Distributed Algorithms for Initialization & Topology Control in Wireless Ad Hoc Networks

Doctoral dissertation submitted by

Subhasis Bhattacharjee

for award of the Ph.D. degree of the

Indian Statistical Institute, Kolkata

Advisor :

Professor Nabanita Das

Advanced Computing and Microelectronics Unit
Indian Statistical Institute
203, B. T. Road, Kolkata, India 700 108
July 2010

Abstract

Wireless ad hoc networking is an upcoming communication technology that makes exchange of information possible without any pre-existing infrastructure. Over the last decade it has grabbed tremendous interest in the research community due to its easy deployability and high flexibility, with numerous applications to social, industrial and personal uses. In this thesis, we designed efficient light weight distributed algorithms based on minimal local information to resolve the problems related to the initialization and topology configuration of wireless ad hoc networks with special emphasis on optimal utilization of limited resources.

Once the ad hoc nodes with in-built radio transceivers are deployed the media access control (MAC) problem is to be resolved first for collision-free message exchange. Here, we studied the Time Division Multiple Access (TDMA) protocol for low-power and guaranteed access of the medium. For any arbitrary distribution of nodes, determining the minimum TDMA frame length is *NP-hard*. Moreover, scheduling a time slot to each node within the given frame length in a purely distributed fashion is much harder. A new upper bound is established on the TDMA frame length for any random distribution of nodes on a two-dimensional region. We proved the fact that this upper bound is of linear order of the maximum node degree. Distributed algorithms are presented for single and multiple slot assignments followed by a scheme for rearranging the slots to adapt with limited perturbations in the network topology.

Next, we investigated the problem of constructing connected dominating set (CDS) as a routing backbone for infrastructure-less ad hoc networks. However, finding the minimum connected dominating set is an *NP-hard* problem. A distributed algorithm is proposed for computing a CDS which terminates in constant number of rounds and outperforms several well-known heuristics in terms of runtime and the quality of results. Similarly, for Bluetooth networks, we proposed a distributed energy efficient algorithm for constructing scatternet as a virtual routing backbone which reduces the number of piconets and bridges in the network. The algorithm imposes a novel synchronization technique, compatible with the original Bluetooth specification, that enables nodes to save significant amount of energy during the procedure.

It is observed that adjustments of transmission power at nodes play a crucial role in improving the network lifetime and reducing the effect of interference. The connectivity preserving optimal power assignment problem in an arbitrary network is *NP-hard*. We investigated the idea of node based transmission power control that maintains the connectivity among the nodes and builds up a network topology for energy-efficient communication. Two distributed algorithms are proposed for two distinct situations, one for mobility-prone networks, and the other for lifetime-critical networks with less mobile nodes. The first algorithm completes transmission power assignment in two rounds of message passing only. The second algorithm takes as many rounds as the most efficient distributed minimum spanning tree algorithm, but it outperforms the latter in terms of node power levels. An interesting observation is to note that the power control reduces node degree, by releasing some links in the network, which in turn reduces the computational time of many algorithms addressed for ad hoc networks, where the time complexity depends on node degree.

Finally, we studied an interesting problem of data gathering in multi-hop wireless sensor networks. Assuming a uniform traffic pattern, we presented a distributed greedy algorithm for extracting a rooted spanning tree to gather sensed data from individual sensors to the sink node to improve the network lifetime.

The most important feature of the proposed distributed algorithms is that each node can execute those with a minimal knowledge about its two-hop neighbourhood which it can gather easily and rapidly using limited number (2 or 3) of communication rounds.

We supplement our theoretical study of the distributed algorithms by extensive simulation studies. Comparison with earlier works showed that our algorithms perform better in terms of various appropriate performance metrics.

Our study enabled us to build up an intricate knowledge about the interdependency of different cross-layer issues, like the media access control, routing backbone design, topology control and energy-efficient data routing in mobile ad hoc networks in general, with some studies on special cases of ad hoc networks, namely the Bluetooth networks and wireless sensor networks. Hence, we hope this work will throw new light on the initialization and self-configuration of these infrastructure-less ad hoc networks, and can help its design for various applications.

