure [1]. In this type of networks, nodes must collaborate and organize themselves to offer both basic network services as routing and management services as security.

1.1.2 Importance and Applications of MANETs

There are many applications of MANETs. As a matter of fact, any day-to-day application such as electronic mail and file transfer can be considered to be easily deployable within an ad hoc network environment. Web services are also possible in case any node in the network can serve as a gateway to the outside world. This type of networks has been used in several applications where such network infrastructure is unavailable: space exploration, undersea operations, environmental monitoring, unreliable communication in battlefield, emergency rescue operations [2], industrial, commercial, cultural, sensor networks, communicating vehicles [3], etc. A wide range of possible military applications of ad hoc networks is not needed to be emphasized. Not to mention, the technology was initially developed keeping in mind the military applications, such as battlefield in an unknown territory where an infrastructure network is almost impossible to establish or maintain. In such situations, the ad hoc networks having self-organizing capability can be effectively used where other technologies either fail or cannot be deployed effectively. Advanced features of wireless mobile systems, including data rates compatible with multimedia applications, global roaming capability, and coordination with other network structures, are enabling new applications. Some well-known ad hoc network applications are:

Collaborative Work: For some business scenarios, the need for collaborative computing might be more important outside office environments than inside a building. After all, it is often the case where people do need to have outside meetings to cooperate and exchange information on a given project.

Crisis-Management Applications: These arise, for example, as a result of natural disasters where the entire communications infrastructure is in disarray (for example, Tsunamis, hurricanes, etc.). Restoring communication quickly is essential. By using ad hoc networks, an infrastructure could be set up in hours instead of day/weeks required for wire-line communications.

Personal Area Networking: A personal area network (PAN) is a short-range, localized network where nodes are usually associated with a given person. These nodes could be attached to someone's cell phone, pulse watch, belt, and so on. In these scenarios, mobility is only a major consideration when interaction among several PANs is necessary. Bluetooth is an example of a technology aimed at, among other things, supporting PANs by eliminating the need of wires between devices such as printers, cell phones, PDAs, laptop computers, headsets, and so on.

1.1.3 Challenges in MANETs

MANET is characterized by limited battery power, limited bandwidth, frequent network topology changes, and rapid mobility. Frequent topology changes result when nodes move or fail or when devices are turned on or off. These characteristics make the design of management solutions and routing protocols a great challenge [4]. A node in ad hoc networks can communicate directly with other nodes located within its radio transmission range. To communicate with a node outside its communication range a sequence of intermediate nodes in ad hoc networks are required to relay messages on behalf of this node, resulting in a multi-hop wireless network. The mobility of nodes in ad hoc networks causes the nodes to be in and out of range from one another; therefore, the connectivity in MANET varies dynamically with time. This dynamic connectivity imposes major challenges for the network layer to determine the multi-hop route between a given pair of source and destination nodes.

A wireless network is more than simply a wired network without the wires. The introduction of the wireless medium comes with its own set of challenges, which are magnified by mobility and multi-hop communication in the case of MANET; see Figure 1.1(originally appeared in [5]). This subsection lists some of the major challenges which exist in MANETs.

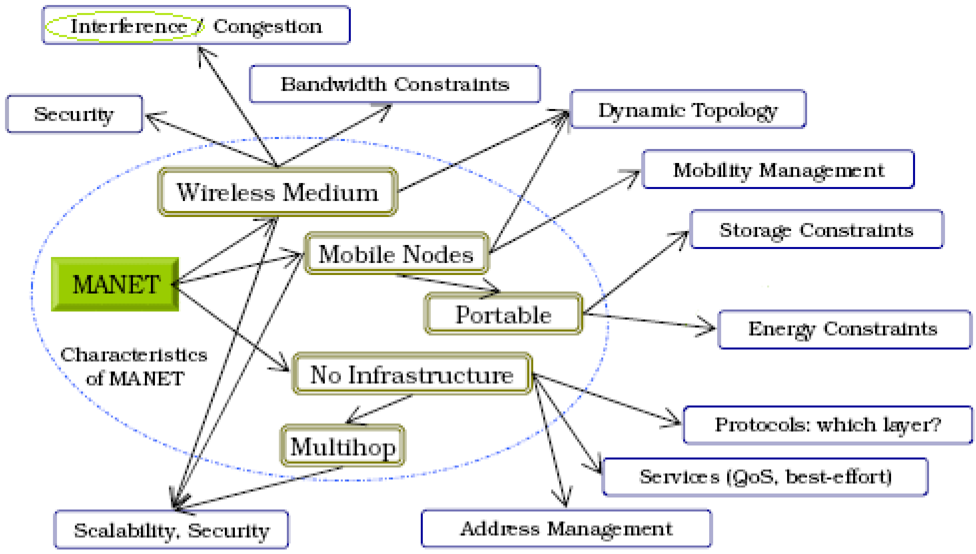


Figure 1.1: Challenges in MANETs.

Dynamic Nature: The nodes in MANETs are mobile, and dynamically appear or disappear, which leads to a highly dynamic topology, with a high probability of link breakage and network partitions. Even when the nodes are not mobile, the wireless nature of communication changes the status of the wireless links rapidly and unpredictably, which also leads to rapid changes in the topology.

Energy Constraints: As a consequence of mobility, nodes in MANETs are generally portable, and hence are low powered, with limited and exhaustible energy resources like batteries. This is exacerbated by the fact that the nodes spend extra energy while handling traffic for other nodes as well as while just listening for packets. Energy conservation is thus critical in such networks.

Interference and Congestion: Due to the shared medium, the packet is sent as radio waves which cause interference in the area surrounding the sender. Interference also exists from other wireless devices like microwaves and cordless phones, which may be sharing the wireless medium. Thus, compared to wired networks, there is an increased possibility of packet losses due to collision and congestion in MANETs [5].

Bandwidth: MANETs is also limited to a lower maximum available bandwidth (11Mbps in 802.11b, and 54Mbps in 802.11a/g) as compared to wired networks. Even this maximum is only theoretical, since the shared medium and the resulting interference limit the available bandwidth in real networks (for example, maximum of 4-5Mbps for 802.11b in reality [6, 7]).

Others: The portability of mobile nodes also means limited storage resources, while all the characteristics of MANETs described above also make providing security as well as scalability of such networks quite challenging.

1.2 Wireless Sensor Networks (WSNs)

1.2.1 Background

A wireless sensor network (WSNs) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, or pollutants, at different locations [8, 9].

Wireless sensor networks (WSNs) are a particular type of ad hoc networks, in which the nodes are sensors equipped with wireless transmission capability. Hence, they have the characteristics, requirements, and limitations of an ad hoc network [10]. The term ad hoc network describes a type of wireless network without a fixed infrastructure. Conventional wireless networks including WiFi and cellular networks have supporting backbones and are hierarchical. Nodes communicate with each other via the base stations. In an ad hoc network the nodes can communicate with each other directly via multi-hops paths. Usually the network does not have any coordinating node and hence, ad hoc networks are decentralized, self-organized, and self-healing. Messages may be duplicated on the way to the base station to provide extra resilience [11].

A WSN is usually composed of a large number of sensing nodes in the order of tens, hundreds, or even thousands scattered in a sensor field and one or a few base stations/ sinks, which connect the sensor networks to the users via the Internet or other networks. Sensor nodes are equipped with sensing, data processing, and communicating components to accomplish their tasks. Each of the sensor nodes is capable of collecting data and routing the data back to the sink by multi-hopping, as illustrated in Figure 1.2 (originally appeared in [10]).

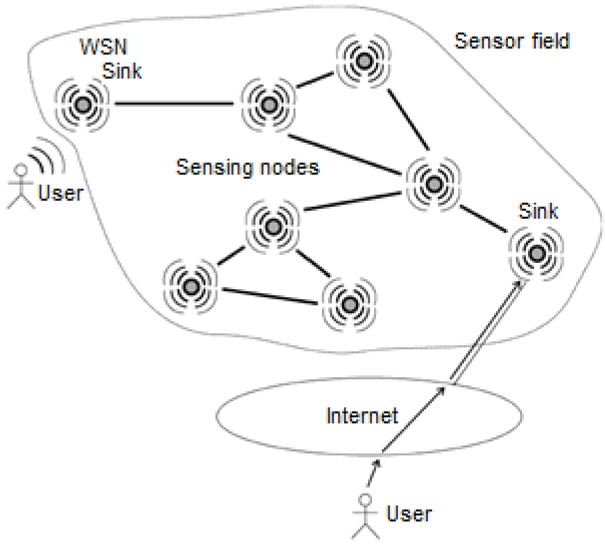


Figure 1.2: Basic structure of a wireless sensor network.

In addition to one or more sensors, as shown in Figure1.3, each node in a sensor network is typically equipped with a radio transceiver or other wireless communication devices, a small microcontroller, memory, an input/output interface that allows the integration of external sensors into the wireless device, and an energy source, usually a battery. The envisaged size of a single sensor node can vary from shoebox-sized nodes down to devices the size of grain of dust. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few cents, depending on the size of the sensor network and the required complexity of individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed, and bandwidth [10].

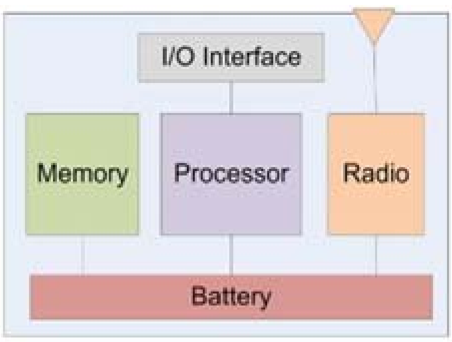


Figure 1.3: General architecture of a wireless sensor device.

1.2.2 Importance and Applications of WSNs

The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance [12, 13]. However, wireless sensor networks are now used in many industrial and civilian application areas including industrial process monitoring and control [14], machine health monitoring, environment and habitat monitoring [15, 16], healthcare applications, home automation, and traffic control [10-11, 17]. The applications for WSNs are many and varied, but typically involve some kind of monitoring, tracking, and controlling. Specific applications for WSNs include habitat monitoring, object tracking, nuclear reactor control, fire detection, and traffic monitoring. In a typical application, a WSN is scattered in a region where it is meant to collect data through its sensor nodes.

1.2.3 Challenges in WSNs

WSNs are characterized by the limited power they can harvest or store, ability to withstand harsh environmental conditions, ability to cope with node failures, mobility of nodes, dynamic network topology, communication failures, heterogeneity of nodes, large scale of deployment, and unattended operation. In general, WSNs share commonalities with existing wireless ad hoc networks which use multi-hop communication (several nodes may forward data packets to the base station) without centralized coordination. WSNs present a series of serious challenges that still need considerable research effort. While some of these challenges are a direct consequence of the constrained availability of resources in the wireless sensor nodes, and therefore very specific to WSNs, others are common challenges faced by most networking technologies. The following list briefly explains the most important challenges faced by WSNs today.

Network lifetime: WSNs are battery powered, therefore, the network lifetime depends on how wisely energy is used. In large scale wireless sensor networks or in dangerous applications (e.g., fire rescue applications, safety control systems) it is important to minimize the number of times batteries need to be changed. It is desirable to have network lifetimes in the order of one or more years. In order to achieve such long network lifetimes it is imperative to operate the sensors in a very low duty cycle.

Scalability: Some applications require hundreds or even thousands of wireless sensor devices. These large-scale WSNs present new challenges not seen in small-scale ones. Algorithms and protocols that work fine in small-scale networks do not work necessarily well in large-scale ones. One typical example is the routing function. Small-scale networks can easily run well-known proactive or reactive routing protocols using Dijkstra's shortest path algorithm. However, this approach will not be energy-efficient for large-scale WSNs. Location-based routing mechanisms using local information are better suited instead. Similar scalability problems arise in other areas.

Interconnectivity: WSNs need to be interconnected so that data reaches the desired destination for storage, analysis, and possible action. WSNs are envisioned to be interconnected with many different networking technologies. However, this is easier said than done. New protocols and mechanisms need to be designed to achieve these interconnections and allow the transfer of data to and from WSNs. Normally these interconnections are being handled by the use of gateway devices, such as the sinks, which require new capabilities for the appropriate discovery of networks and the translation of different communication protocols.

Reliability: Wireless sensor devices are cheap devices with fairly high failure rates. Further, in many applications, these devices have to be thrown to the area of interest from a helicopter, or similar vehicle. As result, several nodes break or partially break affecting their normal functionality. Node reliability is also affected by crucial levels of available energy.

Heterogeneity: New WSNs are embedding wireless sensor devices with different capabilities and functionalities that require new algorithms and communication protocols. For example, cluster-based architectures may utilize more powerful devices to aggregate data and transmit information on behalf of resource constraint nodes. This heterogeneity includes the need of clustering and data aggregation algorithms that are not of trivial design.

Privacy and security: Privacy and security are normal concerns in networking, and WSNs are not the exemption. However, security mechanisms are usually very resource demanding, which is not always in line with the resources availability.

Energy: Since the motto of sensor network is to develop tiny sensor nodes cheap enough to dispose without recharging battery, the energy conservation is a critical issue in WSNs. For instance, a sensor node, Mote, has a total stored energy on the order of 1J [18]. For wireless integrated network sensors (WINS) [20], the total average system supply currents must be less than 30 mA to provide long operating life. Moreover, inaccessibility of sensor nodes after deployment makes energy consumption more critical in wireless sensor networks.

Scalability: To cover areas with sensors which have short transmission distances due to frugal energy budget, WSNs have to scale to much large numbers (more than 10,000) of sensor nodes. Without any centralized coordination, scalability of WSNs makes itself different than wireless ad hoc networks which have up to a few thousands of nodes.

Redundancy: Due to the frequent node failures and inaccessibility of failed nodes, WSNs are required to have high redundancy. Therefore, sensor nodes are normally deployed with a high degree of connectivity instead of minimal connectivity. Because of the high degree of redundancy, the failure of single node can be negligible. At the same time, the redundancy has negative effects on the performance of WSNs because it causes redundant transmission causing broadcast storm problem [20].

In-network Processing: In general, previous transport protocols used in wired and wireless networks have assumed the end-to-end approach guaranteeing that data from senders should not be modified by intermediate nodes until data reach a receiver. However, data at WSNs can be modified or reduced into smaller amount of data by intermediate nodes in order to remove redundancy of information inside data. Therefore, previous solutions cannot accommodate this new concept of in-network processing, called data aggregation or diffusion in WSNs.

Data Centric Processing: WSNs have a large scale in terms of number of nodes which cannot be assigned individually with unique identification, e.g., IP address. Therefore, sensor nodes cannot be accessed by unique ID. Instead of addressing nodes with ID, it is more natural to access the data directly through content, attribute, e.g., location of node. The naming schemes in WSNs are often data-oriented. For example, an environmental monitoring system requests the temperature readings through a query, such as "collect temperature readings in the region bound by the rectangle (x1,y1,x2,y2)", instead of a query such as "collect temperature readings from a set of nodes of which addresses are x, y, and z."

1.3 Interference

1.3.1 Interference in MANETs and Sensor Networks

One of the main challenges of wireless communication is interference. Unfortunately, research in this area is so young that researchers have to investigate different ideas regarding the identification of a universal measure of network interference. According to the Glossary of Telecommunication Terms - Federal Standard 1037C, interference is defined as:

Interference: A coherent emission having a relatively narrow spectral content, e.g., a radio emission from another transmitter at approximately the same frequency, or having a harmonic frequency approximately the same as another emission of interest to a given recipient, and which impedes reception of the desired signal by the intended recipient.

Informally speaking, a node ***u*** may interfere with another node ***v*** if ***u***'s interference range unintentionally covers ***v***. Consequently, the amount of interference experienced by a node ***v*** corresponds to the amount of interference produced by nodes whose transmission range covers ***v***.

1.3.2 Interference Reduction in MANETs and Sensor Networks

In frequency division multiplexing cellular networks, reducing the amount of interference results in fewer channels, which in turn, can be exploited to increase the bandwidth per frequency channel. In systems using code division multiplexing, small interference helps in reducing coding overhead. In the context of ad hoc and sensor networks, there is an additional motivation for keeping interference low. In these networks consisting of battery driven devices, energy is typically scarce and the frugal usage of it is critical in order to prolong system operability and network lifetime. In addition to enhancing throughput, minimizing interference may help in lowering node energy dissipation by reducing the number of collisions (or amount of energy spent in an effort of avoiding them) and consequently retransmissions on the media access layer.

Interference can be reduced by having nodes send with less transmission power. The area covered by the smaller transmission range will contain fewer nodes, yielding less interference. On the other hand, reducing the transmission range has the consequence of communication links being dropped. However, there is surely a limit to how much the transmission power can be decreased. In ad hoc networks, if the node's transmission ranges become too small and too many links are abandoned, the network may become disconnected. Hence, transmission ranges must be assigned to nodes in such a way that the desired global network properties are maintained.

1.4 Topology Control

1.4.1 Definition of Topology Control

Topology Control (TC) is one of the most important techniques used in wireless ad hoc and sensor networks to reduce energy consumption (which is essential to extend the network operational time) and radio interference (with a positive effect on the network traffic carrying capacity). The goal of this technique is to control the topology of the graph representing the communication links between network nodes with the purpose of maintaining some global graph property (e.g., connectivity), while reducing energy consumption and/or interference that are strictly related to the nodes' transmitting range. An informal definition of topology control is the art of coordinating nodes, decisions regarding their transmitting ranges, in order to generate a network with the desired properties. *Interference-efficient topology control* is to find a sub-graph *H* from the original graph *G*, representing a network, to minimize interference while preserving fixed properties (connectivity and low power consumption). *Topology control* is a system-level perspective to optimize the choice of the nodes' transmit power levels to achieve a certain global property while *power control* is a wireless channel perspective to optimize the choice of the transmit power level for a single wireless transmission, possibly along several hops.

1.4.2 Motivation of Topology Control

Topology control techniques have the potential to mitigate two important problems occurring in wireless ad hoc networks: node energy consumption and radio interference. Another major requirement of topology control in MANETs and sensor networks is to maintain connectivity in the network. Once the connectivity is ensured, the second goal is usually to reduce the radio transmission power of individual nodes for two reasons. The first is to reduce the power used for transmitting packets. The second one is to reduce the node degree in the neighborhood. A sparse network is desirable because it can enhance the performance of the MAC protocols. If a CSMA type scheme is used, low network degree means less probability of collisions. If a TDMA scheme is used, slot assignment is easier with fewer nodes and there is less chance of congestion. Moreover, routing is simpler in a sparse network than a dense network because there are fewer routes to consider.

1.4.3 Topology Construction and Maintenance

Topology control has been divided into two sub problems: *topology construction*, which concerns the initial reduction, and *topology maintenance*, which concerns the maintenance of the reduced topology so that characteristics like connectivity and coverage are preserved.

Once the initial topology is deployed, especially when the location of the nodes is random, the administrator has no control over the design of the network; for example, some areas may be very dense, showing a high number of redundant nodes, which will increase the number of message collisions and will provide several copies of the same information from similarly located nodes. However, the administrator has control over some parameters of the network: transmission power of the nodes, state of the nodes (active or sleeping), role of the nodes (cluster-head, gateway, regular), etc. By modifying these parameters, the topology of the network can change.

Upon the same time a topology is reduced and the network starts serving its purpose, the selected nodes start spending energy: The "optimal" reduced topology stops being at the first second of full activity. After some time being active, some nodes will start to run out of energy. Especially in wireless sensor networks with multi-hoping, it is a fact that nodes that are closer to the sink spend higher amounts of energy than those farther away due to packet forwarding. The network must restore the reduced network periodically in order to preserve connectivity, coverage, density, and any other metric that the application requires.

A. Topology Construction Algorithms

There are many ways to perform topology construction: by changing the transmission range of the nodes, turning off nodes from the network, creating a communication backbone, and clustering. Figure 1.4 shows the classification of topology control algorithms [21]. The hierarchical and non-hierarchical algorithms form the upper categories of topology control algorithms.

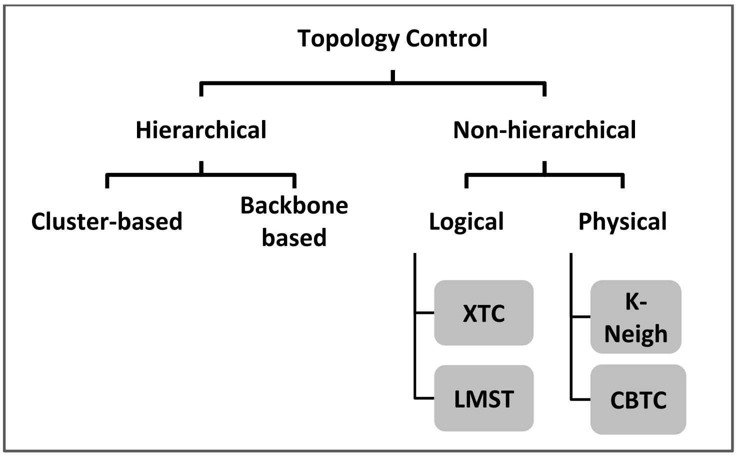


Figure 1.4: Classification and examples of topology control algorithms.

Some hierarchical topology control algorithms are based on the construction of a backbone such as connected dominant set (CDS) for efficient communication. Other hierarchical algorithms divide the whole network into communication clusters for management and energy conservation [22]. The non-hierarchical category is classified into physical or logical algorithms based on their optimization objective. In general, physical topology control algorithms determine the optimal transmission ranges of nodes, while logical topology control algorithms determine the optimal neighbor sets of nodes. Logical TC algorithms can be viewed as "pruning" a communication graph to remove redundant edges on the graph. The major contribution of a physical TC is to generate a reliable underlying structure for connectivity, while a logical TC focuses on a generating a sparse graph, which can simplify the process of routes finding. Physical TC algorithms solve what is called the range assignment problem.

Non-hierarchical topology construction algorithms

1) Geometry based: Gabriel graph (GG) [23], Relative neighborhood graph (RNG) [24]
2) Spanning Tree based: LMST [25]
3) Direction based: Yao and Nearest neighbor graph [26, 27], Cone Based Topology Control (CBTC) [28],
4) Neighbor based: KNeigh [29], XTC [30]

5) Routing based: COMPOW [31]

Hierarchical topology construction algorithms

1) CDS-based: EECDS [32], CDS-Rule K [33]
2) Cluster-based: Low Energy Adaptive Clustering Hierarchy (LEACH) [34], HEED [35]

B. Topology Maintenance Algorithms

In the same manner as topology construction, there are many ways to perform topology maintenance: by global algorithms, local algorithms, dynamic algorithms, static algorithms, hybrid algorithms, and algorithms triggered by time, energy, density, and randomly.

Global topology maintenance algorithms [36]

1) Dynamic Global Topology Recreation (DGTRec): periodically, wake up all inactive nodes, reset the existing reduced topology in the network and apply a topology construction protocol.

Dynamic Global Time-based Topology Recreation – DGTTRec: Every time interval the topology maintenance algorithm terminates the previous reduced topology and invokes the topology construction algorithm to create a new one.

Dynamic Global Energy-based Topology Recreation – DGETRec: Every time a node reaches a critical energy threshold, the TM algorithm terminates the previous reduced topology and invokes the topology construction algorithm to create a new one.

2) Static Global Topology Rotation (SGTRot): initially, the topology construction protocol must create more than one reduced topology (hopefully as disjoint as possible). Then, periodically, wake up all inactive nodes, and change the current active reduced topology to the next, like in a Christmas tree.

3) Hybrid Global Topology Rotation and Recreation (HGTRotRec): works as the SGTRot, but when the current active reduced topology detects a certain level of disconnection, resets the reduced topology and invokes the topology construction protocol to recreate that particular reduced topology.

Hybrid Global Time-based Topology Rotation and Recreation – HGTTRotRec: Every time interval the topology maintenance algorithm rotates the active reduced topology for one of the preplanned ones. If the new preplanned topology cannot provide the expected service (has no connection with the sink), the hybrid TM algorithm invokes the topology construction algorithm to create a new reduced topology on the fly.

Hybrid Global Energy-based Topology Rotation and Recreation – HGETRotRec: Every time a node reaches a critical energy threshold, the topology maintenance algorithm rotates the active reduced topology for one of the preplanned ones. If the new preplanned topology cannot provide the expected service (has no connection with the sink), the hybrid topology maintenance algorithm invokes the topology construction algorithm to create a new reduced topology on the fly.

Local topology maintenance algorithms [36]

1) Dynamic Local DSR-based Topology Maintenance (DL-DSR): this protocol, based on the Dynamic Source Routing (DSR) routing algorithm, recreates the paths of disconnected nodes when a node fails.

1.5 Connected Dominating Set

Definition of minimum dominating set (MDS)

In a graph, a dominating set is a subset of nodes such that for every node v in graph, either a) v is in the dominating set or b) a direct neighbor of v is in the dominating set. The minimum dominating set problem asks for a dominating set of minimum size. Its formal definition is as follows:

Instance: A graph $G = (V, E)$.

Problem: Find a subset D with minimum cardinality for G, i.e., a subset $D \subseteq V$ such that for all $u \in V - D$, there is a $d \in D$ for which $(u, d) \in E$.

In common, connected minimum dominating set problems try to find a subset covering neighbors with minimum cardinality. In network research area, MDS problem has been considered as one of popular approaches to solve various networking problems [37].