Dedicated to

my parents

Acknowledgement

I am profoundly grateful to my supervisor Professor Nabanita Das for her constant motivation, guidance and advice, patience and untiring efforts throughout this work. My words are inadequate to convey my gratitude towards her.

I would like to convey my gratitude towards Professor Jayasree Dattagupta and Professor Subhas C. Nandy who inspired me since my earlier days in this institute. Their cooperation and patience helped me to concentrate in my research to a great extent.

I would like to thank Professor Bhabani P. Sinha and Professor Bhargab B. Bhattacharya who greatly enriched my knowledge in various fields of Computer Science with their exceptional insights. I must also mention the encouragement, advice and support I received from Professor Susmita Sur-Kolay, Dr. Krishnendu Mukhopadhaya, Dr. Sandip Das, Dr. Srabani Mukhopadhyaya, Professor Pradip K. Srimani and Professor Dhiraj K. Pradhan. I would like to thank all of them.

I cherished my stay in the institute largely owing to my friends. I thank Pritha, Arijit, Sasthi, Partha, Gautam, Debasis, Sandeep, Arindam, Ritwik, Promita, Mana and many others for their encouragement and delightful company. I am also thankful to Sriram, Tarun, Ranjan, Vipin and many others of Synopsys (India) Pvt. Ltd. for providing me enough opportunities to concentrate on writing this thesis. My special thanks go to my friend Prateek for his contribution and support in successful completion of this thesis. I feel a deep sense of gratitude for my parents and my elder brother for motivating me to pursue this research work.

I would like to thank Indian Statistical Institute, Kolkata for providing me a good research environment, the opportunity to get acquainted with distinguished researchers and financial support during my tenure as a research scholar. My thanks go to all the members of the faculty and office staff of this institute and especially to the members of Advanced Computing and Microelectronics Unit for making this unit and the institute a cordial and wonderful place to work.

(Subhasis Bhattacharjee)

June 27, 2010

Advanced Computing and Microelectronics Unit,

Indian Statistical Institute,

203 B T Road, Kolkata

Pin 700 108, India

Contents

List of Figures

Chapter 1

Introduction

Recent emergence of low-power inexpensive communication and computing devices has resulted in an exponential growth in mobile wireless communication. In traditional wireless networks, such as cellular networks and wireless LAN's, mobile terminals communicate via wired infrastructure of base stations. Recently, researchers are motivated to explore the more challenging scenarios where mobile communication units can exchange messages without the support of any pre-existing backbone. The term *ad hoc network* refers precisely to this type of network: a wireless network of mostly mobile devices that can be set up dynamically without the support of any fixed infrastructure. Using this technology, various portable devices like cellular phones, PDAs, laptops, i-pads, pagers, etc. can be connected together anywhere and at anytime, and eventually via intermediate devices they may access fixed base stations or wireless access points to form an ubiquitous network. Ad hoc networking, the ultimate frontier in wireless communication, has enormous potential to support wide range of applications such as mobile conferencing, search and rescue operations, battlefield communications, home networking, emergency services, Bluetooth networks, sensor networks, etc.

1.1 Overview

Basically, each node in an ad hoc network contains a radio unit, with wireless module capable of transmitting and receiving within a specific range. When one node transmits, all other nodes within its range can listen to it. Thus, in the simplest scenario, two nodes can communicate directly when they are within the wireless transmission range of each other. However, ad hoc networks also support communication between nodes that are not directly connected, but can be made connected by a series of wireless hops through other nodes. So, a communication path can be realized either through a single-hop radio transmission, or by multi-hop relay via some intermediate nodes.

Typically, a set of mobile nodes come together for a period of time to exchange information. The nodes may continue to move while exchanging information [118, 2]. It is obvious that the underlying network topology will change frequently and unpredictably due to this arbitrary node mobility. Two nodes sharing a link at this moment, may move in such a way that they are no longer within the transmission range of each other, and therefore having no direct connection at the next moment. Moreover, even for static nodes, the transmission range can be varied by controlling the transmission power of individual nodes changing the network topology. Hence, the nodes must have the capacity to re-organize themselves into a network *on the fly* without any external support.

In this work, besides the general ad hoc networks we have also considered two special types of it, namely the Bluetooth networks and the wireless sensor networks, each with some special features and characteristics.