1.6 Motivation

In this thesis, we focus on the interference aspects of wireless mobile ad hoc and sensor networks. The interference imposes a potential negative impact on the performance of wireless networks. Hence interference reduction is essential for such networks. The main aim of this thesis is to investigate the performance of wireless ad hoc and sensor networks in terms of interference reduction. As we will see in Chapter 3, a device is interfered if it receives transmissions, most of which may not be intended for it. Hence it is crucial to actively account for and attempt to reduce the excess interference due to this mistaken receive. However, the existing models for interference reduction mistake nodes within the transmission range for the only hidden interfering ones. Distinctly, for a node, all active neighbors within its interference range are potential interfering sources. Our goal is to study the effect of interference reduction through TC on the performance of ad hoc and sensor networks, by introducing an explicit computational model of node interference based on its actual inducement on the physical layer. In this study, we also propose an interference-aware topology construction algorithm to solve the corresponding topology control problem: constructing a network topology with minimum node interference.

1.7 Research Overview

In this thesis, we introduce and validate a novel interference measurement model. Based on the evaluation of such model, a new topology construction algorithm is proposed, namely, IACDS algorithm, a simple, distributed, interference-aware, and energy-efficient topology construction mechanism that finds a sub-optimal Connected Dominating Set (CDS) to turn unnecessary nodes off while keeping the network connected and providing complete communication coverage with minimum interference. IACDS algorithm utilizes a weighted (distance-energy-interference)-based metric that permits the network operator to trade off the lengths of the branches (distance) for the robustness and durability of the CDS (energy and interference).

1.7.1 Approach and Methodology

For this research, we have adopted the following approach:

- Re-visitation of the problem of interference reduction in ad hoc and sensor networks.
- Development and evaluation of model for interference reduction.
- Design of interference-aware topology construction scheme, IACDS, based on the proposed model of interference reduction.
- Empirical evaluations of the proposed scheme through extensive simulations which compare the proposed algorithm to the existing ones.

- Refinement and selection of the best maintenance policy for the proposed topology construction algorithm based on thorough analysis.

1.7.2 Original Contributions

The major contributions of this thesis are summarized as follows:

1. A new interference measurement model is introduced with the following properties: it creates a relationship between all local parts of the network, and takes into account the maximum interference of the network. The proposed interference metric, Equation 3.9, is embedded in a selection metric, Equation 3.10, which produces a value between 0 and 1 that is assigned to each neighbor in the process of selecting the new nodes in the CDS; the higher the value of the metric, the higher the priority. As it can be seen from Equation 3.10, the selection metric gives priority to those nodes with minimum interference, higher energy, and which are farther away from the parent node. The final effect of this choice is to have a CDS with minimum interference, fewer active nodes, and better coverage. However, proper weight manipulation can satisfy different criteria, as needed. As the main goal of the proposed algorithm is to reduce the global interference of the network, the interference metric is weighted to its maximum value, 1. If communication coverage is to be optimized and the average height of the node in the CDS (number of hops) needs to be reduced, the distance metric must be weighted more heavily. The downside is that low energy nodes may be included in the CDS, which may introduce early failures of nodes, and therefore reduce the lifetime of the network. On the other hand, if reliability of the tree is desired, energy must be weighted more. The model is validated since the evaluation gives a realistic and accurate calculation of the total interference at a node.

2. Based on the proposed interference measurement model, an aspect of interference-awareness for performance efficiency in ad hoc and sensor networks is explored, specifically through the design and analysis of an interference-aware topology construction algorithm, namely, IACDS, a simple, distributed, interference-aware and energy-efficient topology construction mechanism that finds a sub-optimal Connected Dominating Set (CDS) to turn

unnecessary nodes off while keeping the network connected and providing complete communication coverage with minimum interference. IACDS algorithm utilizes a weighted (distance-energy-interference)-based metric that permits the network operator to trade off the lengths of the branches (distance) for the robustness and durability of the CDS (energy and interference).

3. The best maintenance policy for the proposed topology construction algorithm is determined through an extensive sensitivity analysis performed on a topology maintenance scheme.

The evaluation of the proposed interference reduction scheme has been performed with the help of simulations, using various network parameters which are as close to the realistic scenarios as possible. In addition to the total interference, a more complete evaluation is presented, by evaluating the overall network lifetime, total coverage area, spent energy ratio, as well as the number of sent messages.

1.7.3 Thesis Organization

The rest of the thesis is organized into five chapters, followed by the appendices.

Chapter 2 gives a background for interference reduction through topology control in MANET and sensor networks. Related works in terms of existing topology control techniques are discussed in Section 2.1 while existing interference-aware topology control protocols are discussed in Section 2.2.

Chapter 3 introduces an analytical model for measuring the interference in the network. A new interference-aware topology construction algorithm is proposed in Section 3.2.

Chapter 4 evaluates the performance of the proposed interference-aware topology construction algorithm, through a detailed empirical analysis and comparisons with other existing algorithms.

Finally, Chapter 5 concludes the thesis with a summary of the major achievements of the study and outlines the future works.

CHAPTER 2

Related Works

2.1 Topology Control

Topology construction can be exercised by reducing the transmission range of all nodes by the same minimum amount, or the minimum transmission range for each node [38]. Other techniques are based on the assumption that nodes have information about their own positions and the position of their neighbors [39], or that they have directional antennas that are used to determine the orientation of the nodes [40, 41]. Although both assumptions are valid, they are costly and not easy to implement. Other topology control methods, such as the one considered in this thesis, are based on the Connected Dominating Set (CDS) paradigm. Here, the idea is not to change the transmission range of the nodes but to turn unnecessary nodes off while preserving important network properties, such as connectivity and communication coverage.

The CDS approach has been utilized in several papers [40, 42-48]. Most CDS-based mechanisms work in two phases: In phase one, they create a preliminary version of the CDS, and in phase two they add or remove nodes from it to obtain a better approximation to the optimal CDS. Two relevant CDS-based mechanisms are the Energy Efficient CDS (EECDS) [47] and the CDS-Rule-K [44] algorithms.

The EECDS algorithm builds a CDS tree creating Maximal Independent Sets (MIS), which are clusters with non-connected clusterheads, and then selects gateway nodes to connect the clusterheads of the independent sets. The EECDS algorithm proceeds in two phases. The first phase begins with an initiator node that elects itself as a clusterhead and announces it to its neighborhood. This set of nodes is now "covered". The now "covered" nodes will pass the message to its uncovered neighbors, 2-hop away from the initiator, which start competing to become clusterheads. Once there is a new clusterhead, the process repeats with the 4-hop away nodes from the initiator, until there are no more uncovered nodes. On the second phase the covered non-clusterhead nodes compete to become gateways between the clusterheads.

The CDS-Rule-K ` utilizes the marking algorithm proposed in [47] and the pruning rule included in [46]. The idea is to start from a big set of nodes that accomplishes a minimum criterion and prune it according to a specific rule. In the first phase, the nodes will exchange their neighbor lists. A node will remain active if there is at least one pair of unconnected neighbors. In the second phase, a node decides to unmark itself if it determines that all its neighbors are covered by marked nodes with higher priority, which is given by the level of the node in the tree: lower level, higher priority. The final tree is a pruned version of the initial one with all redundant nodes with higher or equal priority removed.

Cone Based Topology Control (CBTC) [28] is based on an angular separation parameter α, and distance estimate between the nodes and the neighbor. The fundamental concept of CBTC is that a node u tries to find the minimum power pu,α such that transmitting with pu,α ensures that in every cone of degree α around u, there is some node that u can reach with power pu,α. It was proven in [28] that if $\alpha \leq 5\pi/6$ then connectivity is preserved. CBTC also has an optimization stage (logical TC) to identify energy inefficient stages. CBTC guarantees connectivity provided that the network is connected when all nodes are transmitting with their maximum power. However, the major problem of implementing CBTC is the requirement for directional information, which may not be available in common sensor nodes.

XTC [30] is a logical topology control algorithm that aims at generating a graph optimized for routing. It does not require node positions of an ad hoc network. Rather, it depends on the "link quality" of neighbors, which can be signal attenuation, Euclidean distance, or packet arrival rate to evaluate the quality of a neighbor connection. The operation of XTC starts with neighbors ordering by their link quality. The ordering is then broadcasted to all the neighbors so all nodes will have a copy of this information about their neighbors. Then, every node chooses their edges according to this local information. For each node, XTC maintains a neighbor list in which XTC eliminates the farthest neighbors that are reachable by the relays of closer neighbors, hence reduces power levels to that node while maintaining the same connectivity. XTC is a simple algorithm that preserves the connectivity of the original graph and does not require special hardware. More importantly, it does not assume the network graph to be a unit disk graph, which is the standard assumption of most distributed algorithms (such as CBTC and LMST) but is not always true in reality.

The k-Neighbors protocol [29] is based on the control parameter, node degree *k*, with the additional distance information estimated by radio signal strength or time of arrival. k-Neighbors is a physical TC algorithm for generating networks connected with high probability. The basic algorithm requires initially that every node broadcasts its ID at maximum power. Upon receiving broadcast messages from other nodes, every node keeps track of its neighbors and estimates the distance associated with them. The nodes then compute its k-closest neighbors and these become their k-Neighbors list. The nodes exchange their neighbor lists at maximum power and hence, each node would know the symmetric neighbors in the neighborhood. Unsymmetrical neighbors are deleted. k-Neighbors is a simple algorithm that does not require special hardware but does not guarantee connectivity. It is also degree-bounded by k. The number of messages exchanged in each update is exactly $2n$, where n is the number of nodes.

Local Minimum Spanning Tree (LMST) [25] is another logical topology control algorithm that chooses energy efficient edges in the final communication topology. The concept is similar to finding the minimum spanning tree (MST) for a graph, except that the trees are constructed locally using direct neighbors within a node's maximum transmission range, R. In LMST, each node collects one hop neighbors and builds a local MST. Then, through negotiations each node selects edges only from its local MST to guarantee the network connectivity. It was found that a topology constructed using LMST has a maximum degree of six and network connectivity is preserved [25]. However, its major disadvantage is the requirement for location information. Although the author proposes that the location requirement can be substituted by nodes estimating the distance to all the visible nodes and then exchanging the list, this solution involves a lot more overheads and is less scalable.

Cluster based Topology Control (CLTC) [49] is a framework that collects multi-hop neighbors' information in a distributed setting and take advantage of centralized algorithms applied to the collected information for each node so as to make the network connected or 2-connected. The CLTC framework consists of three phases:

- Phase 1 - Form clusters by any distributed clustering algorithm, where nodes have knowledge only about their restricted-hop neighbors. Note that operations in phases 2 and 3 are independent of the specific clustering algorithm used in this first phase.
- Phase 2 - Utilize an appropriate centralized algorithm to calculate the power assignments for all members of each cluster such that the resulting cluster topology satisfies the given connectivity constraint (i.e. connected or 2-connected).
- Phase 3 - Adjust the power levels of nodes on the borders of clusters so as to provide appropriate connectivity with adjacent clusters. Hence, the network as a whole is connected or 2-connected.

From this framework, it is clear that the performance of the CLTC framework is dependent on the specific clustering algorithm that is used.

NTC is a Delaunay Triangulation (DT)-based topology control algorithm. It builds a DT graph as the starting topology and initiates a negotiation among neighbors to meet the node degree requirement. The resulting topology is degree-bounded and is targeted to make each node have roughly equal degree and so that the distances to all its neighbors are similar.

Low-Energy Adaptive Clustering Hierarchy (LEACH), a communication protocol for microsensor networks [34]. LEACH collects data from distributed microsensors and transmits it to a base station. LEACH uses the following clustering-model: Some of the nodes elect themselves as cluster-heads. These cluster-heads collect sensor data from other nodes in the vicinity and transfer the aggregated data to the base station. Since data transfers to the base station dissipate much energy, the nodes take turns with the transmission – the cluster-heads "rotate". This rotation of cluster-heads leads to a balanced energy consumption of all nodes and hence to a longer lifetime of the network. A modification of LEACH's clusterhead selection algorithm was proposed to reduce energy consumption. The energy needed for the transmission of one bit of data from node *u* to node *v*, is the same as to transmit one bit from *v* to *u* (symmetric propagation channel). Cluster-heads collect *n* k-bit messages from *n* adjacent nodes and compress the data to *cn* k-bit messages which are transmitted to the BS, with $c \leq 1$ as the compression coefficient. The operation of LEACH is divided into rounds. Each of these rounds consists of a set-up and a steady-state phase. During the set-up phase cluster-heads are determined and the clusters are organized. During the steady-state phase data transfers to the base station occur.

Hybrid Energy-Efficient Distributed clustering (HEED) [35], an energy-efficient distributed clustering approach for ad-hoc sensor networks which has four primary goals: (i) prolonging network lifetime by distributing energy consumption, (ii) terminating the clustering process within a constant number of iterations/steps, (iii) minimizing control overhead (to be linear in the number of nodes), and (iv) producing well-distributed cluster heads and compact clusters. HEED periodically selects cluster heads according to a hybrid of their residual energy and a secondary parameter, such as node proximity to its neighbors or node degree. HEED does not make any assumptions about the distribution or density of nodes, or about node capabilities, e.g., location-awareness. The clustering process terminates in O(1) iterations, and does not depend on the network topology or size. The protocol prolongs network operation interval, incurs low overhead in terms of processing cycles and messages exchanged, can be easily tuned to optimize resource usage according to the network density and application requirements. HEED operates in quasi-stationary networks where nodes are location-unaware and have equal significance. HEED can also be useful in multi-hop networks if the necessary conditions for connectivity (the relation between cluster range and transmission range under a specified density model) hold. It also achieves fairly uniform cluster head distribution across the network. A careful selection of the secondary clustering parameter can balance load among cluster heads.

2.2 Interference Reduction via Topology Control

In this section, related works in the field of topology control are discussed with special focus on the issue of interference. Interference reduction is one of the main motivations of topology control besides direct energy conservation by restriction of transmission power. Astonishingly however, all the above topology control algorithms at the most implicitly try to reduce interference. Where interference is mentioned as an issue at all, it is maintained to be confined at a low level as a consequence to sparseness or low degree of the resulting topology graph.

However, M. Burkhart, et al, in [50] reveal that such an implicit notion of interference is not sufficient to reduce interference since message transmission can affect nodes even if they are not direct neighbors of the sending node in the resulting topology. Besides demonstrating the weakness of modeling interference implicitly, [50] introduces an explicit definition for interference in wireless networks. [50] presents a traffic-independent model and defines the interference of a link e = (u, v) as the cardinality of the set of nodes covered by two disks centers at u and v with radius $||uv||$, denoted as *coverage set* of link e, $\text{cov}(e)$. This model, named as *link-interference via coverage*, is chosen from the assumption that whenever a link (u, v) is used for a send-receive transaction all nodes whose distance to node u or node v is less than $||uv||$ will be affected in some way.

K. Moaveni and X. Li, in [51], extend this work and propose *node-interference via coverage* model. The interference of a node u is defined as the maximum coverage set of links incident on u. However, coverage model is based on the question how many other nodes can be disturbed by a given communication node or link. The definition of interference suggested in [51] is problematic in two respects. First, it is based on the number of nodes affected by communication over a given link. In other words, interference is considered to be an issue at the sender instead of at the receiver, where message collisions actually prevent proper reception. It can therefore be argued that such sender-centric perspective hardly reflects real-world interference. The second weakness of the model introduced in [51] is of more technical nature. According to its definition of interference, adding (or removing) a single node to a given network can dramatically influence the interference measure. Addition of one node to a cluster of roughly homogeneously distributed nodes entails the construction of a communication link covering all nodes in the network, accordingly - merely by introduction of one additional node - the interference value of resulting topology is pushed up from a small constant to the maximum possible value, that is the number of nodes in the network. This behavior contrasts to the intuition that a single additional node also represents one additional packet source potentially causing collisions. Moreover, neglect of the case that a particular node might be influenced by multiple communication links with small coverage set might lead discontented results of the proposed algorithms in [51].

An attempt to correct for this deficiency is made by P. Von Richenband, et al, in [52], where an alternative, receiver-centric, interference model is introduced. In this model, node u will be interfered by v whose distance to v is less than Rv, its distance to reach the farthest neighbor, or $\{v \mid ||uv|| \leq Rv\}$ formally. It is denoted as *node-interference via transmission* model. Under the assumption that only symmetric edges are considered, it can be proved that that nodes set, mentioned above, is equivalent to $\{v \mid ||uv|| \leq Ru\}$. Unfortunately, one fatal drawback is that previous works consider the interference range equals to the transmission range. According to the theoretical analysis of actual cause of interference in reference [53], by K. Xu, et al, interference range generally differs from transmission range and hidden terminals located within the $1.78d$ distance (d denotes the communication distance) of the receiver are also disturbing sources, which is neglected in previous works at all times. Researches mistake nodes within the transmission range for the only hidden interfering nodes.

Authors of [54] introduce an explicit definition of interference between edges and establish – based on a time-step routing model – a trade-offs between the concepts of congestion, energy consumption, and dilation. This interference definition is based on the current network traffic. In [54], more attention is also being paid to the fact that if nodes are capable of adapting their transmission power – an assumption already made in early work that can be considered originators of topology control considerations [55, 56] – interference ranges correlate with the length of communication links. More precisely the interference range of a link depends on the transmission power levels chosen by the two nodes communicating over the respective link. While [54] defines interference based on current network traffic, [50] introduces a traffic-independent notion of interference. Moreover, the latter work shows that the above statement that graph sparseness or small degree implies low interference is misleading. The interference model described in [50] builds on the question of how many nodes are affected by communication over a given link. This sender-centric perspective can however be accused to be somewhat artificial and to poorly represent reality, interference occurring at the intended *receiver* of a message. Furthermore, this interference measure is susceptible to drastic changes even if single nodes are added to or removed from a network.

CHAPTER 3

Interference Reduction through TC

3.1 Network and Interference Model

Interference-efficient topology control is to find a sub-graph H from the original graph G to minimize interference while preserving fixed properties.

3.1.1 Network Representation

An ad hoc network is modeled as an Euclidian graph G = (V, E) with vertices in V representing network nodes, and the edges E representing communication links. The Euclidian position of the vertices in the graph corresponds to the physical position of the nodes in the Euclidian two dimensional space, which means that the edge weight, $w(u, v)$, represents the physical distance between nodes u and v. Each node u has a maximum transmission range R_u. In order to prevent existing basic communication between neighboring nodes from becoming unacceptably cumbersome [57], only symmetric edges are considered. Since only undirected links are considered, a link uv can only exist if the Euclidian distance between the nodes u and v is no larger than min(Ru, Rv). Assume that any node can adjust its transmission power to any value from 0 to its maximum transmission power, depending on the desired transmission radius: when transmitting to node v, node u uses the lowest possible transmission power needed to reach v. A common path loss model says that the signal strength received by a node can be described as p/d^{α}, where p is the transmission power used by the sending node, d is the distance between two nodes, and α is a path loss gradient, depending on the transmission environment. Consequently, the energy cost $c(u, v)$ to send a message of fixed length directly from node u to node v is $\theta(|u, v|^{\alpha})$. The energy cost of a path is defined as the sum of the energy costs of all edges in the path.

3.1.2 Measurement of Interference

Intuitionally, a node in the network G is interfered by others, if messages are received but not intended for it [58]. From the perspective of the physical layer, a signal arriving at a receiver is assumed to be valid if the SNR (Signal to Noise Ratio) is above a certain threshold T_{SNR}. Assume a transmission to a receiver with transmitter-receiver d meters apart and at the same time, an interfering node *d* meters away from the receiver starts another transmission. According to analysis in [59], a crucial conclusion is made that interference range is $\sqrt[4]{T_{SNR}} * d$, with an approximation value of 1.78*d when T_{SINR} is set to 10 for instance. Previous researchers mistake nodes within the transmission range for the only hidden interfering ones. Distinctly, for a node, all active neighbors within its interference range are potential interfering sources. Consequently, interference amount is defined as the maximum cardinality of active interference neighbors set. Given a network N = (V, E), the interference neighbors set of a node *u* communicating with *v* in N, denoted as INS_u^v, is defined as follows:

$$INS_u^v = \{w \in V \mid w \in D(w, \sqrt[4]{T_{SNR}} * \|uv\|)\} \qquad 3.1$$

Consequently, the interference amount of the node is defined as:

$$\text{IA (receiver)} = \max \text{INS}_\text{i} \qquad 3.2$$

Where D(u, *r*) denotes the set of nodes located in the circular area centered at node *u* with radius *r*, and ||uv|| the communication distance. The receiver node of a ***H****ello* ***M****essage* computes its interference amount using the following algorithm:-

Algorithm 3.1
Purpose: Calculating the interference amount at the receiver of HM
Inputs: Hello Message HM
Outputs: Total interference amount IA(receiver)

Procedure:

1. *For (i=1 to numberOfNeighbors) {*
   a. *IR = 1.78 * d (receiver, Neighbors (i))*
   b. *INS (i) = 0*
   c. *For (j=1 to numberOfNeighbors) {*
      i. *If (d (receiver, Neighbors (j)) ≤ IR)*
      ii. *INS (i) ++*
   d. *}*
2. *}*
3. *IA (receiver) = max* INS_i

Where:
IR refers to Interference Range
d refers to the Euclidean distance
INS refers to Interference Neighbor Set
IA refers to Interference Amount

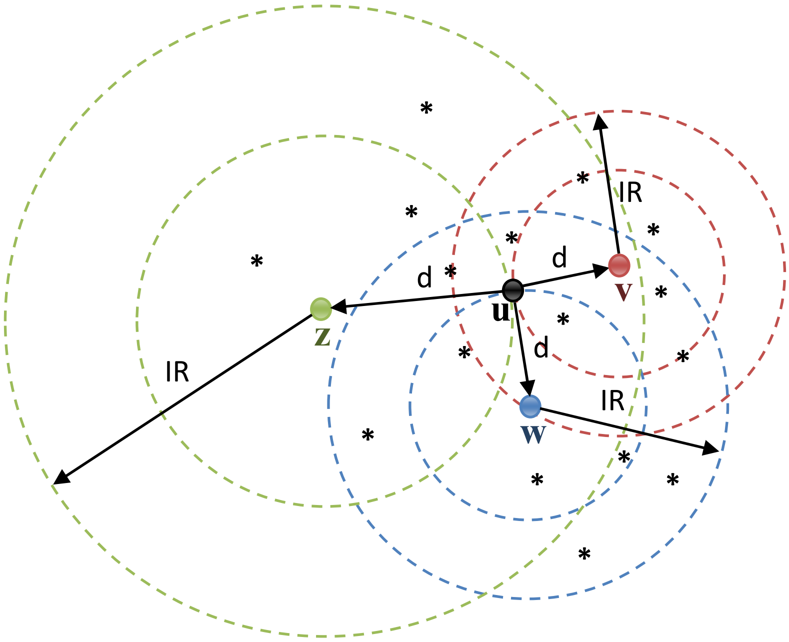


Figure 3.1: Example network to demonstrate the first interference metric.

Figure 3.1 shows an example network consisting of twenty nodes. The interference neighbor set of node *u* when communicating with node *v* is seven, while its interference neighbor set when communicating with node *w* is eleven, and when communicating with node *z* its interference neighbor set equals to ten. The maximum of its interference amounts is 11. Based on the previous definition node *u* suffers from interference, and it can be measured as follows:

$$INS_u^v = 7 \quad 3.3$$

$$INS_u^w = 11 \quad 3.4$$

$$INS_u^z = 10 \quad 3.5$$

$$I(u) = \max_{x=v,w,z} INS_u^x \quad 3.6$$

$$I(u) = INS_u^w = 11 \quad 3.7$$

The previous definition is problematic, since it works according to the principle: The global interference in a network depends solely on the local part with the highest interference. Reducing the interference in that part by definition reduces the interference of the entire network. One problem is that the metric does not consider the interference in general; a network with high interference in one place and low interference everywhere else could have the same interference as another network with equally high interference everywhere. We extend the previous work by defining an average interference neighbors set as the sum of the interference neighbors sets divided by the number of neighbors.

$$IA(receiver) = \sum_{i=1}^{|INS|} INS_i / |INS| \quad 3.8$$

Despite the previous extended metric makes a relationship between all local parts of the network, from another point of view it suffers from some weakness: it does not take into account the real distribution of the interference in the network, which means that several networks with different interference amounts in their local parts may have the same global interference. In other words, there will be local parts with higher interference than the global interference of the entire network which is not realistic, e. g. a network with high interference in one place and low interference everywhere else.

We propose to form an interference measure which functions with the following properties: creates a relationship between all local parts of the network, and takes into account the maximum interference of the network. This can be achieved by mixing the previous two metrics in one equation.

$$IA(receiver) = \sum_{i=1}^{|INS|} INS_i / |INS| * max\, INS_{i=1}^{|INS|} \quad 3.9$$

3.2 Interference-aware CDS-based Topology Construction Algorithm IACDS: *(Proposed Algorithm)*

Topology control is a well-known strategy to save energy and extend the lifetime of wireless mobile ad hoc and sensor networks. In this thesis we exploit the benefits of topology control in order to reduce interference in the entire network. So, we propose the IACDS algorithm, a simple, distributed, and energy-efficient topology construction mechanism that finds a sub-optimal Connected Dominating Set (CDS) to turn unnecessary nodes off while keeping the network connected and providing complete communication coverage with minimum interference. IACDS algorithm utilizes a weighted distance-energy-interference-based metric that permits the network operator to trade off the lengths of the branches (distance) for the robustness and durability of the tree (energy and interference).

Interference-aware connected dominating set-based topology construction algorithm: IACDS (in general)

Interference-efficient topology control is to find a sub-graph *H* from the original graph *G* to minimize interference while preserving fixed properties (connectivity and low power consumption).