Bluetooth technology is used to meet the ever-growing demand for wireless connectivity among consumer electronic gadgets such as cell phones, handsets, PDAs, digital cameras, laptop computers, etc. in a personal area network [9, 10]. In Today's world almost every electronic device comes with a Bluetooth chip attached to it that can connect it with other Bluetooth enabled devices without any wire. It is a very common scenario that in an apartment complex of fifty residential apartments there are more than hundreds of Bluetooth enabled devices.

Another special type of ad hoc network that created enormous interest in the research community during the last decade is the *Wireless Sensor Network* (WSN) [4]. In WSN, the nodes are tiny smart sensors with advanced sensing capability (thermal, pressure, acoustic, chemical, and many more) equipped with a small processor and a short-range wireless transceiver. Thousands of such inexpensive nodes may be deployed [175] in harsh and inhospitable areas to collect ground data, and then through multihop paths the information may be transmitted to a central location, called *sink* for further processing and interpretation leading to better understanding of the phenomenon under consideration. A US National Research Council report titled *Embedded Everywhere* notes that the use of such networks throughout society "could well dwarf previous milestones in the information revolution" [88].

In ad hoc networks, the initializations refer to the procedures required to be accomplished by the nodes to bring up the network to a state so that the required message exchanges are possible. The topology control refers to the process of transforming the underlying network to make efficient use of the limited resources, like

bandwidth, battery-power, etc., to accomplish the required objectives. The process of the initialization and the topology control, when done in a distributed way, can be called self-organization among the nodes.

Though the absence of any infrastructure makes ad hoc networks attractive, at the same time it poses additional challenges over and above the traditional wireless communication problems [38, 46, 127]. The ad hoc nodes are in general battery-powered, and hence, energy is the most scarce resource that is mainly drained in communication and computation. Since the nodes are mobile, and also can control the transmission power, the network topology varies dynamically. Moreover, the absence of any centralized control poses additional difficulty in running centralized algorithms with global knowledge for initializing and operating such networks. As a consequence, algorithms for ad hoc networks should use: i) as little communication as possible, and ii) as little computation as possible.

The distributed algorithms running on individual nodes are one among the feasible solutions for initialization and topology control in ad hoc networks. Various goals can be achieved by developing the so-called light-weight *localized algorithms* that use only local information and computes within a small number of communication rounds.

Under these limitations, this thesis looks at the problem that given a distribution of ad hoc nodes, how they can run localized distributed algorithms on individual nodes to determine media access control, routing decisions, and topology control to build up an energy-efficient self-organized network as a whole. We keep our discussion limited to the domain of initialization and topology control in ad hoc networks.

1.2 Initialization and Topology Control: Associated Challenges

As initialization is likely to be the most fundamental requirement for any network, we start our discussion with various problems associated with the initialization processes [24]. Here, we discuss various issues related to the medium access control [11] and routing [138] which are primary requirements for message exchange. The discussion continues further to highlight various issues related to the use of the limited resources in ad hoc networks with special emphasis on energy consumption and operational lifetime for general ad hoc networks and special purpose ad hoc networks

like Bluetooth and WSN [19].

The following subsections attempt to present a brief overview of the challenges related to the design issues considered in this work.

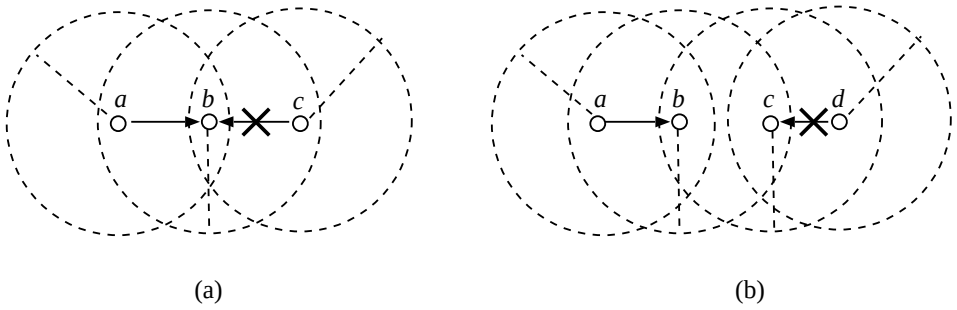
1.2.1 Medium Access Control Problem

Wireless ad hoc networks usually operate over a single shared communication channel. Hence, it is obvious that it requires a medium access control (MAC) to allow devices to access the channel to reduce collision and hence retransmission.