Algorithm 3.2

Purpose: CDS topology such that the resulting topology is connected and with minimal interference.

Inputs: Original network G =(V, E)

Outputs: $H_{CDS}=(V_H, E_H)$

Procedure:

1. $V_H = \{sink\}$
2. *Start with the sink node: discover its neighborhood NH*
3. *For each node* **v** ∈ *NH, calculate the interference metric*
4. *Sort nodes in NH in an ascending order of the interference metric*
5. *While NH is not empty*
6. *Select* **v** ∈ *NH with minimum interference metric and outside the coverage area of other node in the neighborhood*
   - *if sink and* **v** *are not connected in* H_{CDS} *then*
     $V_H = V_H \cup \{\mathbf{v}\}$
   - *end if*
   - $NH = NH \setminus \{v\}$
7. *End while*
8. *Repeat step 2 with all* **v***'s in* V_H
9. $H_{CDS} = (V_H, E_H)$

3.2.1 IACDS TC Algorithm Details

Interference-aware CDS-based Topology Construction Algorithm

1. The *sink node* sends an initial **H**ello **M**essage
2. For each neighbor that received the **H**ello **M**essage
   1) If the node has not been covered yet
      - set its state as covered
      - adopt the sender as its parent node
      - answer back with a **P**arent **R**ecognition **M**essage

2) If the node has been already covered
   - ignore the **H**ello **M**essage

3. The parent node sets a *timeout* to receive the answers from its neighbors
4. Once this *timeout* expires,
   1) If it does not receive any **P**arent **R**ecognition **M**essage
      - Turn off
   2) Otherwise
      - Sort the list in decreasing order with respect to the *selection metric*
      - Broadcasts a **C**hildren **R**ecognition **M**essage (includes the selected list) to all its candidates
5. Once the candidate nodes receive the list,
   - Set a *timeout* period (proportional to their position in the candidate list) waiting for **S**leeping **M**essage from their brothers
6. Once the *timeout* expires,
   1) If the node receives a **S**leeping **M**essage
   - Turns itself off
   2) Otherwise
   - Send a **S**leeping **M**essage to turn its brothers off
   - Become a new parent and starts its own process of looking for candidates

   Bonus opportunity
7. Once every node receives a **S**leeping **M**essage,

Set a timer to send **H**ello **M**essage and start its own building process.

3.2.2 Algorithm Description

The IACDS algorithm assumes no prior knowledge about the position or orientation of the nodes; therefore, the nodes do not have an exact geometric view of the topology. However, nodes can determine how far a node is based on the strength of the signal received, and this information is enough to select a close-to-optimal CDS tree, based on the belief that farther nodes will offer better area of communication coverage. The IACDS algorithm is executed in 4 moments: Neighborhood Discovery, Children Selection, State Decision Based on Selection Metric, and Second Opportunity.

1) Discovering the surrounding neighborhood: The CDS building process is started by a predefined node that might be the sink, right after the nodes are deployed. The sink, node A in Figure 3.2a, starts the protocol by sending an initial *Hello Message*. This message will allow the neighbors of node A to know their parent node. In Figure 3.2a, nodes B, C, D, and E will receive the message. Node F is out of reach from node A [60].

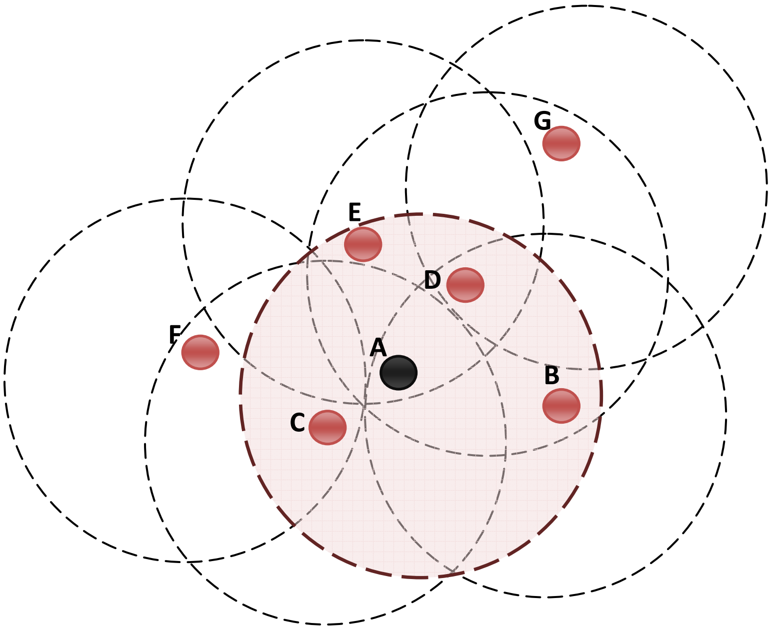


Figure 3.2a: Sending *Hello Message* (neighborhood discovery).

If the node that receives the message has not been covered by another node, it sets its state as covered, adopts the sender as its parent node, and answers back with a *Parent Recognition Message*, as shown in Figure 3.2b. This message also includes a selection metric (explained later) which is calculated based on the signal strength of the received *Hello Message,* the remaining energy in the node, and the interference of that node. The metric will be used later by the parent node to sort the candidates. If the receiver has been already covered by another node, it ignores the *Hello Message*. The IACDS algorithm uses four types of messages: *Hello Message*, *Parent Recognition Message*, *Children Recognition Message*, and *Sleeping Message*. If a parent node does not receive any *Parent Recognition Messages* from its neighbors, it also turns off, such as the case of nodes E and B in the final topology, as shown in Figure 3.2f, given that they have no children.

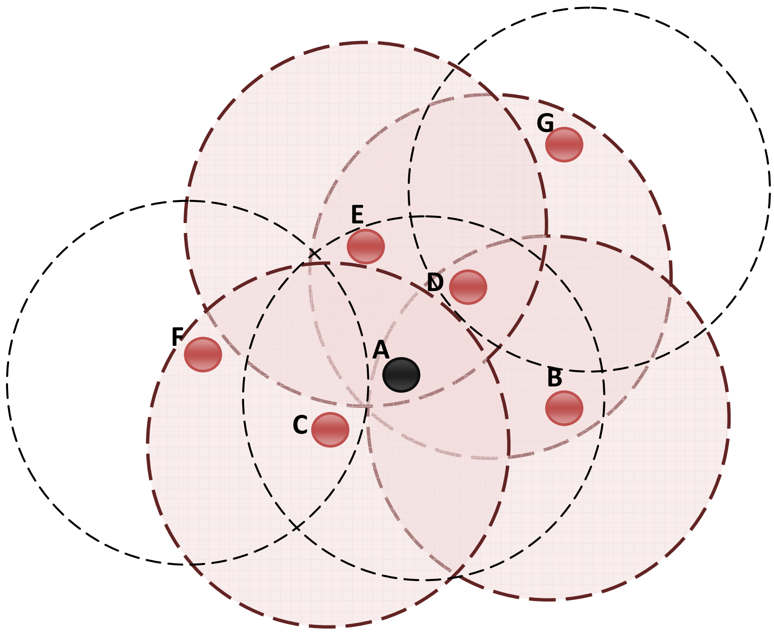


Figure 3.2b: Sending *Parent Recognition Message* (answer from candidates with selection metric).

2) Forming the children candidates list: The parent node sets a timeout for a certain amount of time to receive the answers from its neighbors. Each metric is stored in a list of candidates. Once this timeout expires, the parent node sorts the list in decreasing order according to the selection metric. The parent node then broadcasts a *Children Recognition Message* that includes the complete sorted list to all its candidates. In Figure 3.2c, node A sends the sorted list to nodes B, C, D, and E. Once the candidate nodes receive the list, they set a timeout period proportional to their position on the candidate list. During that timeout, nodes wait for *Sleeping Messages* from their brothers.

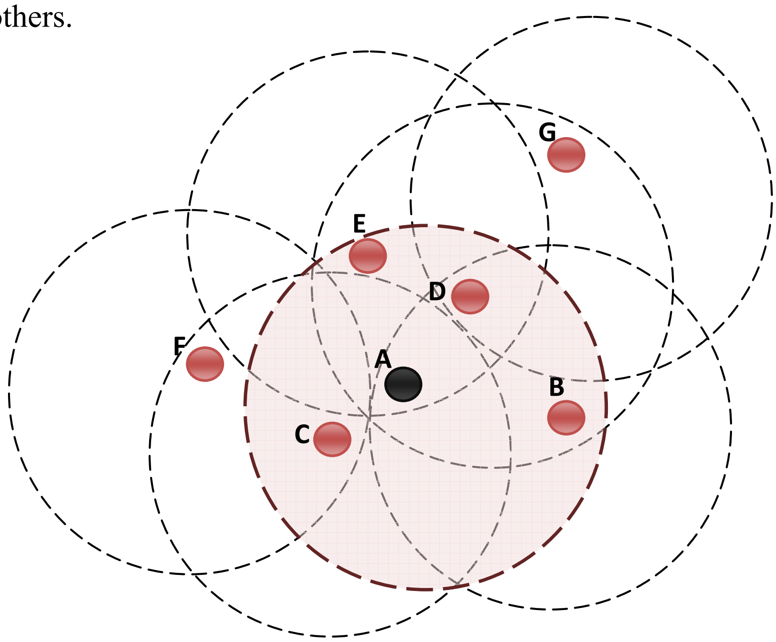


Figure 3.2c: Sending *Children Recognition Message* (includes a list of selected children).

If a node receives a *Sleeping Message* during the timeout period, it turns itself off, meaning that one of its brothers is better qualified to become part of the tree. Based on this scheme, the best node according to the metric will send a *Sleeping Message* first, blocking any other node in its range. Therefore, only the other candidate nodes outside its area of coverage have the opportunity to start their own generation process. For example, in Figure 3.2d, node D received a *Sleeping Message* from E before its timer expired, so it turned off. Otherwise, it sends a *Sleeping Message* to turn its brothers off. At that time, this particular node becomes a new parent node and starts its own process of looking for candidates.

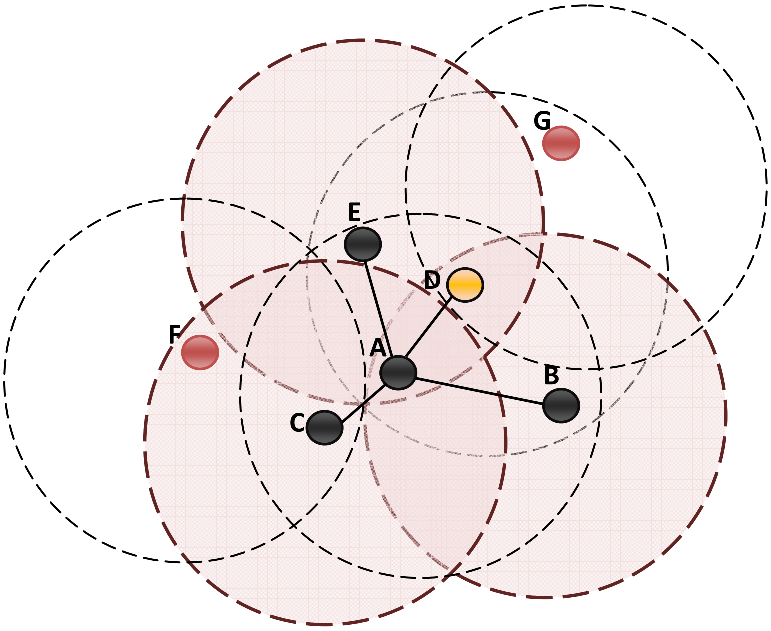


Figure 3.2d: Sending *Sleeping Message* (nodes which receive this message will turn off).

3) State decision based on the proposed selection metric: As explained before, nodes that received a *Hello Message* from the parent node, calculate the Received Signal Strength Indicator (RSSI) of the received signal, and then calculate a metric that is sent back to the parent in the *Parent Recognition Message*. Upon receiving the metrics from all its children, the parent node creates and sends a sorted list that, at the end, determines which nodes will be part of the CDS. The IACDS algorithm calculates the benefit of including a new node in the CDS using a metric that is proportional to the interference of that node, remaining energy of the node, and the distance from the parent node, as defined in Equation 3.10:

$$Mc,p = Wi * \frac{1}{IA(c)} + We * \frac{Ec}{Emax} + Wd * \frac{RSSIp}{RSSImin} \quad 3.10$$

Where *c* is the candidate node, *p* is its parent node, *We* is the weight for the remaining energy in the node, *Ec* is the remaining energy in node *c*, *Emax* is the maximum initial energy, *Wd* is the weight for the distance from the parent node, *RSSIp* is the received signal strength from the parent node, and *RSSImin* is the minimum RSSI to ensure connectivity, which is given by the sensitivity of the receiver. The interference amount $IA(c)$ of the receiver node is calculated using Equation 3.9.

Equation 3.10 produces a value between 0 and 1 which is assigned to each neighbor in the process of selecting the new nodes in the CDS; the higher the value of the metric, the higher the priority. As it can be seen from Equation 3.10, the selection metric gives priority to those nodes with minimum interference, higher energy, and which are farther away from the parent node. The final effect of this choice is to have a CDS with minimum interference, fewer active nodes, and better coverage.

However, proper weight manipulation can satisfy different criteria, as needed. As the main goal of the proposed algorithm is to reduce the global interference of the network, the interference metric is weighted to its maximum value, 1. If communication coverage is to be optimized and the average height of the node in the CDS (number of hops) needs to be reduced, the distance metric must be weighted more heavily. The downside is that low energy nodes may be included in the CDS, which may introduce early failures of nodes, and therefore reduce the lifetime of the network. On the other hand, if reliability of the tree is desired, energy must be weighted more. The downside is that the CDS may present more active nodes. In my work, a balanced average that compromises these two aspects is used.

4) Bonus opportunity: Although this methodology works very well, there are some cases in which a node sent to sleep is a bottleneck access to a section of the network. In order to avoid this situation, every node sets a timer once it receives the *Sleeping Message* to send a *Hello Message* and starts its own building process. As shown in Figure 3.2e, node D will wake up, send a *Hello Message* and will find node G uncovered, so node D will become active. This operation increases the overhead of the algorithm, but guarantees total coverage of the nodes in the graph.

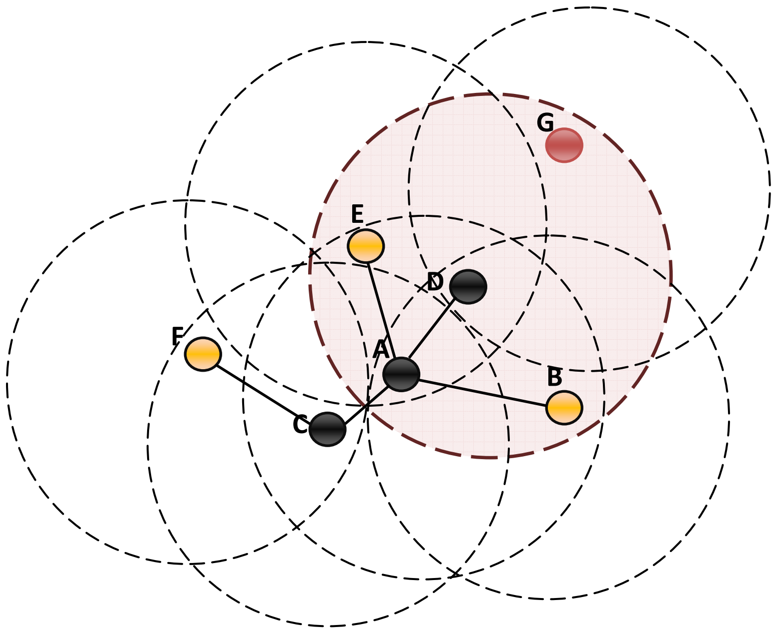


Figure 3.2e: Checking for children.

It is worth emphasizing that IACDS is completely distributed and needs no synchronization scheme. The process finishes when the last node finishes its own creation process. Each node is responsible for its own process and needs no information about the status of the overall process. Actually, nodes can start their application-related tasks as soon as they are selected as part of the CDS tree.

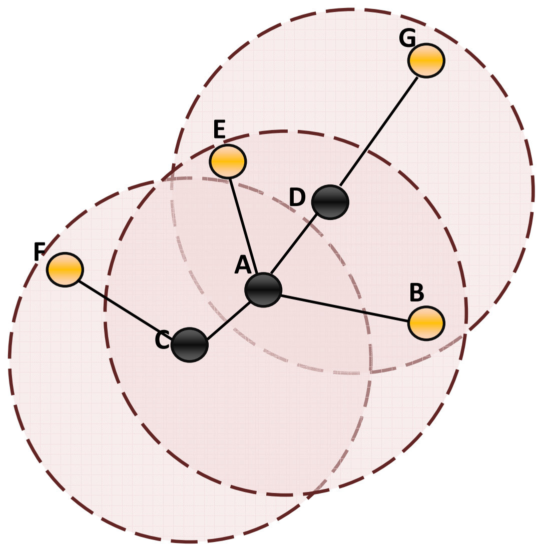


Figure 3.2f: Final topology.

The computational complexity of the IACDS algorithm can be easily calculated based on the fact that the sorting function executed by the parent nodes is the most expensive operation. Therefore, the complexity of the algorithm is given by the complexity of sorting routine, which can be bounded as $O(n\log(n))$. The message complexity is bounded by the worst case of 4 messages in the case of a node that becomes a parent node in the first opportunity: *Hello*, *Parent Recognition*, *Children Recognition*, and *Sleeping* messages. However, the number of messages is still defined by $O(n)$ for a network with n nodes, with a worst case scenario of $4n$ messages.

3.2.3 Finite State Machine and Message Sequence Diagrams

Although describing a protocol in words is useful for understanding its operation, they are more rigorously defined by Finite State Machines, especially in those cases where nodes change several times from one state to another during the execution of the protocol, and by a timeline of message exchanges. These diagrams are shown in Figure 3.3(a, b) and Figure 3.4(a, b), respectively.

The first part of the algorithm is a *HELLO-REPLY* sequence that is used in the neighborhood discovery process. The message sequence is simple: one message announces the presence of a node, and a set of nodes, whose number is unknown, answers back with a reply message. Given that the initiator has no idea of how many nodes are within its transmission range, it waits for a certain amount of time. This timer can be static (a fixed value or a random value defined on the fly) or dynamic (value is changed after the reception of a new reply message).

The second part of the protocol consists of the selection and notification processes. In this case, the sender node selects the next generation of possible active nodes based on a policy (the selection metric), and notifies them one by one using unicast messages.

The third part is the initiation of the other protocols. Topology construction protocols are only used to reduce the size of the initial topology, but they do not necessarily take care of maintaining this topology during its operation, or send data messages to the sink.

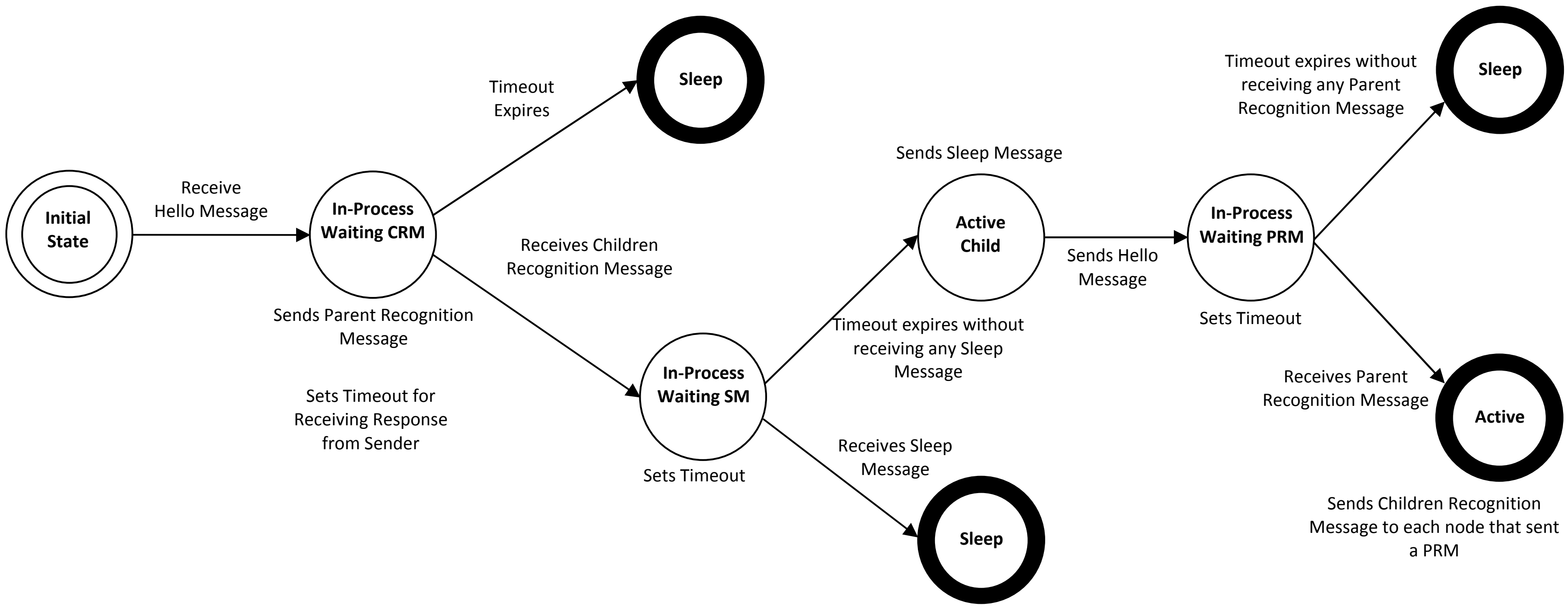


Figure 3.3a: Finite state machine illustrates state changes in IACDS algorithm.

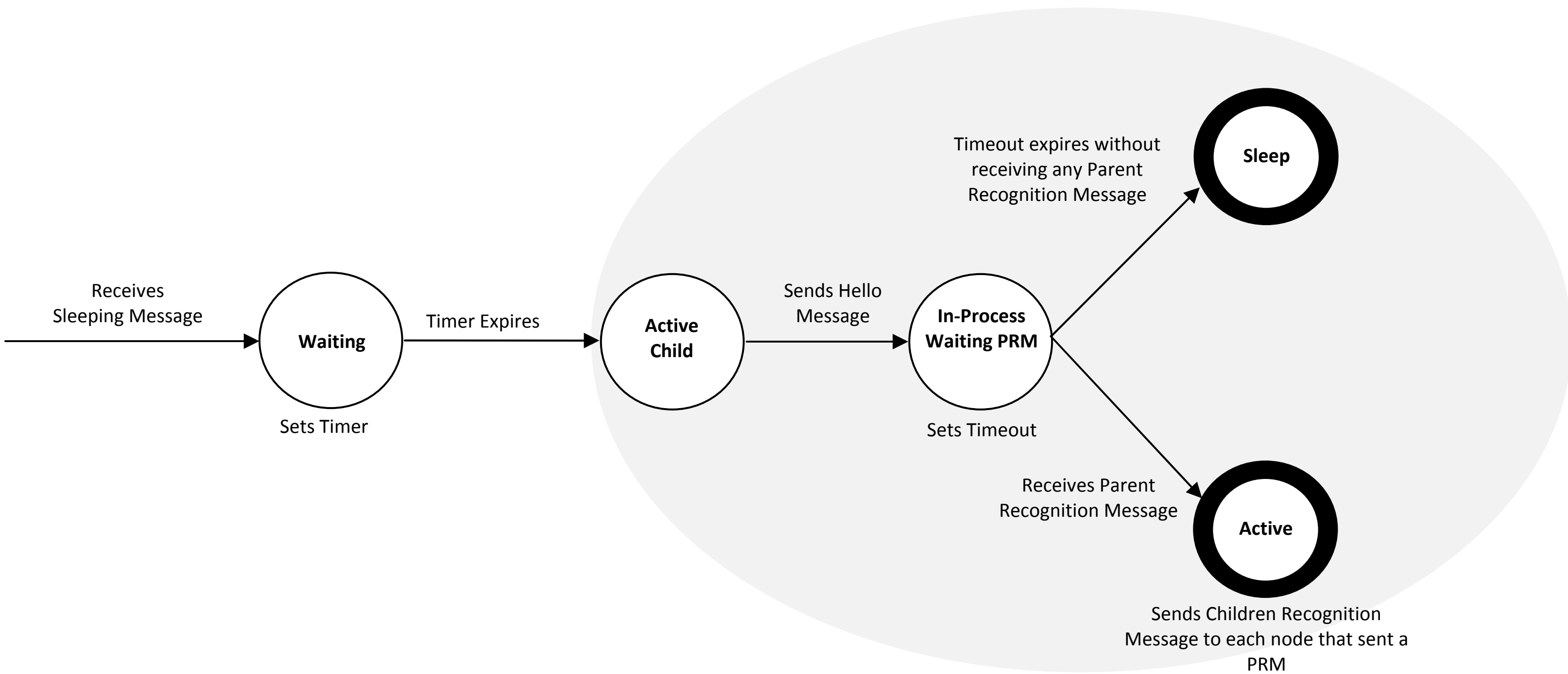


Figure 3.3b: Finite state machine illustrates state changes in IACDS algorithm (Bonus opportunity).

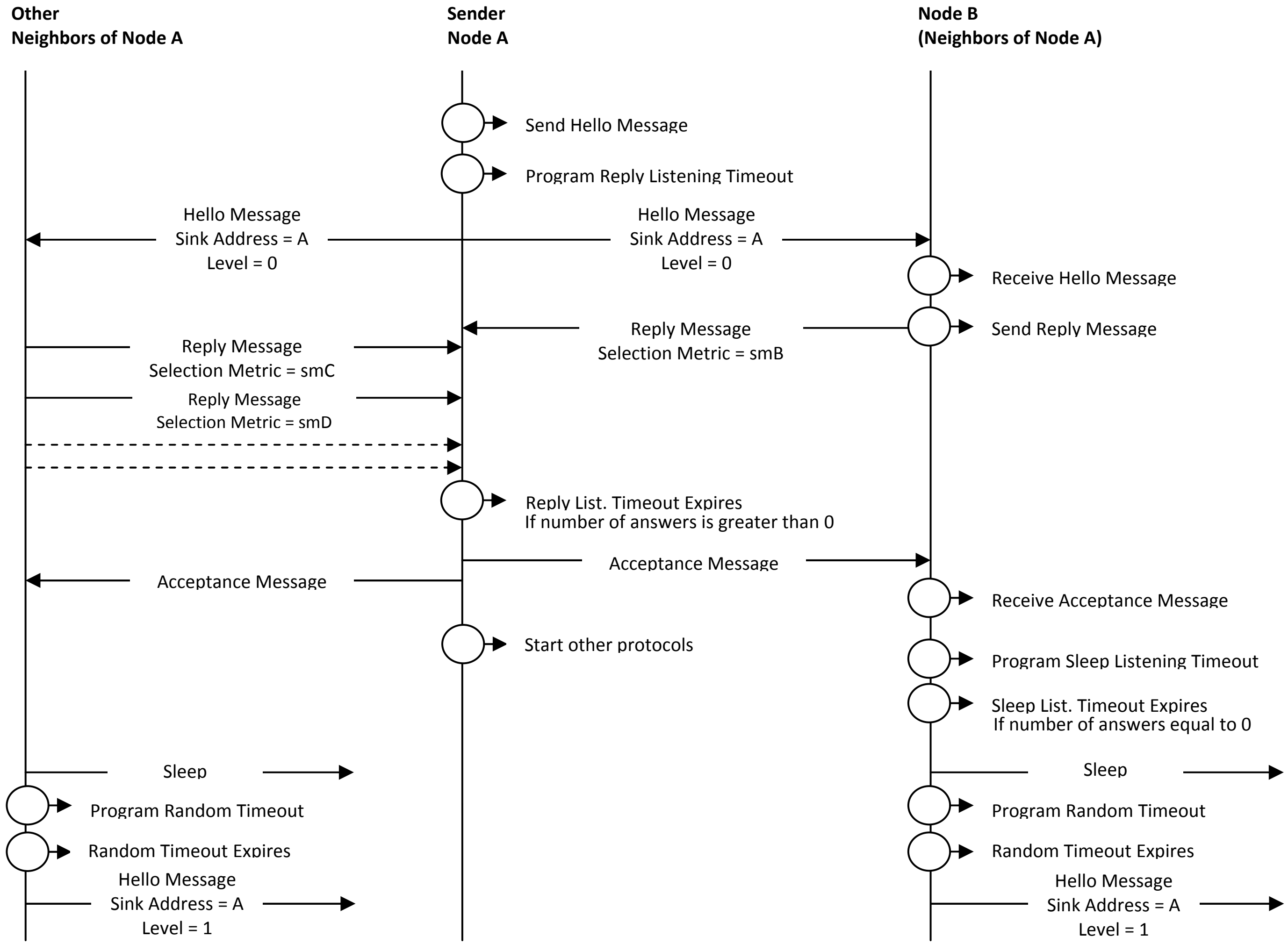


Figure 3.4a: message sequence diagram illustrates message exchanges in IACDS algorithm.

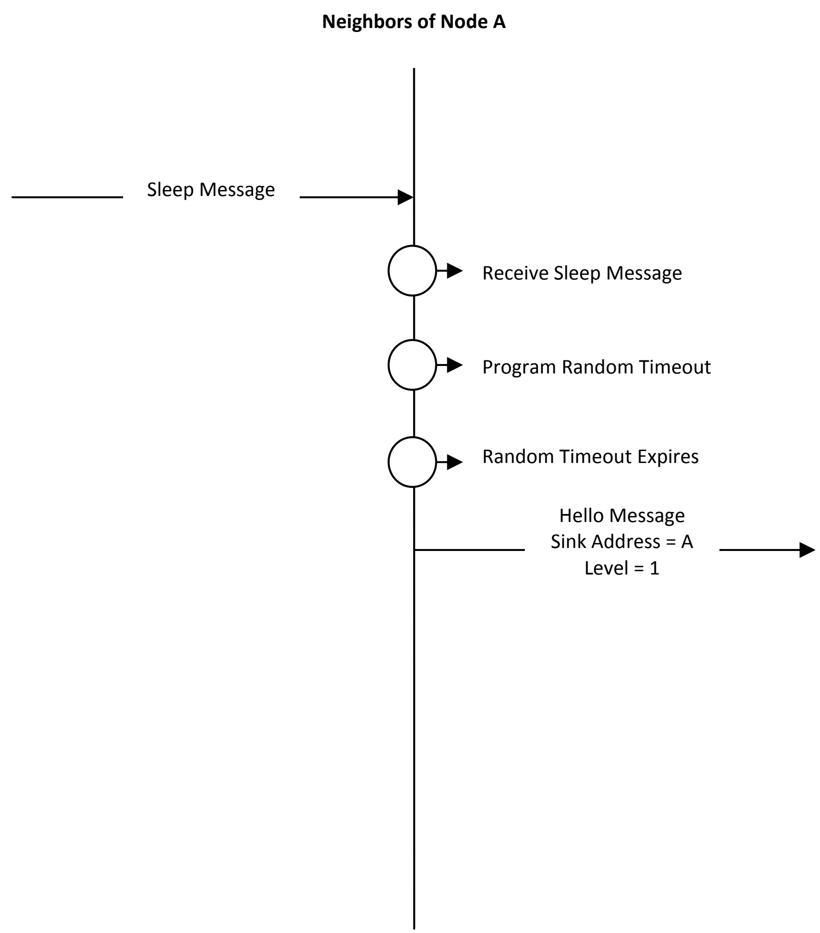


Figure 3.4b: Message sequence diagram illustrates message exchanges in IACDS algorithm (Bonus opportunity).

CHAPTER 4

Performance Evaluation of IACDS TC Algorithm

In this chapter, the results of the performance evaluation of IACDS algorithm with comparisons to other algorithms are presented.

4.1 *Atarraya* Simulator

Topology Control is a well-known technique in wireless mobile ad hoc and sensor networks. Despite the fact that topology control algorithms and protocols have been extensively studied, they are currently unavailable in most, if not all, simulation tools. In this thesis we use *Atarraya*, a discrete-event simulation tool specifically designed for testing, designing, implementing, and teaching topology control algorithms for wireless mobile ad hoc and sensor networks. *Atarraya* is an event-driven simulator developed in Java that presents a new framework for designing and testing topology control algorithms. It is an open source application [61].

4.2 Validation of the Proposed Interference Reduction Mechanism: (IACDS) Algorithm

Case 1: Number of nodes is 10 and communication radius is 63.

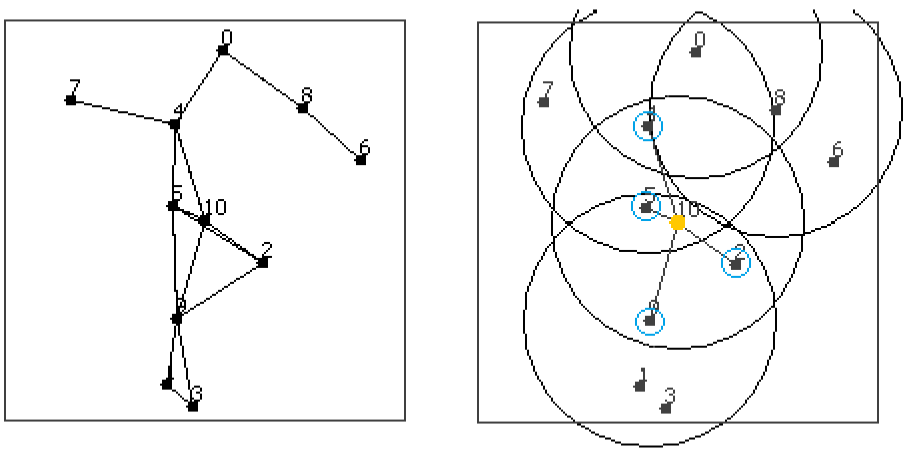


Figure 4.1: The initial deployment of 10 nodes network and starting the IACDS algorithm.

The sink, node 10 in Figure 4.1, starts the algorithm by sending an initial *Hello Message*. This message will allow the neighbors of node 10 to know their parent node. As illustrated in Figure 4.1, nodes 2, 4, 5, and 9 are the only nodes that will receive the *Hello Message*, while nodes 0, 1, 3, 6, 7, and 8 are out of reach from node 10. Consequently, nodes 2, 4, 5, and 9 set their state as covered, adopt node 10 as their parent node, and answer back with a *Parent Recognition Message*. This message includes the selection metric which is calculated based on the signal strength of the received *Hello Message,* the remaining energy in the node, and the interference amount of that node. Node 10 in turn receives the *Parent Recognition Message*s from its neighbors, which include the selection metric of each one , sorts the list of candidates in decreasing order according to the selection metric, and then broadcasts a *Children Recognition Message* that includes the complete sorted list to all its candidates (nodes 2, 4, 5, and 9). Once the candidate nodes receive the candidates list, they set a timeout period proportional to their position on the candidate list.

Table 4.1 shows the list of candidates of the sink node 10 and their corresponding selection metrics after applying the proposed IACDS algorithm. The list contains four candidates, namely, nodes 9, 4, 2, and 5.

Table 4.1: Candidates of node 10 (sink node) with their corresponding metrics.

Node ID	Selection Metric
9	0.90
4	0.89
2	0.78
5	0.63

It is clearly seen from Table 4.1 that node 9 is the best candidate according to the selection metric. It will send a *Sleeping Message* first, and thereby blocking other candidate nodes in its range, namely, nodes 5 and 2 from starting a new generation process. Node 4 is another candidate node and outside the coverage area of node 9 (will not be affected by the *Sleeping Message* of node 9); it has the opportunity to start its own generation process. At this moment only nodes 10, 9, and 4 are active.

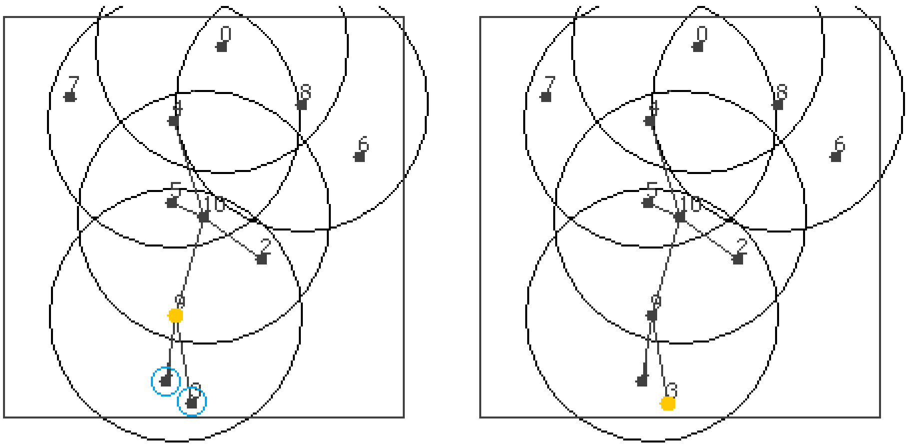


Figure 4.2: Nodes 9 and 3 start their own generation process.

Node 9 becomes new parent node and starts its own process of looking for candidates in the same manner as done by node 10. As illustrated in Figure 4.2, node 9 receives a *Parent Recognition Message*s from only two neighbors, namely, nodes 3 and 1. Consequently, it sorts the candidates list according to the selection metric and then broadcasts a *Children Recognition Message* to its candidates in the sorted list.

Table 4.2 shows the list of candidates of node 9 and their corresponding selection metrics. The list contains two candidates.

Table 4.2: Candidates of node 9 with their corresponding metrics.

Node ID	Selection Metric
3	0.85
1	0.76

As shown in Table 4.2, node 3 is the best candidate according to the selection metric. So, node 3 blocks node 1 from starting a new generation process by sending a *Sleeping Message* first, and then starts its own generation process which ends with the formation of the sorted list of candidates. As shown in Figure 4.2, node 3 becomes new parent node and starts its own process of looking for candidates. The candidates list of node 3 is empty and node 3 does not receive any *Parent Recognition Messages* from its neighbors so, it turns off.

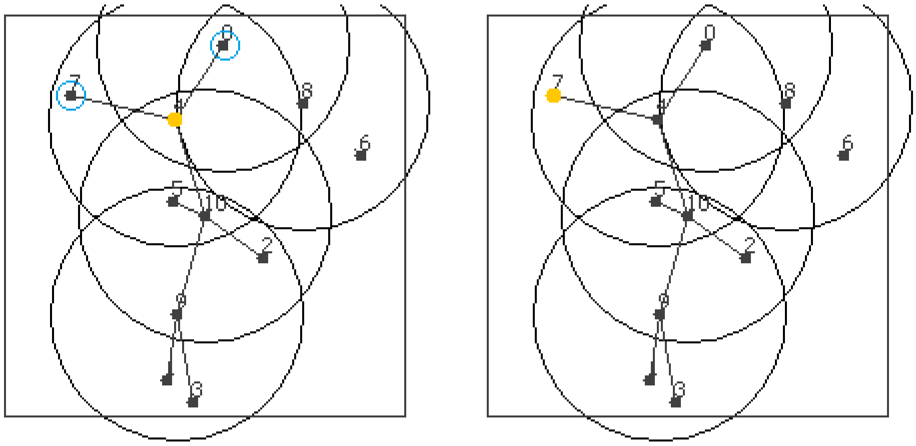


Figure 4.3: Nodes 4 and 7 start their own generation process.

Similarly, node 4 becomes a new parent node and starts its own process of looking for candidates. As illustrated in Figure 4.3, node 4 receives *Parent Recognition Message*s from only two neighbors, namely, nodes 0 and 7. Consequently, it sorts the candidates list according to the selection metric and then broadcasts a *Children Recognition Message* to its candidates in the sorted list.

Table 4.3 shows the list of candidates of node 4 and their corresponding selection metrics. The list contains two candidates, namely, nodes 0 and 7.

Table 4.3: Candidates of node 4 with their corresponding metrics.

Node ID	Selection Metric
7	0.92
0	0.84

As shown in Table 4.3, node 7 is the best candidate according to the selection metric. It will send a *Sleeping Message* first, and thereby blocking other candidate nodes in its range, namely, node 0 from starting a new generation process. Then, it starts its own generation process which ends with the formation of the sorted list of candidates. As shown in Figure 4.3, node 7 starts its own process of looking for candidates. The candidates list of node 7 is empty and node 7 does not receive any *Parent Recognition Message* from its neighbors, so, it turns off.

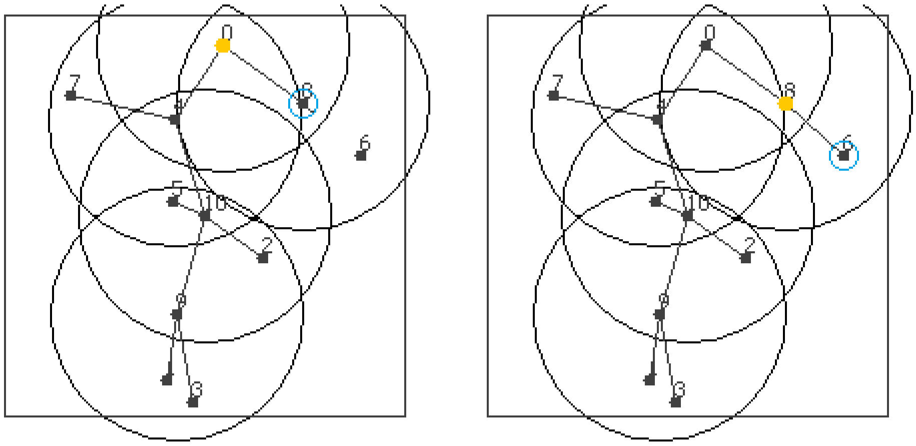


Figure 4.4: Nodes 0 and 8 start their own generation process.

Every node sets a timer once it receives the *Sleeping Message* to take the opportunity to send a *Hello Message* and starts its own building process (bonus opportunity). As shown in Figure 4.4, node 0 will wake up and send a *Hello Message.* Consequently, it finds node 8 uncovered. Node 8 sets its state as covered, adopts node 0 as its parent node, and answers back with a *Parent Recognition Message*. Node 0 receives the *PRM* from node 8, and forms its candidates list. At this moment node 0 will become active.

Table 4.4 shows the list of candidates of node 0 and their corresponding selection metrics. The list contains only one candidate, namely, node 8.

Table 4.4: Candidates of node 0 with their corresponding metrics.

Node ID	Selection Metric
8	0.88

As shown in Table 4.4, node 8 is the only candidate in the list and it has the opportunity to become a new parent and start its own process of looking for candidates. Consequently, it finds node 6 uncovered. Node 6 sets its state as covered, adopts node 8 as its parent node, and answers back with a *Parent Recognition Message*. Node 8 receives the *PRM* from node 6, and forms its candidates list. At this moment node 8 will become active. This operation increases the overhead of the algorithm, but guarantees total coverage of the nodes in the graph.

Table 4.5 shows the list of candidates of node 8 and their corresponding selection metrics. The list contains only one candidate, namely, node 6.

Table 4.5: Candidates of node 8 with their corresponding metrics.

Node ID	Selection Metric
6	0.80

As shown in Table 4.5, node 6 is the only candidate in the list and it has the opportunity to become a new parent and start its own process of looking for candidates.

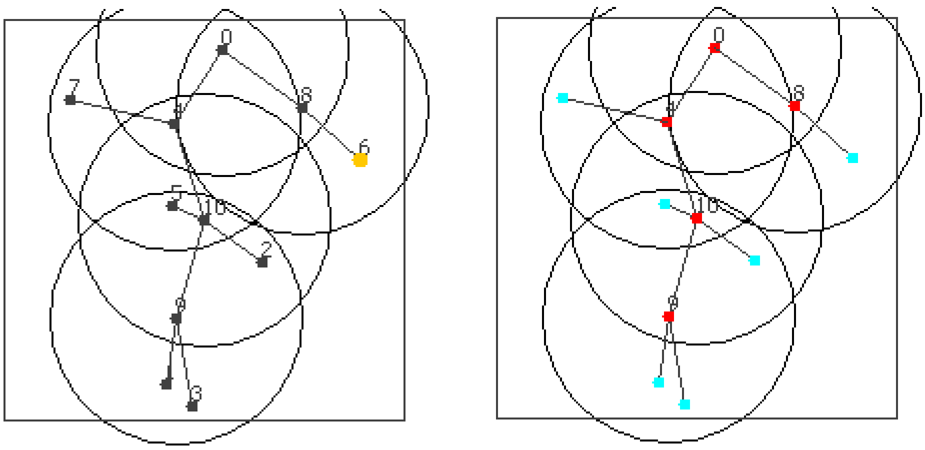


Figure 4.5: Node 6 starts its own generation process, and the final network.

Node 6 starts its own process of looking for candidates in the same manner as node 8. Figure 4.5 shows that the candidates list of node 6 is empty and it does not receive any *Parent Recognition Message* from any neighbor, so, it turns off. The final topology of this scenario is illustrated in Figure 4.5. The IACDS algorithm ends with a topology of only 5 active nodes (10, 9, 4, 0, and 8) from original 10 nodes network; with guarantees of total coverage of the nodes in the network and that only the best candidate in the coverage area is active, which leads to minimum interference in the resulting topology, minimum energy consumption, and consequently extending the network life time. The final topology shows that among the candidates of node 10, only node 9 is active in the coverage area of node 9 and similarly in the coverage area of node 4, only node 4 is active. The bonus opportunity allows node 0 to be active, this guarantees the coverage of the nodes in the network. Without this opportunity, nodes 8 and 6 would not be covered.

Case 2: Number of nodes is 20 and communication radius is 63.

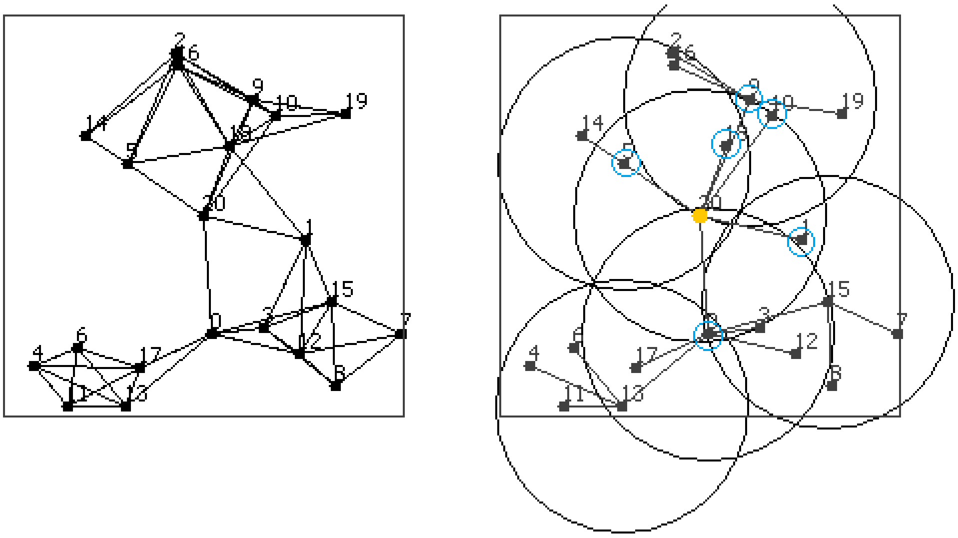


Figure 4.6: The initial deployment of 20 nodes network and starting the IACDS algorithm.

The sink node, 20, starts the algorithm by sending an initial *Hello Message*. This message will allow the neighbors of node 20 to know their parent node. As illustrated in Figure 4.6, nodes 0, 1, 5, 9, 10, and 18 are the only nodes that will receive the *Hello Message*, while other nodes are out of reach from node 20. Consequently, nodes 0, 1, 5, 9, 10, and 18 set their state as covered, adopt node 20 as their parent node, and answer back with a *Parent Recognition Message*. This message includes the selection metric which is calculated based on the signal strength of the received *Hello Message,* the remaining energy in the node, and the interference amount of that node. Node 20 in turn receives the *Parent Recognition Message*s from its neighbors, which include the selection metric of each one , sorts the list of candidates in decreasing order according to the selection metric, and then broadcasts a *Children Recognition Message* that includes the complete sorted list to all its candidates (nodes 0, 1, 5, 9, 10, and 18). Once the candidate nodes receive the candidates list, they set a timeout period proportional to their position on the candidate list.

Table 4.6 shows the list of candidates of the sink node 20 and their corresponding selection metrics after applying the proposed IACDS algorithm. The list contains six candidates, nodes 0, 1, 5, 9, 10, and 18.

Table 4.6: Candidates of node 20 (sink node) with their corresponding metrics.

Node ID	Selection Metric
9	0.99
10	0.99
0	0.96
1	0.91
5	0.86
18	0.79

It is clearly seen from Table 4.6 that node 9 is the best candidate according to the selection metric. It will send a *Sleeping Message* first, and thereby blocking other candidate nodes in its range, namely, nodes 10 and 18 from starting a new generation process. Node 0 is another candidate node and outside the coverage area of node 9 (will not be affected by the *Sleeping Message* of node 9); it also has the opportunity to start its own generation process. Node 0 will send a *Sleeping Message* first, blocking other nodes in its range, namely, node 1. Node 5 is another candidate node outside the area of coverage of both nodes 9 and 0 (will not be affected by the *Sleeping Message* sent by either node 9 or 0); it similarly has the opportunity to start its own generation process. At this moment only nodes 20, 9, 0, and 5 are active.

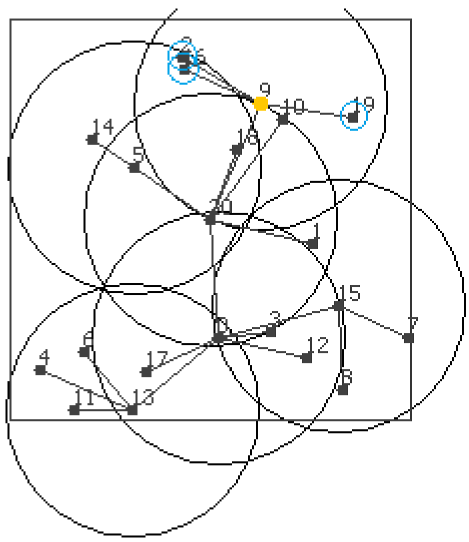


Figure 4.7: Node 9 starts its own generation process.

Node 9 becomes a new parent and starts its own process of looking for candidates in the same manner as done by node 20. As illustrated in Figure 4.7, node 9 receives *Parent Recognition Messages* from three neighbors, namely, nodes 2, 16, and 19. Consequently, it sorts the candidates list according to the selection metric and then broadcasts a *Children Recognition Message* to its candidates in the sorted list.

Table 4.7 shows the list of candidates of node 9 and their corresponding selection metrics. The list contains three candidates.