The traditional Medium Access Control (MAC) protocols can be classified as contention-based or contention-free depending on the access strategy [92]. A collision is said to occur when two or more nodes send data over a single channel at the same time. The contention-based schemes, such as ALOHA, slotted-ALOHA, CSMA and its different variations, are more natural, but are appropriate for sporadic data transfer only. They use various collision detection mechanisms and incorporate various methods to reduce the probability of collision. On the contrary, the contention-free mechanisms are normally employed to provide bounded end-to-end delay and minimum bandwidth, privileging delay sensitive applications such as audio and video streams [179]. They employ certain techniques which ensure that only one node sends data at a time and thus collisions never occur.

In wired networks, the sender can easily detect collision by transmitting and sensing the channel simultaneously. However, in free space the collision detection at the sender does not make any sense. No collision at the sender does not guarantee a successful reception. Thus, a successful reception is not guaranteed because senders may not sense a possible collision. This phenomenon known as the *hidden terminal problem* is depicted in Figure 1.1(a); the dotted circle around a node represents its transmission range. For example, when the node a is transmitting data to node b, as node c cannot sense the ongoing transmission, it may also start transmitting data to b, resulting in a collision at b. Here, nodes a and c are *hidden* from each other.

To cope with the hidden terminal problem in wireless networks, the Multiple Access with Collision Avoidance (MACA) protocol employs a three-way handshaking through RTS (Request-To-Send)-CTS (Clear-To-Send) control packets [83]. Here, the medium is reserved for a transmission proactively avoiding collisions. If the CTS is not received after an RTS transmission, a collision is inferred and the sender usually performs some collision resolving mechanism, such as *binary exponential*



(a) (b)

Figure 1.1: The wireless MAC issues: (a) hidden terminal problem and (b) exposed terminal problem

back-off. However, MACA does not completely solve the hidden terminal problem and collision may occur between control and data frames. The MACA for Wireless LAN (MACAW) protocol [17] uses the acknowledgment for data frames to recover the lost frames immediately. Although the RTS/CTS mechanism avoids the hidden terminal problem to some extent, it may accentuate another typical problem of ad hoc networks, the *exposed terminal problem*. As shown in Figure 1.1(b), when nodes a and b reserves the medium for a transmission from a to b, node d cannot transmit to c even if this transmission does not interfere at destination node b.

Many direct or hybrid extensions to MACA or CSMA have been proposed to solve the hidden and exposed terminal problems to a greater extent [59, 115, 152, 161]. The IEEE 802.11 [77], standardizes the CSMA/CA (CSMA with Collision Avoidance) protocol that combines characteristics of CSMA, MACA/MACAW, and FAMA. It specifies two medium access algorithms: Distributed Coordination Function (DCF) and Point Coordination Function (PCF). DCF [20] is a distributed mechanism which uses CSMA/CA. On the other hand, PCF, targeted for infrastructured networks, is a centralized mechanism where an access point controls medium access. However, the primary objective of all these protocols is to reduce the probability of collision rather than totally eliminating them.

It is clear from the above discussions that in case of a single shared medium, none of the above protocols can guarantee totally contention-free access. In ad hoc networks, collision and hence retransmission may waste energy, the most scarce resource in the devices. Moreover, the difficulty of collision detection and the hidden and the exposed terminal problems demand new distributed collision avoidance or contention-free protocols for MAC. The contention-free mechanisms essentially divide the single medium, based on certain dimensions, into several virtual media

which can be accessed independently, hence without contention. The various dimensions, commonly used to divide the medium, are time (in case of Time Division Multiple Access or TDMA), frequency (in case of FDMA), code (in case of CDMA), space (in case of SDMA), and polarization (in case of PDMA), respectively. These protocols predefine assignments of these independent media to the communicating nodes, allowing the nodes to transmit without contending for the medium. Among them the TDMA is the most suitable technique in ad hoc networks for energy conservation and hence enhanced lifetime. Though the time synchronization problem is there, but still TDMA with its predetermined schedule, offers the opportunity to keep the radio in the off (*sleep*) mode when in the schedule the node is neither to transmit nor to receive. It gives an enormous facility to conserve power at individual nodes by reducing the duty cycle significantly. However, efficient scheduling problems in wireless networks are NP-complete. So, it is a great challenge to find the optimum number of time slots and the optimal assignment of slots to the nodes. It is specially harder for ad hoc networks because of the lack of centralized control and limited knowledge of the topology.

1.2.2 Routing Challenges

It is to be noted that in wireless ad hoc networks, the MAC protocol essentially enables us to exchange messages only among the nodes within range, but it does not allow to communicate among the nodes beyond the transmission range. The routing is a process of exchanging messages between any two nodes in a connected network, via multihop paths, if necessary.

The design of efficient routing protocols in ad hoc networks is a major challenge as the status of the communication links between the nodes is a function of several factors such as the mobility of the nodes, the transmission power level, and the interference between neighbouring nodes, etc. In general, the routing protocols for wired networks first perform route discovery, followed by periodic updates of routing database. To adapt these protocols for mobile ad hoc networks, the frequency of periodic updates must be fast enough to maintain the consistency of routing information. Nevertheless, the action of increasing the frequency of routing updates that produces more and more control messages, is prohibitive in mobile ad hoc networks due to the limited battery-power of the nodes and the reduced capacity of the wireless links [148]. So, routing protocols for wired networks are not suitable

for routing in wireless ad hoc networks.

Extensive research has been done so far to devise efficient routing protocols for ad hoc networks.

The topology-based routing protocols find a route from a source to a destination according to the metrics of the network links and they forward packets based on the address of the destination node. In proactive topology-based routing, nodes share routing information even if there is no specific request for a route to maintain consistent and up-to-date routing database. Destination Sequenced Distance Vector (DSDV) [129], Optimized Link State Routing (OLSR) [44, 45], and Wireless Routing Protocol (WRP) [111] are some of the well-known proactive routing protocols. The DSDV is a modified version of the Bellman-Ford algorithm whereas the OLSR is based on the link-state algorithm. Each of these protocols periodically exchange routing information with other nodes mostly by flooding. However, they sometimes apply certain techniques to reduce the flood traffic, e.g. OLSR employs multipoint relays (MPRs) where each node selects a subset of neighbours to retransmit its update packets. Unfortunately, such techniques are in general computationally hard.

Reactive topology-based routing initiates a route discovery process only when there is an explicit request. It significantly reduces the memory consumption in the nodes and generates control traffic only when needed. However, it typically floods the network with control messages when multiple nodes initiate route discovery. The Ad Hoc On-Demand Distance Vector (AODV) [128], Dynamic Source Routing (DSR) [80], Lightweight Mobile Routing (LMR) [48], Temporally-Ordered Routing Algorithm (TORA) [126] and Associativity-Based Routing (ABR) [162] are some of the well-known reactive routing protocols.

The position-based routing protocols find the best way to forward packets to the geographical position of the destination without requiring the establishment and maintenance of routes [35, 184]. For computation, the information about the geographical positions of the communicating nodes should be readily available. Here, each node needs to determine its own position using GPS (Global Positioning System) or some other kind of positioning system [72]. Each node also propagates its position information to all other nodes using flooding. The advantage of exchanging position information is that it consumes significantly less bandwidth than exchanging complete routing tables even if the network is flooded. But, this positional information limits its applicability where such systems are not embedded or the accuracy is not satisfactory. The Distance Routing Effect Algorithm for Mobility

(DREAM) protocol [14], Grid (based on Grid Location Service or GLS in short) [97] are some of the well-known position-based protocols.

In summary, the existing routing protocols always use flooding or selective flooding to establish and maintain routes. They just differ in the number of necessary routing-related tables, in the methods by which changes in network topology are accounted, or in the way they tackle the issues related to flooding, updating routing tables, the strategy used to reduce the control traffic, etc. It is immediately apparent that the amount of flood requests and thus the amount of control traffic are directly proportional to the number of nodes participating in the routing update process. So, the efficiency of the existing routing protocols can be enhanced if a suitable routing backbone can be embedded in the network for route establishment and maintenance instead of involving all nodes. A hierarchical backbone, where only a subset of nodes are part of the backbone, generates less control traffic and requires reduced maintenance effort. Here, the routing process for the whole network is governed by the subset of nodes of the backbone only. Hence, a suitable hierarchical backbone, though virtual [103, 119, 166, 167], can overcome the absence of pre-existing routing infrastructure and also can provide enough scope to the existing routing protocols to optimize their performances.