Table 4.7: Candidates of node 9 with their corresponding metrics.

Node ID	Selection Metric
19	0.87
2	0.85
16	0.83

As shown in Table 4.7, node 19 is the best candidate according to the selection metric. It will send a *Sleeping Message* first, and thereby blocking other candidate nodes in its range, and then starts its own generation process which ends with the formation of the sorted list of candidates. Node 2 is another candidate node and outside the coverage area of node 19; it also has the opportunity to start its own generation process. Node 2 will send a *Sleeping Message* first, blocking other nodes in its range, namely, node 16.

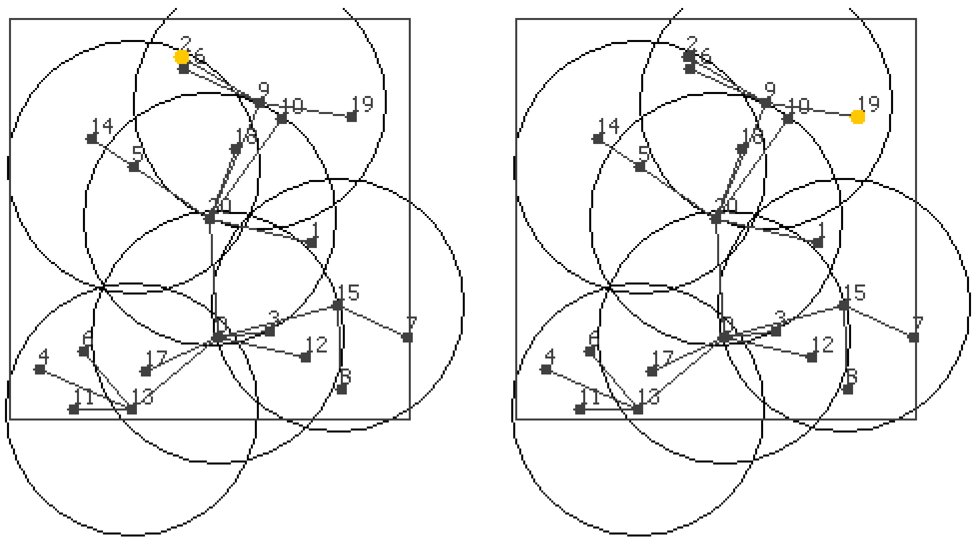


Figure 4.8: Nodes 2 and 19 start their own generation process.

Nodes 19 and 2 become new parents and start their own process of looking for candidates. As illustrated in Figure 4.8, the candidates lists of both node 19 and 2 are empty and neither node 19 nor node 2 receive any *Parent Recognition Message* from their neighbors so, both turn off.

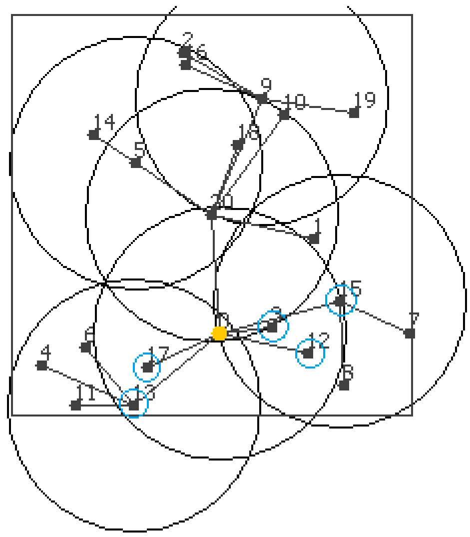


Figure 4.9: Node 0 starts its own generation process.

Node 0 which is the second candidate of the sink becomes a new parent and starts its own process of looking for candidates in the same manner as done by node 9. As illustrated in Figure 4.9, node 0 receives *Parent Recognition Message*s from five neighbors, namely, nodes 3, 12, 13, 15, and 17. Consequently, it sorts the candidates list according to the selection metric and then broadcasts a *Children Recognition Message* to its candidates in the sorted list.

Table 4.8 shows the list of candidates of node 0 and their corresponding selection metrics. The list contains five candidates.

Table 4.8: Candidates of node 0 with their corresponding metrics.

Node ID	Selection Metric
15	0.99
13	0.94
12	0.85
17	0.81
3	0.70

It is clearly seen from Table 4.8 that node 15 is the best candidate according to the selection metric. It will send a *Sleeping Message* first, and thereby blocking other candidate nodes in its range, namely, nodes 12 and 3 from starting a new generation process. Then starts its own generation process which ends with the formation of the sorted list of candidates. Node 13 is another candidate node and outside the coverage area of node 15 (will not be affected by the *Sleeping Message* of node 15); it has the opportunity to start its own generation process. Node 13 will send a *Sleeping Message*, blocking other nodes in its range, namely, node 17, from starting a new generation process.

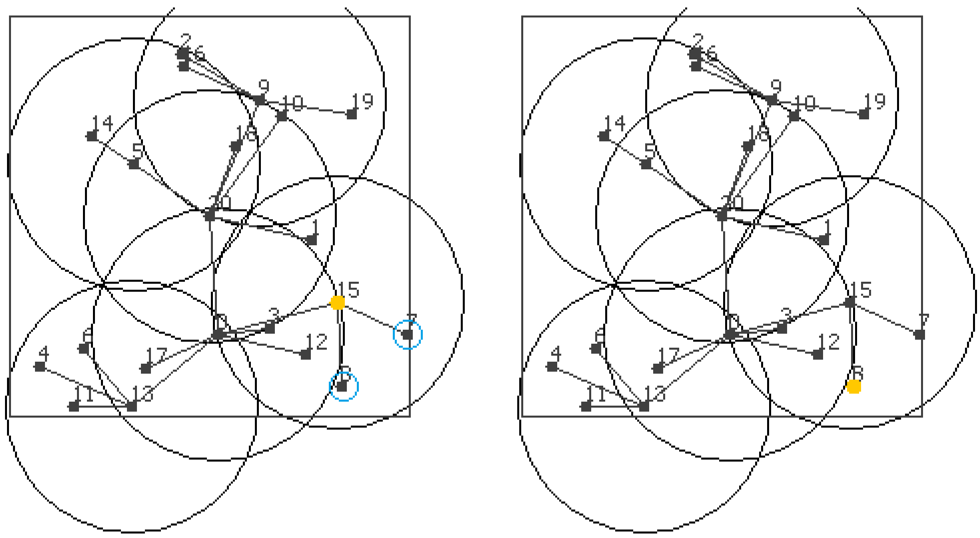


Figure 4.10: Node 15 starts its own generation process.

Node 15 becomes a new parent and starts its own process of looking for candidates in the same manner as done by node 0. As illustrated in Figure 4.10, node 15 receives *Parent Recognition Messages* from two neighbors, namely, nodes 7 and 8. These messages include the selection metric which is calculated based on the signal strength of the received *Hello Message,* the remaining energy in the node, and the interference amount of that node. Consequently, it sorts the candidates list according to that selection metric and then broadcasts a *Children Recognition Message* to its candidates in the sorted list. At this moment also node 15 is active.

Table 4.9 shows the list of candidates of node 15 and their corresponding selection metrics. The list contains two candidates, namely, nodes 7 and 8.

Table 4.9: Candidates of node 15 with their corresponding metrics.

Node ID	Selection Metric
8	0. 83
7	0.80

Table 4.9 shows that node 8 is the best candidate according to the selection metric. It will send a *Sleeping Message* first, and thereby blocking other candidate nodes in its range, namely, node 7, from starting a new generation process. Then, it starts its own generation process which ends with the formation of the sorted list of candidates. As shown in Figure 4.10, the candidates list of node 8 is empty and node 8 does not receive any *Parent Recognition Message* from its neighbors so, it turns off.

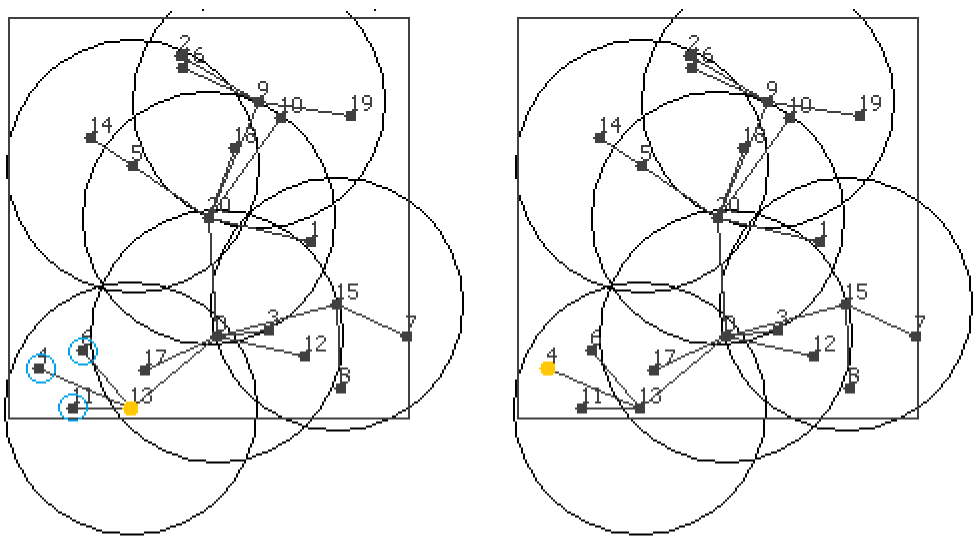


Figure 4.11: Node 13 starts its own generation process.

Node 13 becomes a new parent and starts its own process of looking for candidates in the same manner as done by node 15. As illustrated in Figure 4.11, node 13 receives *Parent Recognition Message*s from three neighbors, namely, nodes 4, 6, and 11. Consequently, it sorts the candidates list according to the selection metric and then broadcasts a *Children Recognition Message* to its candidates in the sorted list. At this moment also node 13 is active.

Table 4.10 shows the list of candidates of node 13 and their corresponding selection metrics. The list contains three candidates, namely, nodes 4, 6, and 11.

Table 4.10: Candidates of node 13 with their corresponding metrics.

Node ID	Selection Metric
4	0.89
6	0.80
11	0.73

Table 4.10 shows that node 4 is the best candidate according to the selection metric. It will send a *Sleeping Message* first, and thereby blocking other candidate nodes in its range, namely, nodes 6 and 11, from starting a new generation process. Then it starts its own generation process which ends with the formation of the sorted list of candidates. As shown in Figure 4.11, the candidates list of node 4 is empty and it does not receive any *Parent Recognition Message* from its neighbors so, it turns off.

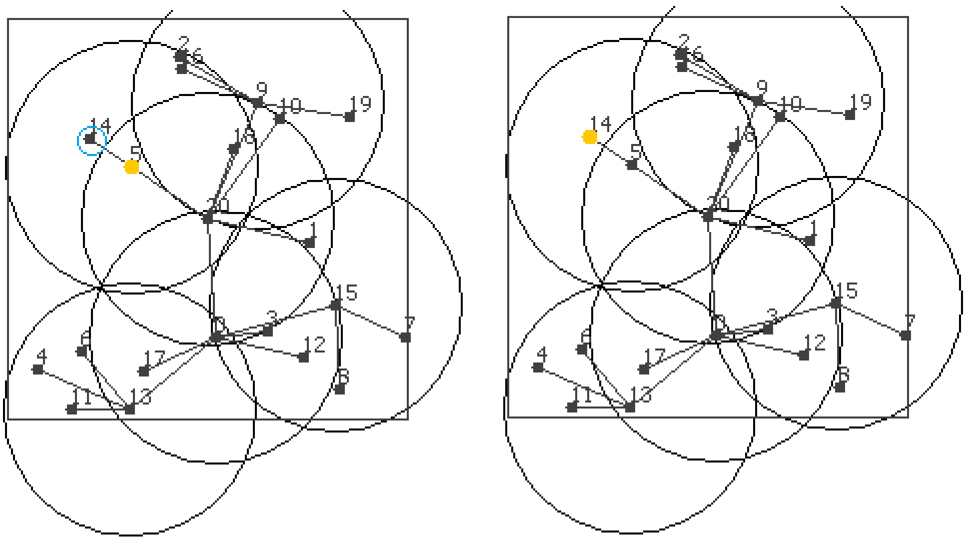


Figure 4.12: Node 5 starts its own generation process.

Node 5 which is the other candidate of the sink node becomes a new parent and starts its own process of looking for candidates in the same manner. As illustrated in Figure 4.12, node 5 receives a *Parent Recognition Message* from only one neighbor, namely, node 14. Consequently, it sorts the candidates list according to the selection metric and then broadcasts a *Children Recognition Message* to its candidates in the sorted list.

Table 4.11 shows the list of candidates of node 5 and their corresponding selection metrics. The list contains only one candidate.

Table 4.11: Candidates of node 5 with their corresponding metrics.

Node ID	Selection Metric
14	0.69

As shown in Table 4.11, node 14 is the only candidate in the list and it has the opportunity to become a new parent and start its own process of looking for candidates. Figure 4.12 shows that the candidates list of node 14 is empty and it does not receive any *Parent Recognition Message* from its neighbors so, it turns off.

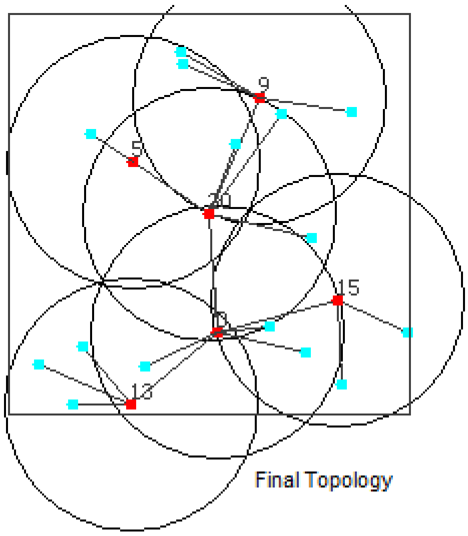


Figure 4.13: Final network.

The final topology of this scenario is illustrated in Figure 4.13. The IACDS algorithm ends with a topology of only 6 active nodes (20, 9, 0, 5, 13, and 15) from original 20 nodes network; with guarantees of total coverage of the nodes in the network and that only the best candidate in the coverage area is active, which leads to minimum interference in the resulting topology, minimum energy consumption, and consequently extending the network life time. The final topology shows that among the candidates of node 20, only node 9 is active in the coverage area of node 9 and so on.

4.3 Performance Evaluation

Three sets of simulations are included in this section. The first set maintains the number of nodes fixed and increases the node degree by changing the communication range of the nodes. The second set, on the other hand, varies the network density by changing the number of nodes while maintaining a fixed communication range. In these two simulations, the nodes are uniformly distributed in the area of deployment. The third set of simulations includes an additional theoretical comparison considering an ideal grid topology in which all nodes have the same number of neighbors. Two different grid topologies are used: the Grid HV topology, in which each node can listen to its horizontal and vertical neighbors; and the Grid HVD topology, in which each node can listen to its horizontal, vertical, and diagonal neighbors. Therefore, the number of neighbors in those topologies is at most 4 and 8, respectively. These scenarios are clearly illustrated in Figure 4.14. In all simulations, each result represents the average value of 50 random scenarios.

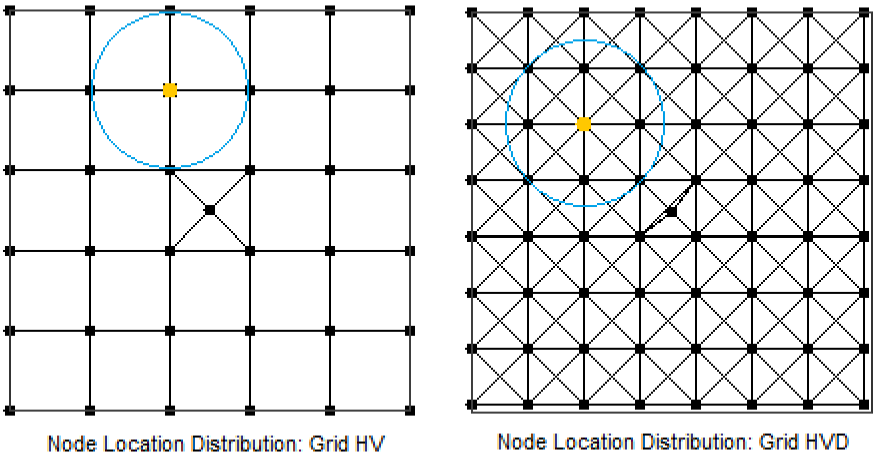


Figure 4.14: The ideal grid scenarios: Grid HV and Grid HVD.

To vary the communication range of the nodes, the critical transmission range CTR is used. CTR(n) is defined as the minimum communication range that will produce a connected topology. The theoretical formula of CTR presented in [62] is given by Equation 4.1:

$$CTR(n) = \sqrt{\frac{ln(n)+f(n)}{n\pi}} \qquad 4.1$$

Where *f(n)* is an arbitrary function such that $\lim_{n\to\infty} f(n) = +\infty$. Based on simulations, it has been seen that this CTR provides an average number of neighbors between 4 and 8, even for small values of n. As in [63], in this thesis *f(n) = ln(ln(n))* is used. To assess the performance of the proposed interference-aware topology construction algorithm, three main performance metrics are utilized: 1) number of active nodes in the resulting network, 2) number of messages used in the CDS building process, and 3) amount of energy used in the process. The first metric shows how the proposed interference reduction mechanism can effectively reduce the amount of active nodes while preserving network connectivity and coverage. The other two metrics show how efficient the algorithm is in terms of overhead and energy consumption, where:

$$Total\ spent\ energy\ ratio = \frac{Total\ initial\ energy - Total\ final\ energy}{Total\ initial\ energy} \quad 4.2$$

Two topology construction algorithms namely EECDS and CDS Rule K are used to evaluate the performance of the proposed interference-aware topology construction algorithm. The algorithms are evaluated in scenarios with sparse, medium-dense, and dense topologies. The node degree and the density of the network are modified by increasing the communication range of the nodes and the number of nodes in the network. The three algorithms were implemented in a Java-based simulation tool called *Atarraya* [61].

4.3.1 Simulation Environment and Parameters

The following assumptions were made during the simulation:

1. Nodes are located in a two dimensional space and have a perfect communication coverage disk.
2. The initial graph is connected.
3. Distances can be calculated as a metric perfectly proportional to the Received Signal Strength Indicator (RSSI).
4. Idle state energy consumption is assumed negligible.

The networks are constructed by uniformly distributing nodes in a 200×200 square area. Without loss of generality, the mean result is derived from 50 networks randomly generated with a fixed number of nodes and different transmission ranges for the first simulation (changing the node degree) and different number of nodes and fixed transmission range for the second one (changing the node density).

Table 4.12 presents a summary of the simulation parameters used in the performance evaluation of the proposed interference reduction mechanism.

Table 4.12: Simulation parameters.

	Simulation 1	Simulation 2	Simulation 3
Deployment area	200m × 200m		
Number of nodes	100	10, 20, 40, 60, 80, 100	36, 64
Transmission range	28, 42, 56, 70, 84m	63m	40m
Node distribution	Uniform (200,200)		Grid HV and Grid HVD
Instances per topology	50 instances		
Maximum energy	1 Joule		
IACDS weights	$W_I = 0.5$, $W_E = 0.5$, $W_D = 0.5$		

4.3.2 Simulation 1: Changing the Node Degree

This simulation mainly aims to compare the algorithms when the node degree of the network is changed by increasing the transmission range of the nodes while maintaining the number of nodes fixed = 100. Given that these algorithms work based on information from neighbors, it is important to measure their performance with neighborhoods of different sizes.

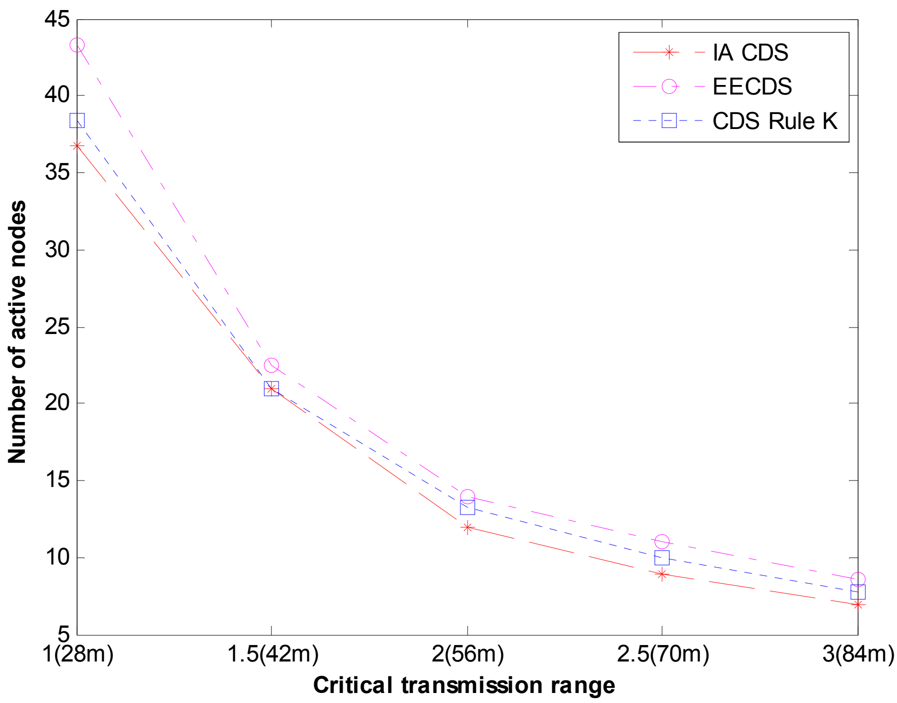


Figure 4.15: Number of active nodes versus transmission range of the nodes.

As it can be seen from Figure 4.15, the three algorithms produce CDSs with almost similar number of nodes. However, IACDS generates fewer nodes in all scenarios.

Another note to be seen from this figure, all the algorithms tend to decrease the number of active nodes with the node degree, as expected.

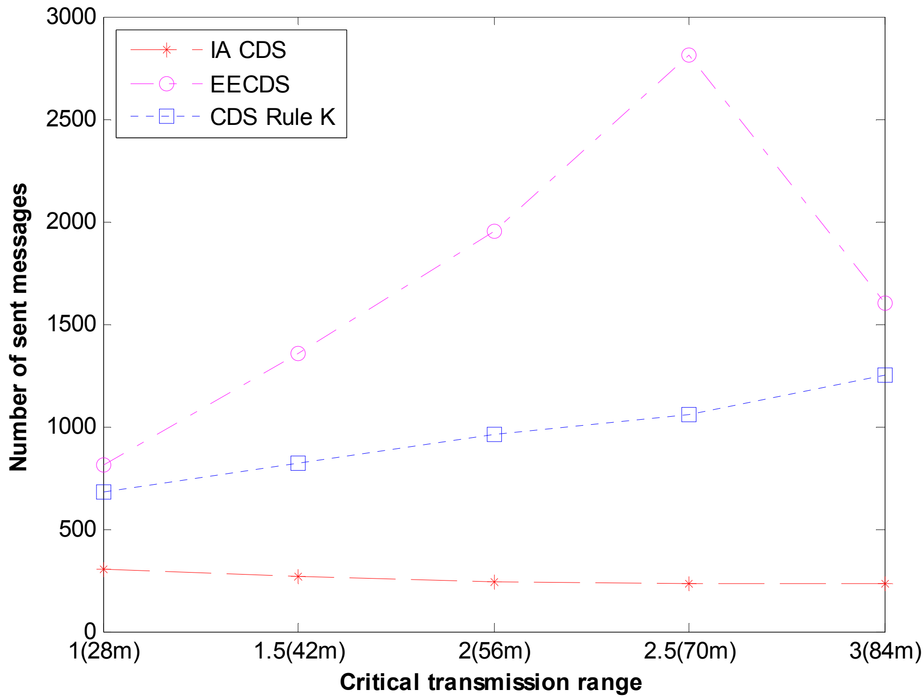


Figure 4.16: Number of sent messages versus transmission range of the nodes.

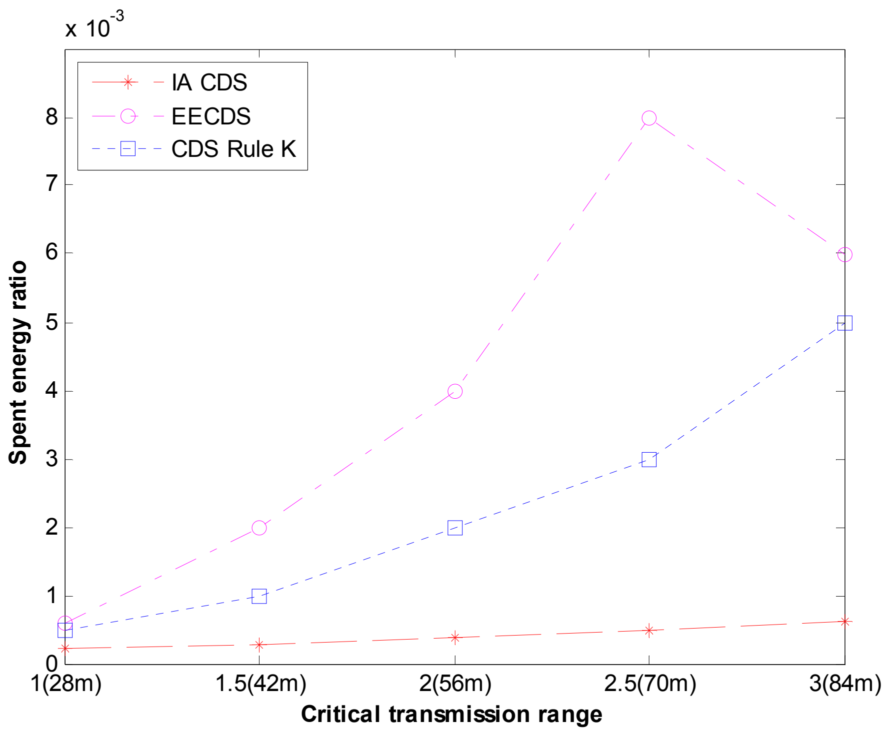
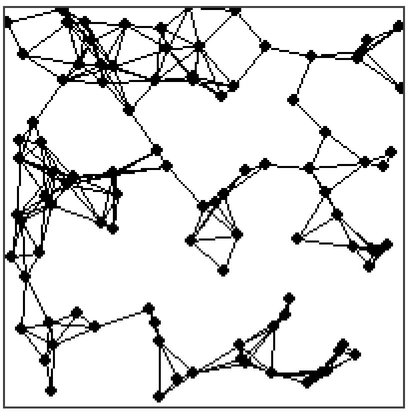


Figure 4.17: Spent energy ratio versus transmission range of the nodes.