However, building a hierarchy that optimizes the number of head nodes preserving overall connectivity in the network [15, 64], itself is a challenge. This task is even more difficult in ad hoc networks as the hierarchy is to be built in a distributed way based on local information only [36]. In addition, finding an optimum hierarchy that limits the end-to-end communication delay, the routing path length, and the energy consumed at each node, is definitely even more challenging [60, 113]. It is also interesting to investigate the routing challenges in special types of ad hoc networks, namely the Bluetooth networks and the wireless sensor networks as well.

1.2.3 Routing in Bluetooth Networks

As mentioned earlier, Bluetooth is a wireless technology that is being used to deploy personal area networks using IEEE 802.15.1 standard [78]. A Bluetooth enabled device operates in the unlicensed ISM band at 2.4 GHz. Unlike other ad hoc nodes, the system employs a frequency hop transceiver to combat interference and fading, and provides many FHSS (Frequency Hopping Spread Spectrum) carriers.

The Bluetooth architecture consists of a basic unit called *piconet* where a group

of Bluetooth enabled devices share a common frequency hopping sequence. Each piconet has a *master* unit which selects a frequency hopping sequence for the piconet and controls the access to the channel. Other participants of the piconet, known as *slaves*, are synchronized to the hopping sequence of the piconet master. Within a piconet, the channel is shared using a slotted time division duplex (TDD) protocol, with $625\mu s$ slot time, where a master uses polling to allocate time-slots to slaves. Communication is always between a master and one or more slaves (point-to-point or point-to-multipoint). There is no direct communication between slaves.

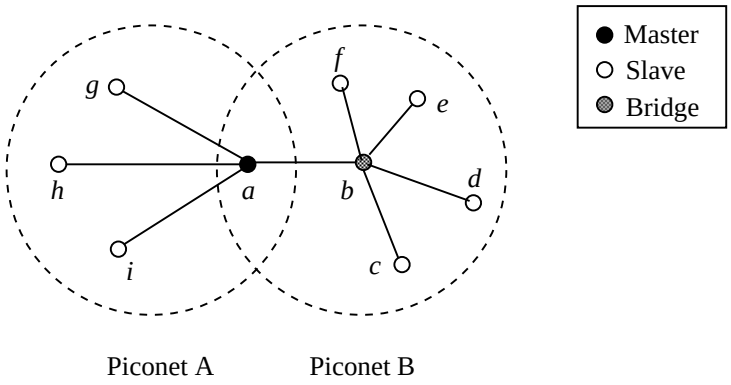


Figure 1.2: Bluetooth piconet and scatternet

There are mainly two limitations associated with a piconet that restrict its applicability to large networks. The first limitation is that the maximum number of slaves that can be active simultaneously in a piconet is a predetermined constant κ, (usually $\kappa = 7$). If there are more than κ nodes connected to a master, at any time instant, only κ slaves can remain active and can exchange messages, and remaining slaves become inactive (known as *parked* in Bluetooth terminology). Secondly, the extent of a piconet is limited within the limited transmission range of the master.

However, Bluetooth network can be extended by interconnecting several piconets. Use of different hopping sequences made it possible to configure multiple piconets in a common area. In this case, the network is called a *scatternet* (Figure 1.2). The piconets are interconnected via *bridge* nodes. The bridges between the piconets can have the role of slave in all piconets to which they belong to or of a master in one piconet and slave on the others. A bridge cannot be the master in more than one piconet because the master assigns the frequency hopping sequence for a piconet. Figure 1.2 shows a typical scatternet in which the bridge node a is a master in the piconet A and a slave in the piconet B.