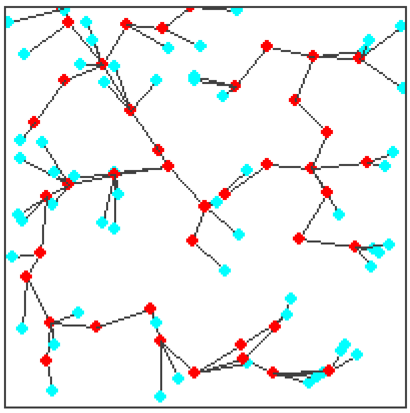
Figures 4.16 and 4.17 show two important metrics: the total energy and number of messages used to build the CDSs. In this case, the IACDS mechanism shows its superior performance. IACDS presents an almost constant energy consumption and number of messages compared with the EECDS and CDS-Rule-K algorithms, which show a non-linear increase trend. These results can be easily explained.

The non-linear behavior of the EECDS mechanism is explained by the competition used in both phases of the algorithm. This is due to the fact that with a higher communication range, more nodes are covered, and the network has fewer nodes in higher levels. This, at the same time, reduces the amount of nodes competing to become part of the CDS in the outer regions of the topology. In the case of the CDS-Rule-K algorithm, the factor that increases the amount of messages (and energy, consequently) is related to its pruning process in which every node must update nodes two hops away when it is unmarked. This overhead increases with the number of neighbors because more nodes will retransmit the message. Also, when the node degree increases, more nodes get unmarked and will produce this extra overhead. The linearity of IACDS is a consequence of the bounded number of messages that each node needs to transmit, which remains almost identical and never goes over 4*n* in ideal conditions. As mentioned before in chapter 3, the IACDS algorithm uses four types of messages: *Hello Message*, *Parent Recognition Message*, *Children Recognition Message*, and *Sleeping Message*. Figure 4.18 illustrates the behavior of the proposed interference-aware CDS topology control algorithm, IACDS, in a graphical manner. In this case, the number of nodes is fixed to 100 and the transmission ranges are varied.

Transmission Range: CTR=28m



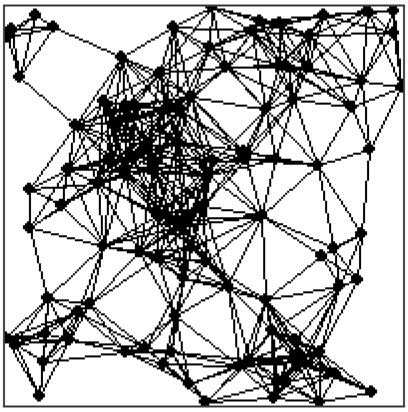
Original network



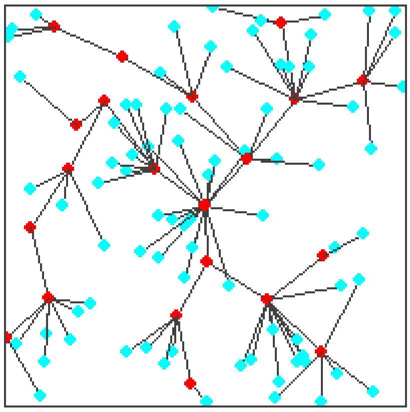
Resulting Network after applying IACDS

Figure 4.18: Topologies obtained after applying the proposed algorithm, *cont.*

Transmission Range: CTR=42m

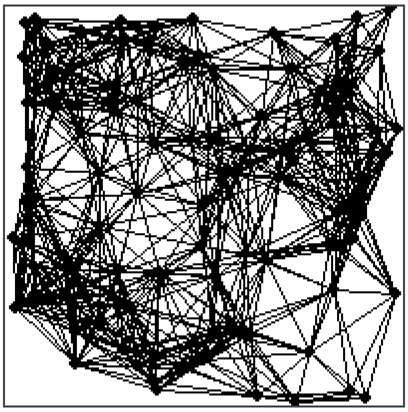


Original network

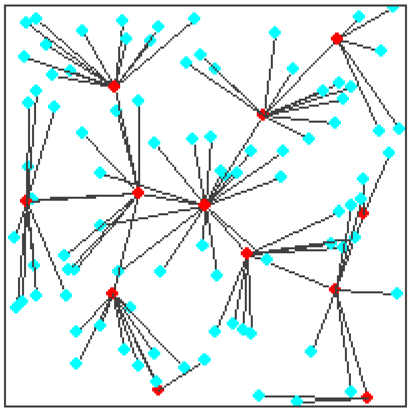


Resulting Network after applying IACDS

Transmission Range: CTR=56m

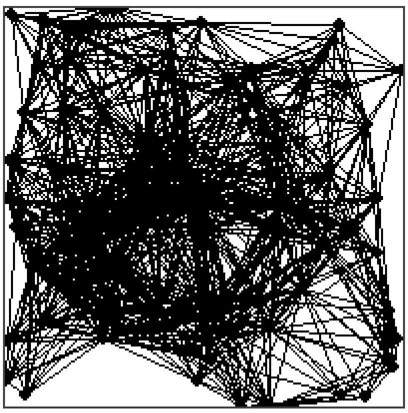


Original network

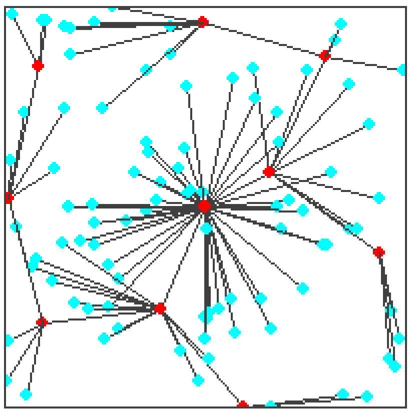


Resulting Network after applying IACDS

Transmission Range: CTR=70m



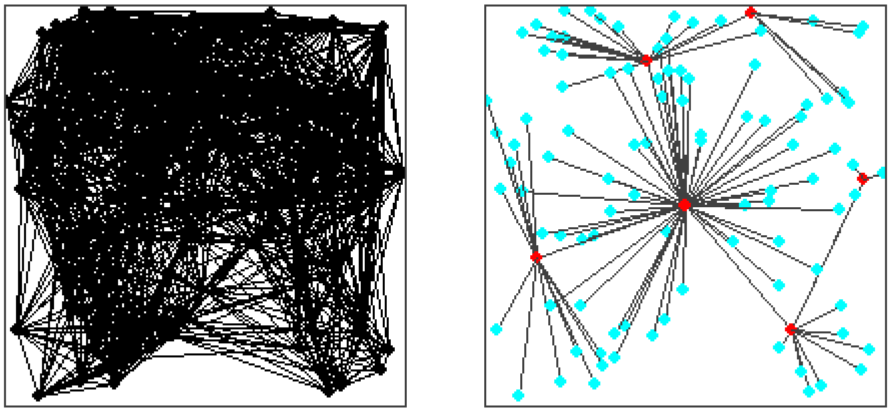
Original network



Resulting Network after applying IACDS

Figure 4.18: Topologies obtained after applying the proposed algorithm, *cont.*

Transmission Range: CTR=84m



Original network *Resulting Network after applying IACDS*

Figure 4.18: Topologies obtained after applying the proposed algorithm.

4.3.3 Simulation 2: Changing the Node Density

The main goal of this simulation is to compare the algorithms when the network density is changed by varying the number of nodes in the deployment area while keeping a fixed communication range of 63.

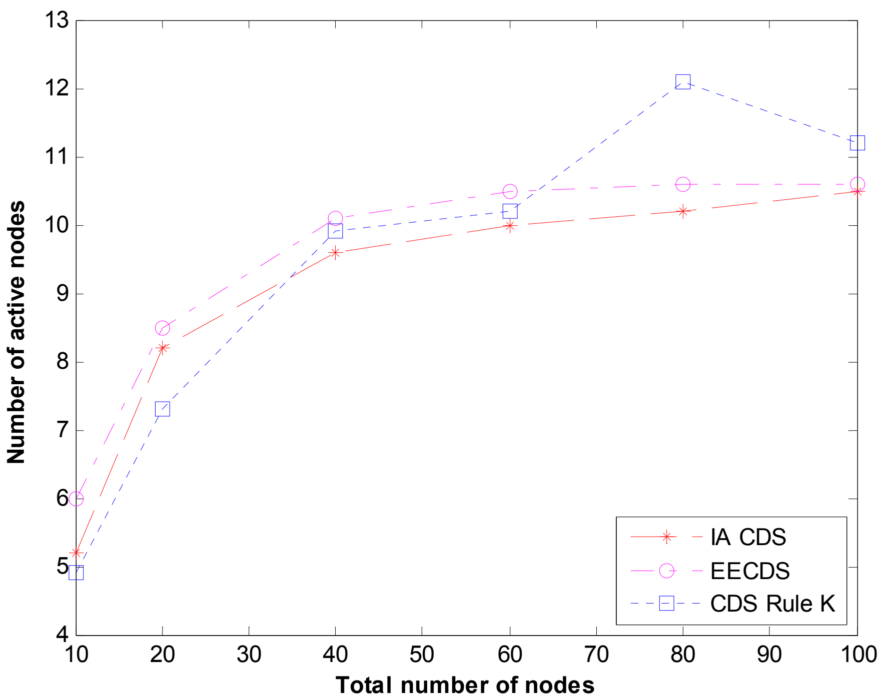


Figure 4.19: Number of active nodes versus the number of nodes in the area.

Communication range of 63 is equivalent in this simulation to 1 × CTR(10). From Equation 4.1:

$$1 \times CTR(10) = 1 \times \sqrt{\frac{ln(10)+ln(ln(10))}{10\pi}} \quad 4.3$$

$$= 1 \times \sqrt{0.0998} \times area\ side \quad 4.4$$

$$= 0.315 \times 200 \quad 4.5$$

$$= 63m \quad 4.6$$

This simulation is important to show how scalable the algorithms are in dense topologies and how the resource usage depends on the number of nodes. The results shown in Figure 4.19 are similar to the ones shown in simulation 1.

Figure 4.19 shows that all algorithms need a similar amount of active nodes, although before 35, CDS-Rule-K shows a small advantage over IACDS, after 35 both EECDS and CDS-Rule-K algorithm go above IACDS. After 60 the CDS-Rule-K algorithm goes up to reach its maximum peak at 80, after 80 it goes down, but still above IACDS algorithm.

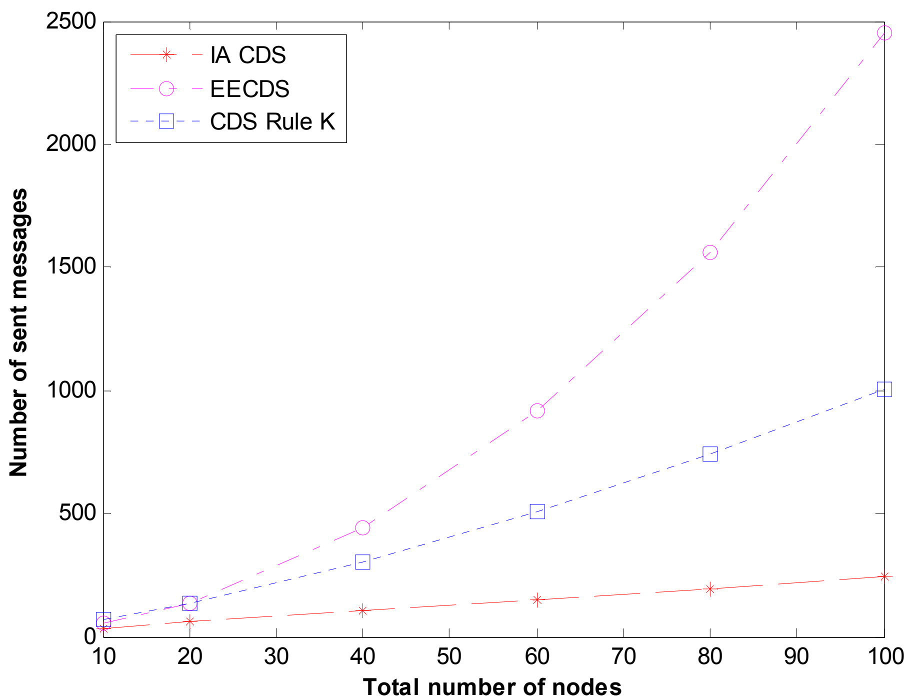


Figure 4.20: Number of sent messages versus the number of nodes in the area.

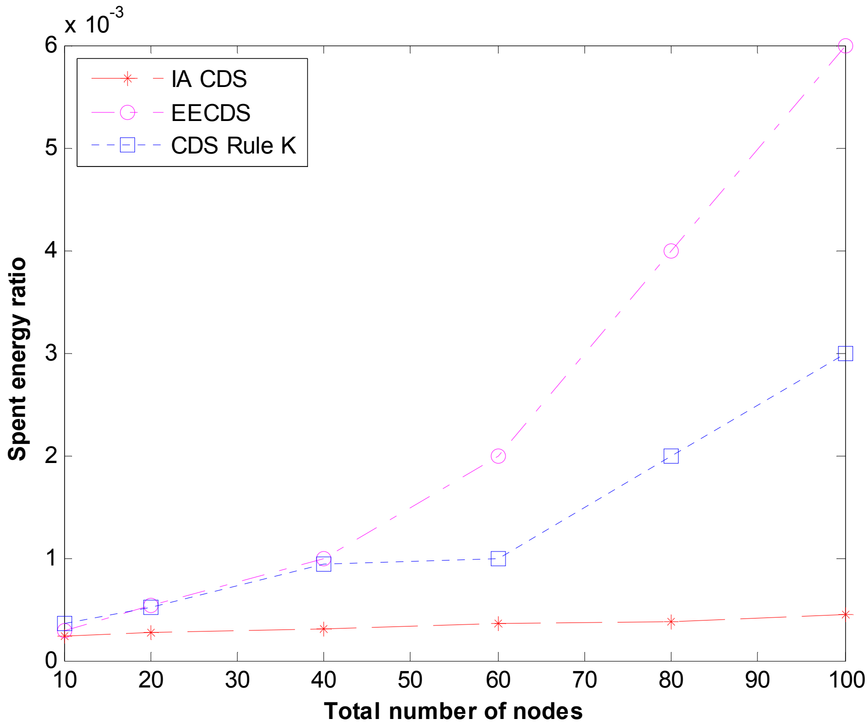
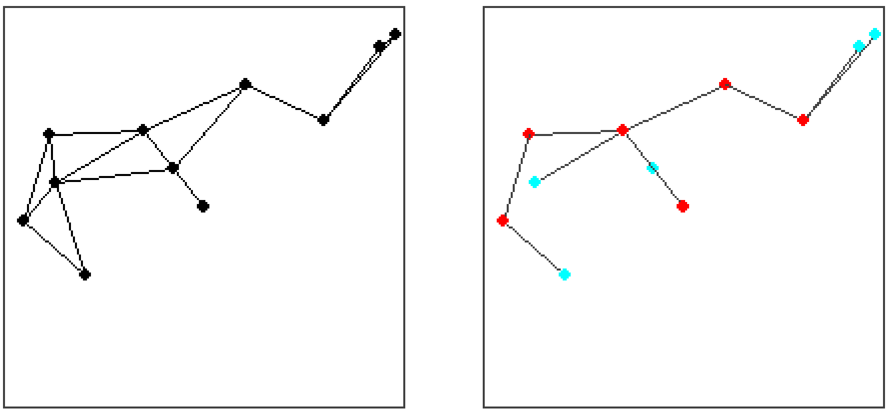


Figure 4.21: Spent energy ratio versus the number of nodes in the area.

Figures 4.20 and 4.21 show that in terms of the message complexity and energy efficiency, the trends are similar. The EECDS and the CDS-Rule-K algorithms present a non-linear increase, while the IACDS algorithm shows a low and linearly bounded number of messages and energy consumption. This shows that the proposed algorithm is scalable and is not highly affected by the number of nodes deployed. Figure 4.22 illustrates the behavior of the proposed interference-aware CDS topology control algorithm, IACDS, in a graphical manner. In this case transmission range is fixed to 63 and the number of nodes is varied.

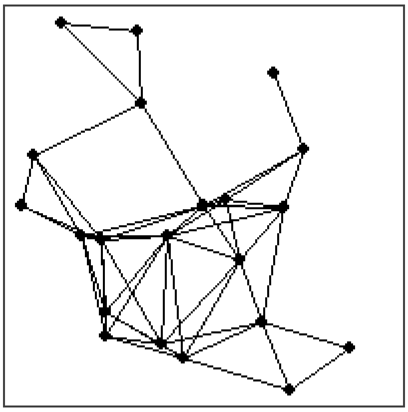
Number of Nodes = 10



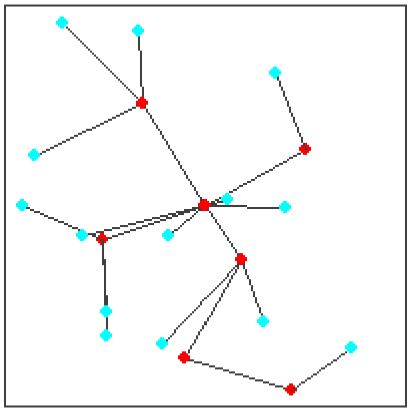
Original network *Resulting Network after applying IACDS*

Figure 4.22: Topologies obtained after applying the proposed algorithm, *cont.*

Number of Nodes = 20

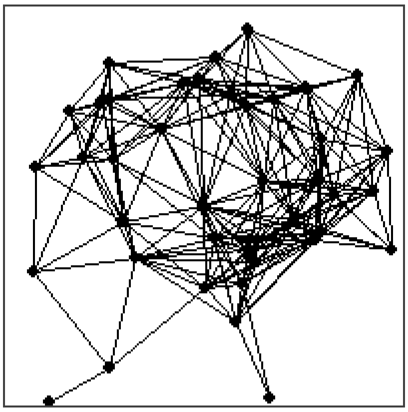


Original network

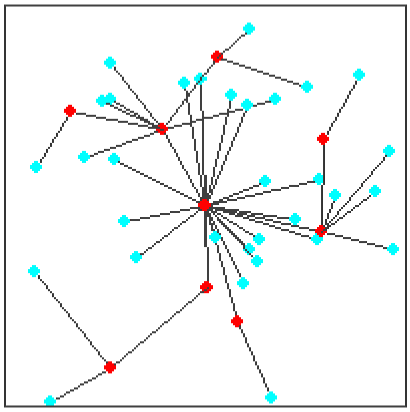


Resulting Network after applying IACDS

Number of Nodes = 40

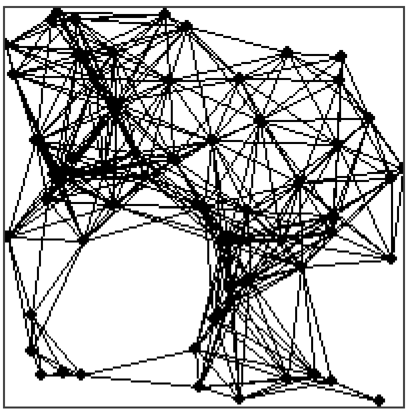


Original network

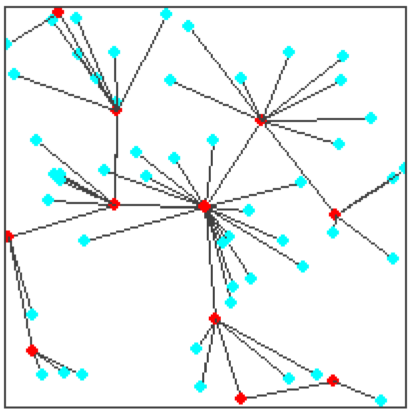


Resulting Network after applying IACDS

Number of Nodes = 60



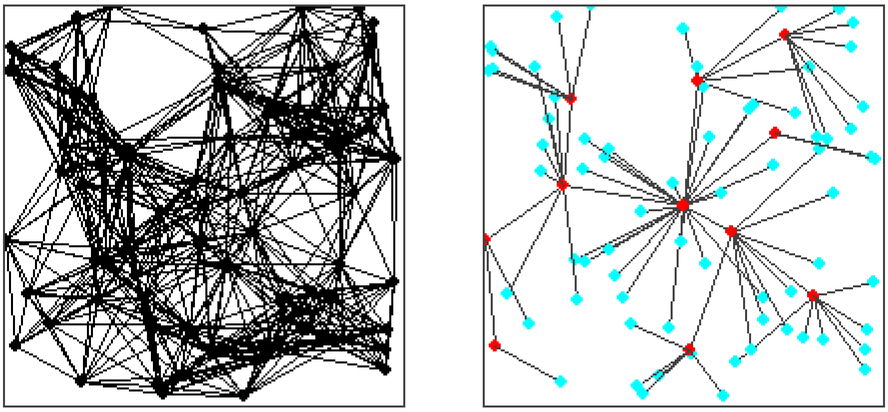
Original network



Resulting Network after applying IACDS

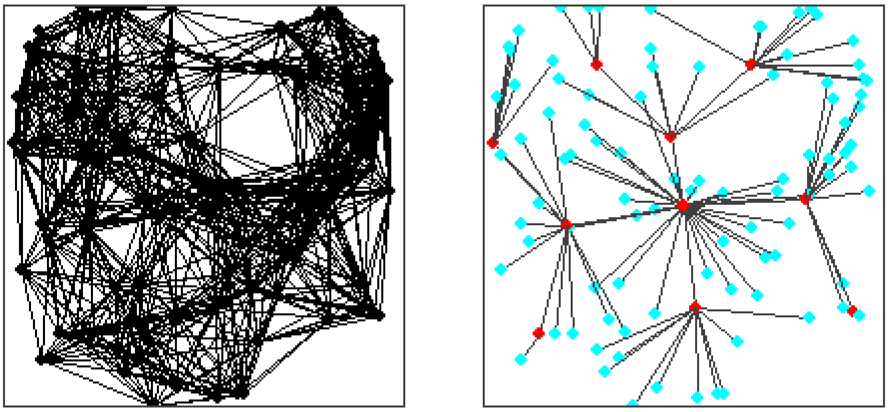
Figure 4.22: Topologies obtained after applying the proposed algorithm, *cont.*

Number of Nodes = 80



Original network *Resulting Network after applying IACDS*

Number of Nodes = 100



Original network *Resulting Network after applying IACDS*

Figure 4.22: Topologies obtained after applying the proposed algorithm.

4.3.4 Simulation 3: Performance using Ideal Grid Topologies

The third simulation considers the ideal grid scenario with two variants of node location distribution: Grid HV and Grid HVD, as shown in Figure 4.23. This simulation shows the performance of the algorithms in a perfectly homogeneous topology, with ideal condition of density and node degree, which could be considered a predefined scenario. From Figure 4.23a, it can be seen that the IACDS algorithm shows similar or better results in the number of active nodes metrics, including 58% of the nodes in the Grid HV and 34% in the Grid HVD scenarios, versus 64% and 41% from EECDS, and 61% and 31% from CDS-Rule-K algorithms. The other two metrics show an increasing trend for EECDS and CDS-Rule-K while IACDS still shows a bounded cost in overhead and energy as seen in Figure 23b and Figure 23c, respectively. Table 4.13 summarizes the parameters that can be defined for a homogeneous family of nodes.

Table 4.13: Grid HV and Grid HVD.

Grid H-V	Distribute nodes in the deployment area with a distance of *communication radius* between nodes, so nodes are adjacent with their vertical and horizontal neighbors
Grid H-V-D	Distribute nodes in the deployment area with a distance of *communication radius* $\times\ \sqrt{2}$ between nodes, so nodes are adjacent with their vertical, horizontal and diagonal neighbors

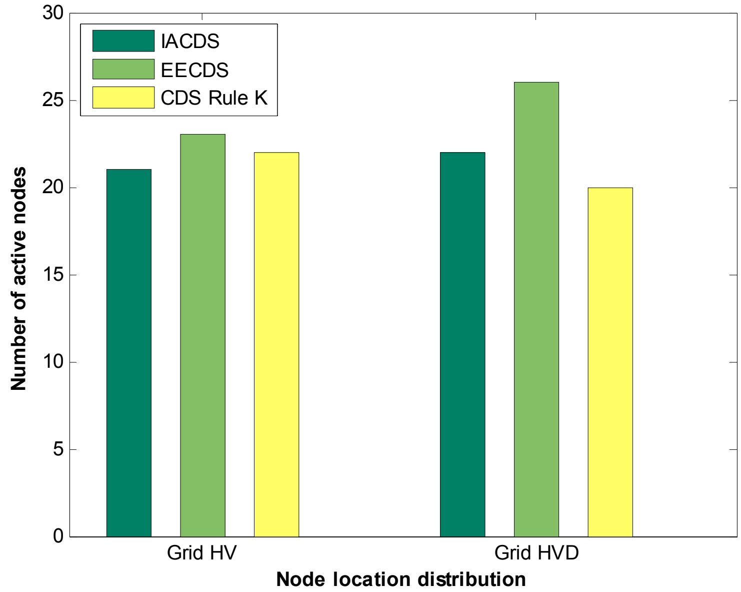


Figure 4.23a: Number of active nodes.

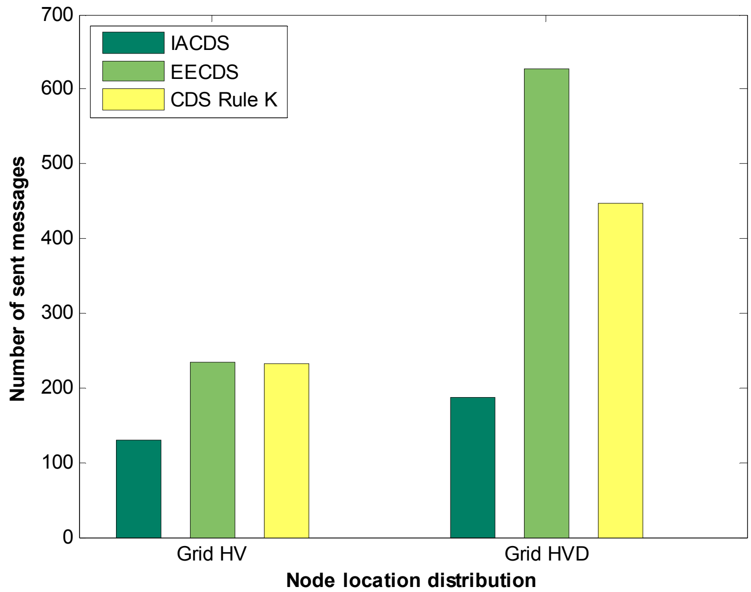


Figure 4.23b: Number of sent messages.

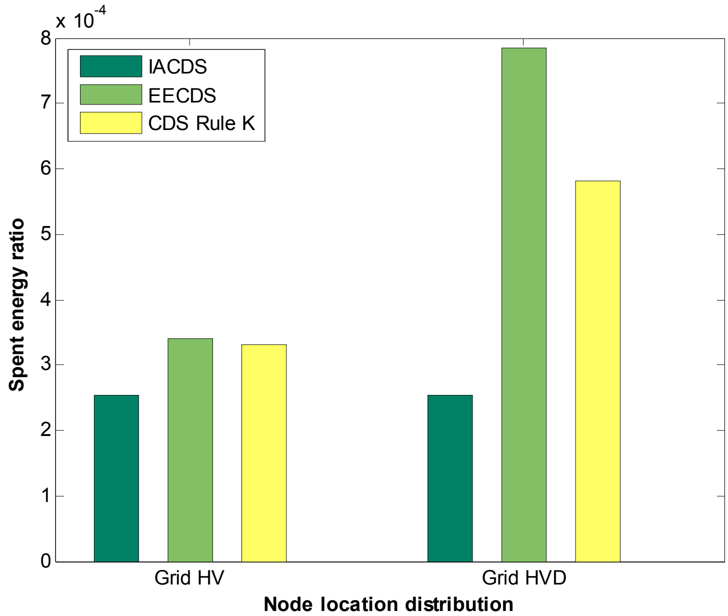


Figure 4.23c: Spent energy ratio in the CDS creation process.

Figure 4.24a shows graphically the behavior of the proposed IACDS algorithm in the case of Grid HV. The number of active nodes is 20 from original 36 nodes. Nodes are distributed in the deployment area with a distance of *communication radius* between nodes; nodes are distributed close to each other. Results show that the number of active nodes is large with respect to the total number of nodes.

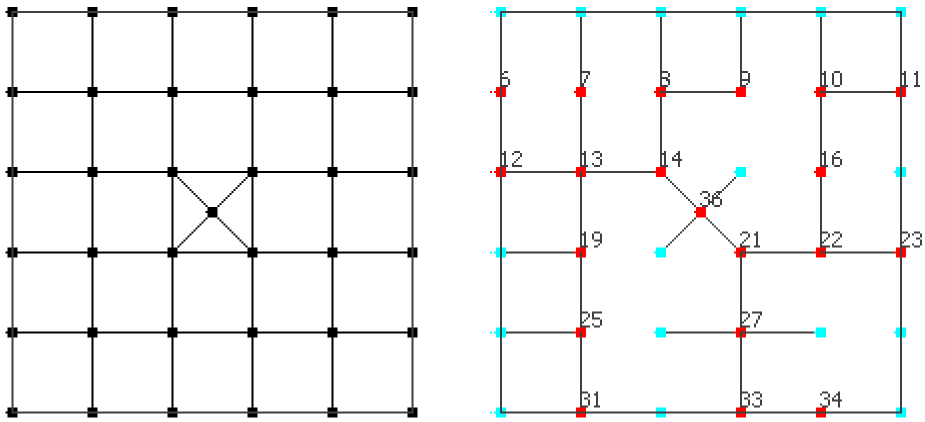


Figure 4.24a: Grid HV node location distribution.

Figure 4.24b shows graphically the behavior of the proposed IACDS algorithm in the case of Grid HVD. The number of active nodes is 21 from original 64 nodes. Nodes are distributed in the deployment area with a distance of *communication radius* $\times \sqrt{2}$ between nodes; nodes are distributed separate from each other. Results show that the number of active nodes is small with respect to the total number of nodes.

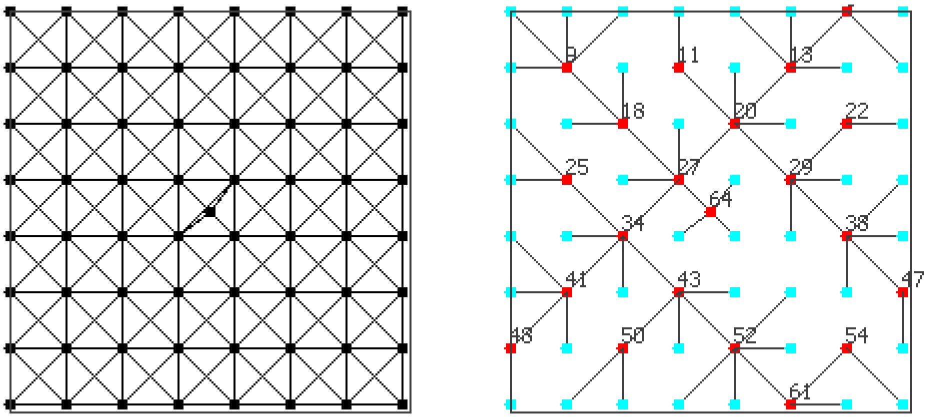


Figure 4.24b: Grid HVD node location distribution.

4.3.5 Area of Communication Coverage

When applying these algorithms, the active nodes determine the communication coverage area. This area is expected to cover as much of the deployment area as possible. Figure 4.25 shows the average communication area covered by the algorithms using the scenarios from Simulation 2. As it can be seen from this figure, although all algorithms produce an almost similar coverage with the selected active nodes, IACDS is still better; it covers the same or more area but using fewer resources than the others.

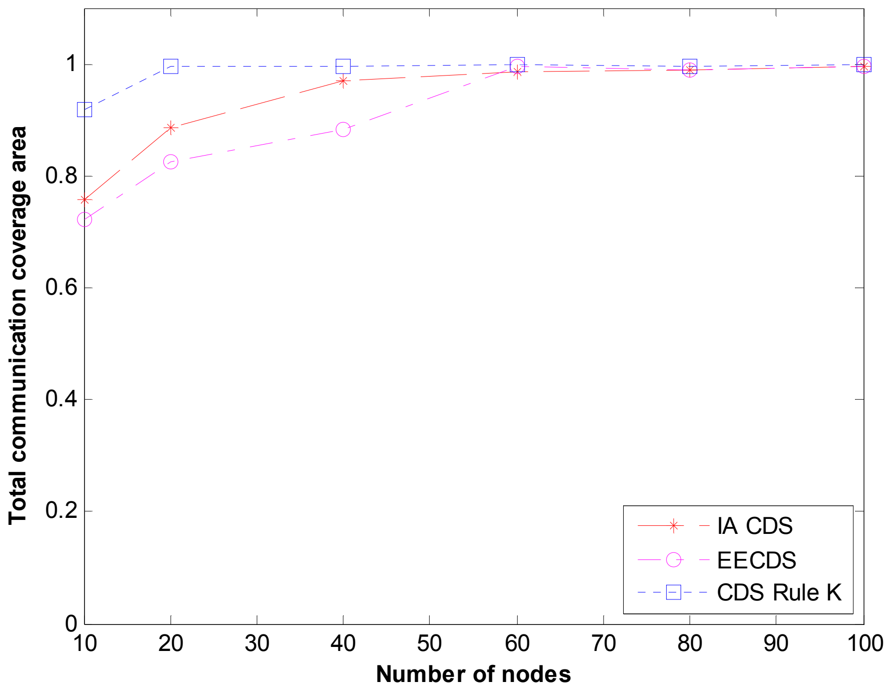


Figure 4.25: Total communication coverage area.

4.4 Topology Comparison and Other Simulation Studies

4.4.1 Simulation Environment and Parameters

In this experiment, extensive simulations have been conducted to study and compare the performance of the proposed interference-aware topology control algorithm, IACDS, with Burkhart's LIFE in [64]. The networks are constructed by distributing nodes randomly in a 30×30 square deployment area. Without loss of generality, the mean result is derived from 50 networks, generated randomly with different number of nodes N and different transmission range TR.

Table 4.14 presents a summary of the simulation parameters used in this experiment.

Table 4.14: Simulation parameters for comparison with LIFE.

Deployment area	30m × 30m
Number of nodes	N: 20, 30, 40, 50, 60, 70, 80
Transmission range	TR: 5, 10, 15
Node distribution	Random (30, 30)
Instances per topology	50 instances
Maximum energy	1 Joule

In order to measure the interference reduction quantitatively, *interference amount reduction ratio* IARR is defined as:

$$IARR = \frac{IA(H1)-IA(H2)}{IA(H2)} \times 100\% \qquad 4.7$$

Where IA(H1) denotes the interference amount of the topology generated by LIFE algorithm, and IA(H2) represents the interference amount of the resulting structure constructed by the proposed IACDS algorithm, with the metric defined in Equation 3.10.

The total interference amount of a network *H* is defined as:

$$IA_{Total}(H) = \sum_{u \in V(H)} IA(u) \qquad 4.8$$

Where from Equation 3.9:

$$IA(u) = \sum_{i=1}^{|INS|} INS_i/|INS| \; * \; max\, INS_{i=1}^{|INS|}$$

4.4.2 Topology Comparison

To compare between the proposed IACDS algorithm and LIFE algorithm, different number of nodes, N, from 20 to 80 with increment 5 has been put and different transmission ranges, TR, from 5 to 15 with increment 5 have been used.

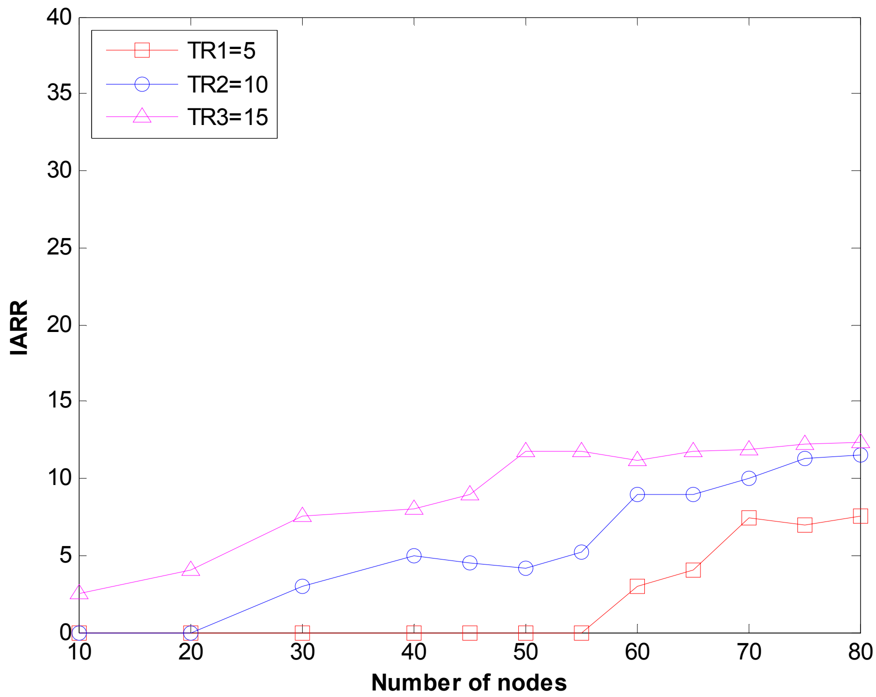


Figure 4.26: Interference amount reduction ratio IARR versus the number of nodes.

Figure 4.26 depicts the interference amount reduction ratio IARR of IACDS TC algorithm in comparison to LIFE algorithm [64]. The results show that IARR is positive, with approximate peak value of 12.3%. It also shows that the IARR is improved as TR and N increase, because the choices of paths between any two nodes of the network also increase. Hence, the proposed mechanism works more efficiently. A critical observation is that when TR equals to 15 and the density of nodes is large, it has little influence on IARR because for such a value TR has great probability to include all the possible neighbors in the resulting topology. In the same way, when N is less than a fixed value, which depends on TR, IARR equals to 0 nearly and the curves are flat, for the reason of fewer edges in the original graph and no guarantee of connectivity. For instance, the value is about 55 in the simulations when TR is 5 and about 29 in the simulations when TR is 20 while in the simulations when TR is 15 this value is below 10.

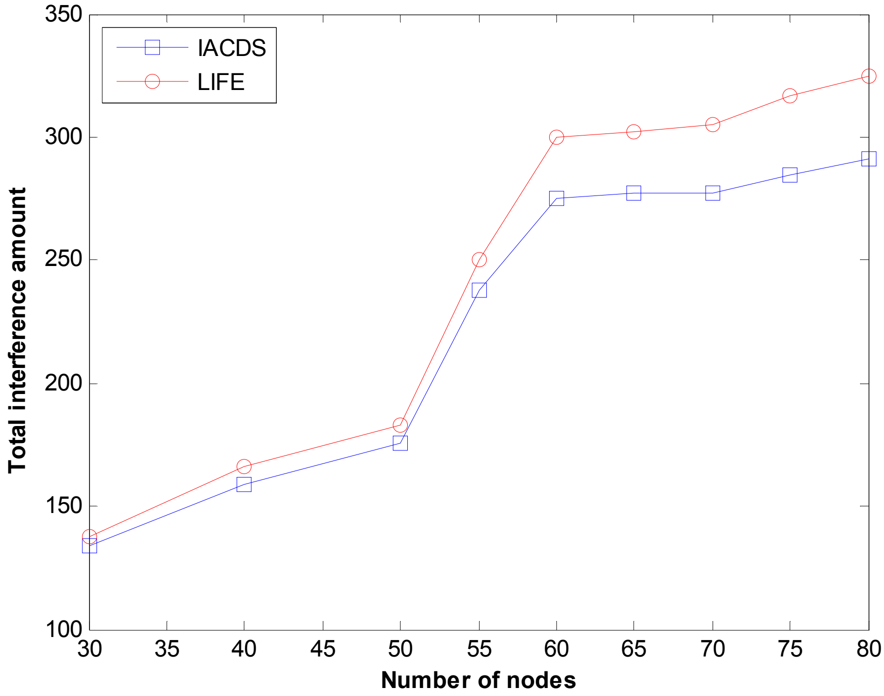
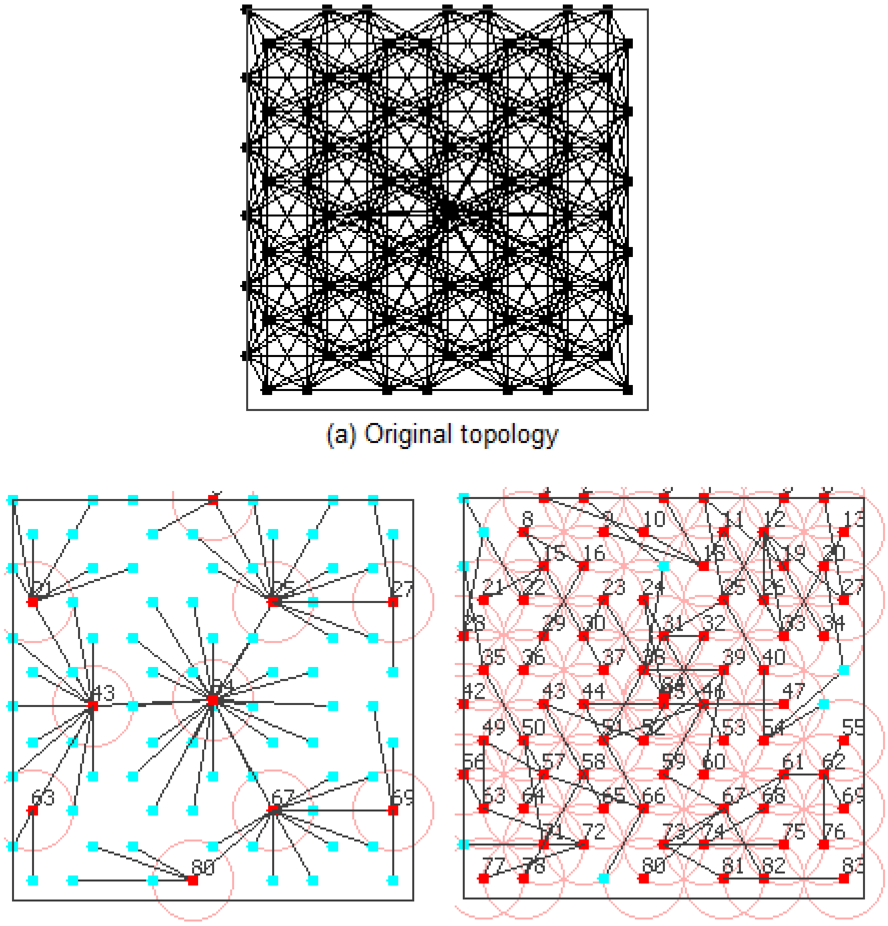


Figure 4.27: Interference amount reduction ratio IARR of IACDS and LIFE.

Figure 4.27 illustrates the performance comparisons of IACDS algorithm and LIFE algorithm simultaneously in terms of total interference amount $IA_{Total}(H)$ with the transmission range TR = 10. We can note that when N is less than 30, meaninglessness occurs due to disconnectivity. These results demonstrate that IACDS algorithm can significantly reduce the interference amount for a given network while the total interference amount is still maintained, far from degradation.

4.5 Difference between Topologies of Resulting Graph

An additional theoretical comparison considering an ideal grid topology in which all nodes have the same number of neighbors is included. Two different grid topologies are used: the square topology, and the triangle topology. The difference between topologies of resulting graph is depicted distinctly in Figure 4.28. Another group is illustrated in Figure 4.29 and Figure 4.30. The red circles have been plotted on vertices to demonstrate the active nodes in the resulting topology. Difference in the process of choosing nodes can be clearly noticed. For example, when node location distribution is *triangle* the IACDS algorithm produces a topology with 11 active nodes, on the other hand, in the case of CDS Rule K algorithm it produces 78 active nodes. An observation is that IACDS algorithm chooses nodes which satisfied the selection metric to avoid more nodes interfered by the communicating neighbors, which also is illustrated in Figure 4.29 and Figure 4.30, in case of square node location distribution and communication radius of 53 and 42, respectively. The number of active nodes that can still reach the sink node - those that still can provide information to the sink - is and important value because if the sink gets isolated, no matter how many active nodes remain, all of them are useless because the information they produce gets lost.

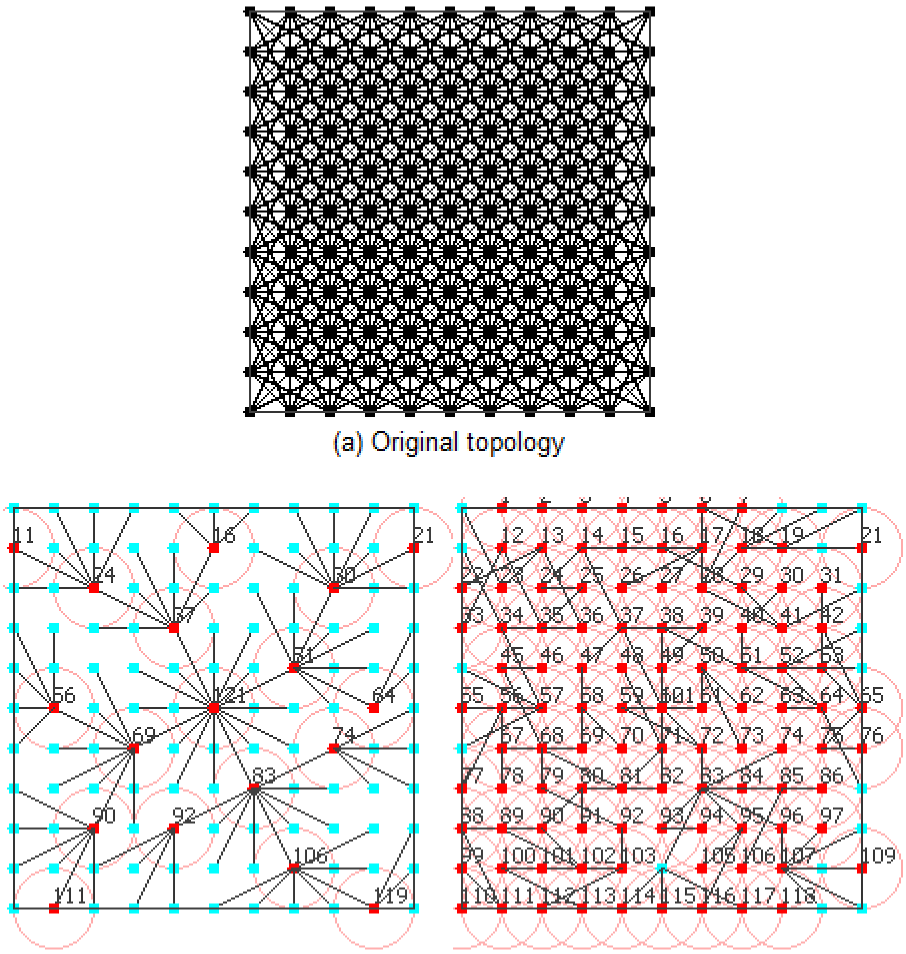


(a) Original topology

(b) Topology generated by IACDS (c) Topology generated by CDS Rule K

Figure 4.28: Topology generated by IACDS and CDS Rule K.

Figure 4.28 shows the different behaviors of IACDS and CDS Rule K algorithms in the case of *triangle* node location distribution, total number of nodes is 85, and communication radius is 63. Difference in the process of choosing nodes can be clearly noticed in the figure. IACDS algorithm produces only 11 reachable active neighbors from the sink; on the other hand, CDS Rule K algorithm produces 78 reachable active neighbors from the sink.



(a) Original topology

(b) Topology generated by IACDS (c) Topology generated by CDS Rule K

Figure 4.29: Topology generated by IACDS and CDS Rule K.

Figure 4.29 shows the different behaviors of IACDS and CDS Rule K algorithms in the case of *square* node location distribution, total number of nodes is 122, and communication radius is 53. Difference in the process of choosing nodes can be clearly noticed in the figure. IACDS algorithm produces only 19 reachable active neighbors from the sink; on the other hand, CDS Rule K algorithm produces 106 reachable active neighbors from the sink.

Figure 4.30 similarly shows the different behaviors of IACDS and CDS Rule K algorithms in the case of *square* node location distribution, total number of nodes is 122, and communication radius is 42. Difference in the process of choosing nodes can be clearly noticed in the figure. IACDS algorithm produces only 27 reachable active neighbors from the sink; on the other hand, CDS Rule K algorithm produces 115 reachable active neighbors from the sink.

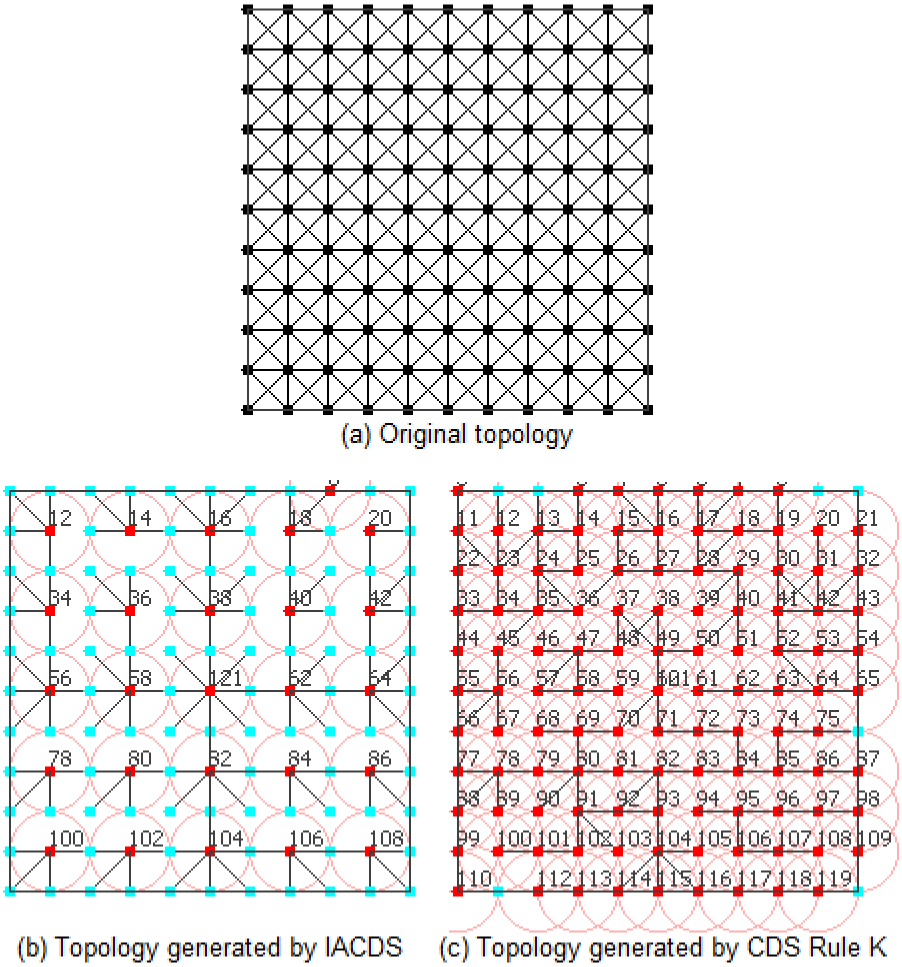


Figure 4.30: Topology generated by IACDS and CDS Rule K.

4.6 Sensitivity Analysis on Topology Maintenance Scheme

The networks are constructed by distributing nodes randomly in a 200 × 200 square area. Simple *sensor & data*, and *routing/forwarding* protocols are used, which are available by the *Atarraya* simulator. Table 4.15 presents a summary of the simulation parameters used in this experiment of sensitivity analysis performed on a TM scheme.

Table 4.15: Simulation parameters for sensitivity analysis performed on TM scheme.

Deployment area	200m × 200m
Number of nodes	100
Transmission range	35 equivalent to: 1.25xCTR(100)
Node distribution	Uniform (200,200)
Maximum energy	1 Joule
Energy threshold	Energy percentage used to invoke energy-triggered topology maintenance protocols. Every time the energy of a node reaches this energy threshold value, since the last invocation of the topology maintenance protocol, the node will invoke the protocol again.

The final experiment shows a sensitivity analysis performed on a topology maintenance scheme, which tests the influence of the energy threshold on the lifetime of the network. Figure 31 shows the number of active nodes over time of the topology maintenance protocol called Dynamic Global Energy Topology Recreation (DGETRec), using 4 different energy thresholds; they are set to 5%, 10%, 25%, and 50% of nodes' remaining capacity. Using this topology maintenance scheme, a new topology construction takes place using the IACDS every time a node reaches the energy threshold set, i.e., when it has consumed 95%, 90%, 75% and 50% of its total energy.

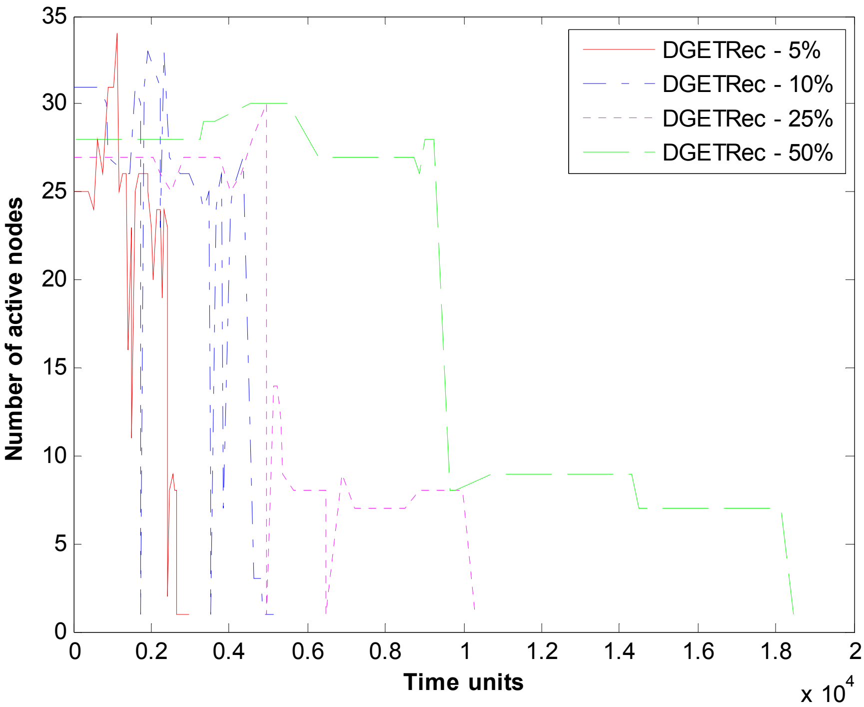


Figure 4.31: A test of sensitivity of DGETRec topology maintenance protocol.

Figure 4.31 shows that increasing energy threshold positively influences the lifetime of the network. In other words, nodes are kept active over time of the topology as longer as possible. This result can be justified. The best case when the threshold is set, for example, to 50% of node's remaining energy. Every time a node consumed half its remaining energy, by means of (DGETRec) topology maintenance scheme, a new topology construction will take place using the IACDS algorithm, which gives the opportunity to change the states of the nodes based on their remaining energy. This opportunity increases as the energy threshold increases, i.e., when the remaining energy is 50% is better than 25%. In the case of 10% threshold value, the number of active nodes is 31 at the beginning of the simulation time. By the work of the sensor & data and routing/forwarding protocols, at 1736.106 simulation time, the number of active nodes reduced sharply. By the invocation of the IACDS topology construction algorithm when the threshold is reached, the number of active nodes increased again. The same scenario is repeated at simulation time 3540.749. These behaviors increase the network life time by keeping the nodes active over the time of the topology. Gradually, at simulation time 4639.417 until the end of the simulation, nodes become dead. This kind of experiments is very useful to determine the best maintenance policy for the proposed topology construction protocol. Figure 4.32 shows the number of active nodes over time of 4 different topology maintenance protocols (DGETRec, SGETRot, DGTTRec, and SGTTRot) when the threshold is set to 25% of nodes' remaining energy.

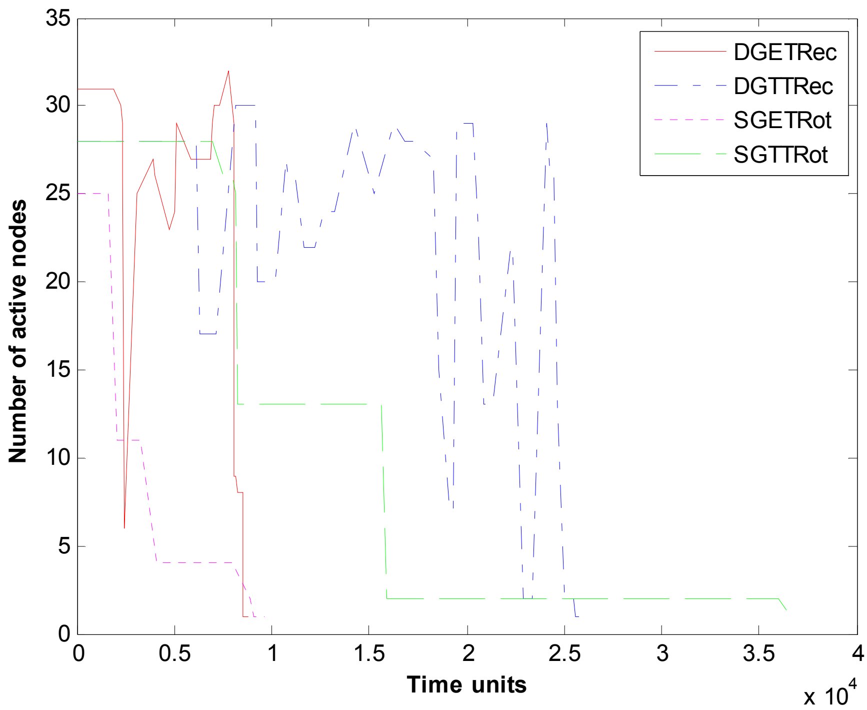


Figure 4.32: A test of sensitivity of topology maintenance protocols.

Clearly seen from Figure 4.32, the best maintenance policy for the proposed IACDS topology construction algorithm is DGTTRec, since it satisfied the larger network lifetime.

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

Due to physical constraints, wireless ad hoc and sensor nodes are primarily powered by exhaustible energy resources like batteries. Consequently, energy is the limiting factor for network lifetime. Great efforts have been made to reduce node energy consumption and thus extend network lifetime. One of the foremost approaches to achieve substantial energy conservation is by minimizing interference between network nodes. Confining interference lowers energy consumption by reducing the number of collisions and consequently packet retransmissions on the media access layer. Topology control draws considerable attentions recently in wireless mobile ad hoc and sensor networks for interference reduction. It is also a well-known strategy to save energy and extend the networks lifetime. The concept of topology control restricts interference by reducing the transmission power levels at the network nodes and cutting off long-range connections in a coordinated way. At the same time, transmission power reduction has to proceed in such a way that the resulting topology preserves connectivity.

In this thesis, such benefits of topology control were exploited in order to reduce the network interference. A new interference computation model was proposed which aims to reflect the interference inducement on the physical layer, and functions with the following properties: creates a relationship between all local parts of the network, and takes into account the maximum interference of the network. This was achieved by mixing the two metrics proposed in Equations 3.2 and 3.8 in a single equation, 3.9. The proposed interference metric (Equation 3.9) was embedded in a selection metric, Equation 3.10, which produces a value between 0 and 1 that is assigned to each neighbor in the process of selecting the new nodes in the CDS; the higher the value of the metric, the higher the priority. As it can be seen from Equation 3.5, the selection metric gives priority to those nodes with minimum interference, higher energy, and which are farther away from the parent node. The final effect of this choice is to have a CDS with minimum interference, fewer active nodes, and better coverage. However, proper weight manipulation can satisfy different criteria, as needed. As the main goal of the proposed algorithm is to reduce the global interference of the network, the interference metric is weighted to its maximum value, 1. If communication coverage is to be optimized and the average height of the node in the CDS (number of hops) needs to be reduced, the distance metric must be weighted more heavily. The downside is that low energy nodes may be included in the CDS, which may introduce early failures of nodes, and therefore reduce the lifetime of the network. On the other hand, if reliability of the tree is desired, energy must be weighted more.

In this thesis, the primary effort has been devoted to propose a new topology construction algorithm, namely, IACDS algorithm, a simple, distributed, interference-aware and energy-efficient topology construction mechanism that finds a sub-optimal Connected Dominating Set (CDS) to turn unnecessary nodes off while keeping the network connected and providing complete communication coverage with minimum interference. IACDS algorithm utilizes a weighted (distance-energy-interference)-based metric that permits the network operator to trade off the lengths of the branches (distance) for the robustness and durability of the CDS (energy and interference).

Through extensive simulation experiments, results show the superiority of the IACDS algorithm compared with the existing alternatives, EECDS and CDS-Rule-K algorithms, in terms of number of active nodes needed, message complexity, and energy efficiency. The results show that IACDS only needs 10.5% and 52% of the nodes active in dense and sparse scenarios while preserving network connectivity and communication coverage, versus 10.6% and 60%, and 11.2% and 49% for the EECDS and CDS-Rule-K algorithms. More importantly, IACDS provides a low linearly bounded number of messages and energy usage, compared with non-linear increasing trends shown by the CDS-Rule-K and EECDS algorithms. This last aspect is extremely important in order to use this algorithm in a complete topology control solution where the CDS network will have to be changed many times. The results via simulation also show that, compared with LIFE algorithm, the proposed IACDS algorithm outperforms on the fact that interference amount can be reduced up, while the total interference doesn't degrade the network. On the other hand, simulations show that the best maintenance policy for the proposed IACDS topology construction algorithm is DGTTRec, since it satisfied the larger network lifetime.

5.2 Future Work

The major goal of topology control in this thesis concerns the tradeoffs between connectivity and interference reduction. There are, however, other aspects that may be related to topology control, which can be investigated.

- Exploiting topology control in terms of optimizing the network based on data traffic and routing. We can combine both and get a more complete picture of the network structure.
- The development of an algorithm which integrates topology control and routing in wireless networks. The algorithm sets up bandwidth-guaranteed paths between nodes when the demands for such paths arrive sequentially and future demands are unknown. This work extends the concept of minimum interference routing to include topology control. Additionally, the proposed algorithm may require determining an approximation ratio to the optimal solution, as a performance metric of the algorithm.

References

[1] C. Perkins, "Ad hoc networking: an introduction," in *Ad Hoc Networking*, Boston, Addison-Wesley Longman Publishing Co., 2001, ch. 1, pp. 1-28.

[2] S. Corson and J. Macker, "Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations," *Network Work. Group RFC 2501*, pp. 2-3, Jan. 1999.

[3] E. Huang, W. Hu, J, Crowcroft, and I. Wassell, "Towards commercial mobile ad hoc network applications: a radio dispatch system," *Proc. ACM MobiHoc*, pp. 355-365, May 2005.

[4] Y. Shang, and S. Cheng "A stable clustering formation in mobile ad hoc network," *Proc. WCNM*, vol. 2, pp. 714-718, Sept. 2005.

[5] N. Shrestha, "Reception-awareness for energy conservation in ad hoc networks," Ph.D. dissertation, Dept. CDICS, MU, Sydney, 2006.

[6] T. Kuang and C. Williamson, "Real media streaming. performance on an IEEE 802.11b wireless LAN," *Proc. of IASTED Wireless and Optical Commun.*, pp. 306-311, July 2002.

[7] S. Teger and D. Walk, "Wireless is magic -- our evaluation of 802.11g Wi-Fi," BBHR, Rep. 0306_3, Jun. 2003.

[8] K. Roemer and F. Mattern, "The design space of wireless sensor networks," *IEEE* Wireless Comm., vol. 11, no. 6, pp. 54-61, Dec. 2004.

[9] D. Westhoff, J. Girao and A. Sarma "On Security Solutions for Wireless Sensor Networks," *NEC J. of Advanced Technology*, vol. 1, no. 3, July 2006.

[10] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Comput. Surv.*, vol. 37, no. 2, pp. 164-194, Jun. 2005.

[11] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," *IEEE Comm. Magazine*, vol. 40, no. 8, pp. 102-114, Aug. 2002.

[12] R. Chadha, Y-H. Cheng, J. Chiang, G. Levin, S-W. Li, and A. Poylisher, "Policy-based mobile ad hoc network management for DRAMA," *Proc. of Military Commun. Conf.*, vol. 3, pp. 1317-1323, Nov. 2004.

[13] I. Chalmtac, M. Conti, J. Liu, "Mobile ad hoc networking: imperatives and challenges," *Journal of Ad hoc Networks*, vol. 1, no. 1, pp. 13-64, Jan. 2003.

[14] C. Grimes, K. Ong, O. Varghese, X. Yang, G. Mor, M. Paulose, E. Dickey, C. Ruan, M. Pishko, J. Kendig, and A. Mason, "A sentinel sensor network for hydrogen sensing," *Sensors*, vol. 3, no. 3, pp. 69-82, Mar. 2003.

[15] R. Cardell-Oliver, K. Smettem, M. Kranz, and K. Mayer, "A reactive soil moisture sensor network: Design and field evaluation," *Int. J. Distributed Sensor Networks*, vol. 1, no. 2, pp. 149-162, 2005.

[16] W. Hu, V. Tran, N. Bulusu, C. Chou, S. Jha and A. Taylor, "The design and evaluation of a hybrid sensor network for cane-toad monitoring," *Proc. of the 4th Int. Symp. on Inform.*, pp. 503-508, Apr. 2005.

[17] S. Hadim and N. Mohamed, "Middleware: middleware challenges and approaches for wireless sensor networks," *IEEE Distributed Sys*t., vol. 7, no. 3, pp. 1-1, Mar. 2006.

[18] G. Pottie, and W. Kaiser, "Wireless integrated network sensors," *Commun. of the ACM*, vol. 43, no. 5, pp. 551–558, May 2000.

[19] S. Vardhan, M. Wilczyncki, G. Pottie, and W. Kaiser, "Wireless integrated network sensors (wins): distributed in situ sensing for mission and flight systems," IEEE *Aerospace Conf.*, vol. 7, pp. 459–463, Aug. 2000.

[20] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Proc. of the 5th Ann. ACM/IEEE Int. Conf. on Mobile Computing and Networking*, Aug. 1999.

[21] L. Shum, "Topology Control and Data Handling in Wireless Sensor Networks," Ph.D. dissertation, Dept. EEE, UCL, London, 2009.

[22] J. Pan, Y. Hou, L. Cai, Y. Shi, and S. Shen, "Topology control for wireless sensor networks," *Proc. of the 9th ann. Int. Conf. on Mobile Computing and Networking*, pp. 286-299, 2003.

[23] K. Gabriel and R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, vol. 18, no. 3, pp. 259-278, Sept. 1969.

[24] J. Jaromczyk, G. Toussaint, "Relative neighborhood graphs and their relatives," *Proc. of the IEEE*, vol. 80, no. 9, pp. 1502–1517, Sep. 1992.

[25] B. Chazelle, R. Rubinfeld, and L. Trevisan, "Approximating the minimum spanning tree weight in sublinear time," *SIAM J. Computing*, vol. 34, no. 6, pp. 1370-1379, 2005.

[26] A. Yao, "On constructing minimum spanning trees in k-dimensional spaces and related problems," *SIAM J. on Computing*, vol. 11, no. 4, pp. 721-736, 1982.

[27] D. Eppstein, M. Paterson, F. Yao, "On nearest-neighbor graphs," *Discrete and Computational Geometry*, vol. 17 no.3, pp. 263–282, 1997.

[28] L. Li, J. Halpern, P. Bahl, Y. Wang and R. Wattenhofer, "A cone-based distributed topology-control algorithm for wireless multi-hop networks," *IEEE/ACM Trans. on Networking*, vol. 13, no. 1, pp. 147-159, Feb. 2005.

[29] D. Blough, M. Leoncini, G. Resta and P. Santi, "The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 5, no. 9, pp. 1267-1282, Sept. 2006.

[30] R. Wattenhofer and A. Zollinger, "XTC: A practical topology control algorithm for ad hoc networks," *Proc. of the 4th Int. Workshop on Algorithms for Wireless Mobile Ad Hoc and Sensor Networks*, pp. 26-30, Apr. 2004.

[31] S. Narayanaswamy, V. Kawadia, R. Sreenivas, and P. Kumar, "Power control in ad-hoc networks: theory, architecture, algorithm and implementation of the COMPOW protocol," *Proc. of European Wireless*, pp. 156–62, Feb. 2002.

[32] Z. Yuanyuan, X. Jia, and H. Yanxiang, "Energy efficient distributed connected dominating sets construction in wireless sensor networks," *Proc. of the Int. Conf. on Wireless Commun. and Mobile Computing*, pp. 797-802, July 2006.

[33] J. Wu, M. Cardei, F. Dai, and S. Yang, "Extended dominating set and its applications in ad hoc networks using cooperative communication," I*EEE Trans. Parallel Distributed Syst.*, vol. 17, no. 8, pp. 851-864, Aug. 2006.

[34] M. Handy, M. Haase, and D. Timmermann,. "Low energy adaptive clustering hierarchywith deterministic cluster-head selection," *Proc. of the IEEE Int. Conf. on Mobile and Wireless Commun. Networks*, vol. 12, pp. 368-372, Sept. 2002.

[35] O. Younis, S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. on Mobile Computing*, vol. 3, no. 4, pp. 366 – 379, Dec. 2004.

[36] M. Labrador and P. Wightman "Topology maintenance dynamic techniques," in *Topology Control in Wireless Sensor Networks*, Florida, Springer, 2009, ch 12, sec.2, pp. 141-152.

[37] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithm," *IEEE J. on Select. Areas in Commun.*, vol. 17, no. 8, pp. 1454–1465, Aug. 1999.

[38] P. Santi, "Topology control," in *Topology Control in Wireless Ad Hoc and Sensor Networks*, Chichester, UK: John Wiley & Sons, 2005, ch. 1, sec. 3.

[39] L. Li, J. Halpern, P. Bahl, Y. Wang, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," *Proc. of Ann. ACM Symp. on Principles of Distributed Computing*, pp. 264–273, Aug. 2001.

[40] V. Kumar, T. Arunan, and N. Balakrishnan, "E-span: enhanced-span with directional antenna," *Proc. of IEEE Conf. on Convergent Technologies for Asia-Pacific Region*, vol. 2, pp. 675–679, 2002.

[41] N. Li, J. Hou, and L. Sha, "Design and analysis of an mst-based topology control algorithm," *Proc. of IEEE INFOCOM*, vol. 3, pp. 1702–1712, Apr. 2003.

[42] S. Butenko, X. Cheng, C. Oliveira, and P. Pardalos, "A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks," *Kluwer Academic Publishers*, pp. 61–73, 2004.

[43] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *ACM Wireless Networks*, vol. 8, no. 5, pp. 481–494, Sept. 2002.

[44] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, Apr. 1998.

[45] J. Wu, M. Cardei, F. Dai, and S. Yang, "Extended dominating set and its applications in ad hoc networks using cooperative communication," *IEEE Trans. Parallel Distributed Syst.*, vol. 17, no. 8, pp. 851-864, Aug. 2006.

[46] J. Wu and F. Dai, "An extended localized algorithm for connected dominating set formation in ad hoc wireless networks," *IEEE Trans. on Parallel and Distributed Syst.*, vol. 15, no. 10, pp. 908–920, Oct. 2004.

[47] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," *Proc. of the 3rd ACM int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Commun.*, pp. 7–14, Aug. 1999.

[48] Z. Yuanyuan, X. Jia, and H. Yanxiang, "Energy efficient distributed connected dominating sets construction in wireless sensor networks," *Proc. of the Int. Con. on Wireless Commun. and Mobile Computing*, pp. 797-802, July 2006.

[49] C. Shen, C. Srisathatpornphat, R. Liu, Z. Huang, C. Jaikaeo and E. Lloyd, "CLTC: A cluster-based topology control framework for ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 3, no. 1, pp. 18-32, 2004.

[50] M. Burkhart, P.von Richenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference?," *Proc. of the 5th ACM Int. Symp. on Mobile Ad hoc Networking and Computing*, pp. 9-19, May 2004.

[51] K. Moaveni-Nejad, X. Li, "Low-interference topology control for wireless ad hoc networks," *Ad Hoc & Sensor Wireless Networks*, vol. 1, no. 1-2, pp. 41-64, 2005.

[52] P.von Richenband, S. Schmid, R. Wattenhofer, and A. Zollinger, "A robust interference model for wireless ad-hoc networks," *Proc. of IPDPS'05*, vol. 13, pp. 239, Apr. 2005.

[53] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *J. of Ad Hoc Networks*, vol. 1, no. 1, pp. 107-123, July 2003.

[54] F. Meyer auf de Heide, C. Schindelhauer, K. Volbert, and M. Gruenewald, "Energy, congestion and dilation in radio networks," *Proc. of the 14th Ann. ACM Symp. on Parallel Algorithms and Architectures*, pp. 230-237, 2002.

[55] [54] T. Hou and V. Li, "Transmission range control in multihop packet radio networks," *IEEE Trans. on Commun.*, vol. 34, no. 1, pp. 38-44, 1986.

[56] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Trans. on Commun.*, vol. 32, no. 3, pp. 246-257, 1984.

[57] R. Prakash, "Unidirectional links prove costly in wireless ad-hoc networks," *Proc. of the 3rd Int. Workshop on Discreet Algorithms and Methods for Mobile Computing and Commun.*, pp. 15-22, 1999.

[58] X. Zhang, Q. Liu, D. Shi, Y. Liu, and X. Yu, "An average link interference-aware routing protocol for mobile ad hoc networks," *Proc. of 3rd Int. Conf. on Wireless and Mobile Commun.*, Mar. 2007.

[59] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *J. of Ad Hoc Networks*, vol. 1, no. 1, pp. 107-123, July 2003.

[60] P. Wightman and M. Labrador, "A3: A topology control algorithm for wireless sensor networks," *IEEE Globecom 2008*, Nov. 2008.

[61] P. Wightman and M. Labrador, "Atarraya: a simulation tool to teach and research topology control algorithms for wireless sensor networks," ICST 2nd Int. Conf. on Simulation Tools and Techniques, Feb. 2009.

[62] M. Penrose, "The longest edge of a random minimal spanning tree," *The Ann. of App. Probability*, vol. 7, no. 2, pp. 340–361, 1997.

[63] P. Santi, " The CTR for connectivity: mobile networks," *in Topology Control in* Wireless Ad Hoc and Sensor Networks, Chichester, UK: John Wiley & Sons, 2005, ch. 2, sec. 5.

[64] M. Burkhart, P.von Richenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference?," Proc. of the 5th ACM Int. Symp. on Mobile Ad hoc Networking and Computing, pp. 9–19, May 2004.