Given a collection of Bluetooth-enabled devices, the *scatternet formation problem*, is to identify various groups of nodes, each group forming a piconet around a *master* and to select a few *bridges* to act as relays to forward traffic from one piconet to another [170]. The resulting scatternet should remain connected, and satisfy all constraints posed by the underlying physical layer. The scatternet should attempt to connect all devices by using minimum number of piconets, to reduce the probability of frequency overlapping between piconets. Also, it is necessary to limit the diameter of the scatternet, so that the devices may communicate with less delay. A set of Bluetooth devices can be organized into a scatternet in numerous configurations, however, finding an optimal one is NP-complete. The scatternet formation problem with minimum number of piconets (or bridges) is an NP-hard problem [110]. Moreover, the absence of centralized control along with the energy constraints makes the problem harder that demands energy-efficient distributed algorithms based on local information only.

1.2.4 Energy Consumption and Power Control

Since the primary objective of forming ad hoc networks is to support *ubiquitous computing* [3] without any basic infrastructure or external support, it is unlikely that nodes will get power supply from external sources. Moreover, occasional recharging of the built-in battery may not be possible in cases when they are used in remote areas, e.g. military operation, disaster recovery, etc. Therefore, the battery power, which is to drive all electrical components like the communication and computation units, must be consumed in a very judicious way.

For radio transmission, the energy required to transmit a single bit of information varies as the kth power of the Euclidean distance between the transmitter and the receiver, where $2 \leq k \leq 6$, depending on the condition of the surrounding media [136]. It implies that the energy consumption increases many fold even for small increase in distance. Moreover, packet reception also consumes significant but usually a fixed amount of energy. As each node works independently and without any centralized control, various other tasks related to data communication, (e.g., medium access control, routing decisions, etc.), are to be performed by each node. Such overheads are too significant unless proper measures are taken to reduce the power consumption for each activity [81, 143], with special attention to communication.

It is possible to build an ad hoc network where each node transmits at a constant

power level P_{max}, covering a range R_{max} of transmission. However, it is wasteful in terms of energy for two considerations. Firstly, if distance between two nodes is less than R_{max}, they can communicate with power less than P_{max}. As nodes operate with limited battery power, one needs to develop a scheme to reduce the transmission power of a node to enhance its life [75, 76]. Secondly, the use of high and fixed power level for each transmission does not allow to promote better routing schemes. For example, exposed terminal problem is more severe when nodes use high and fixed power level for transmission [84, 157]. It also reduces the network throughput by increasing the possibility of interference.

Usually, each ad hoc node can potentially adjust its transmission power similar to our cell phones. The power control can be either link-based or node-based. In link-based power control (LBPC), before each transmission, a node adjusts its power level to be just sufficient to communicate along the specific link. However, in ad hoc networks, the mobility of the nodes changes the power requirements of individual links frequently, and to have the current information about it, a node may have to exchange some additional control packets causing additional drainage of power. Also, the adjustment of transmission power level before each communication poses additional load on the power-starved nodes. Therefore, LBPC is, in general, not feasible in such networks as it demands much overhead for link maintenance. On the other hand, in node-based power control (NBPC), based on current topology, each node decides its own fixed transmission power $\leq P_{max}$, same for all its links that maintains its connectivity with the rest of the network, but at the same time helps to reduce the total power required for communication [122]. However, the problem of finding optimal power assignment following NBPC that just preserves connectivity is NP-Complete. The problem is definitely much harder when we need to consider other parameters as well, like the end-to-end delay, the maximum power level for each node, non-uniform power consumption model, the total power consumption on a path, etc.

Moreover, it should be kept in mind that such algorithms must be simple, or light-weight, and should consume less energy by exchanging just a few messages only. For moderately static ad hoc networks, complex algorithms can be applied to optimize the node powers efficiently since computation is to be done only during initialization, or in case of failures.

1.2.5 Data Gathering in Wireless Sensor Networks

As discussed earlier, WSN is primarily used to collect some field data and to forward the same to one or more sinks. Such collection of field data is usually known as *data gathering and aggregation* in literature.

In WSN, the energy dissipation due to communication follows the same model and constraints posed by general ad hoc networks. However, there are several differences between the two. Usually a large number of sensors are deployed over a vast region along with the assumption that many of the deployed sensors may fail to operate. The sensors are very light weight (about 20 to 50 gms) with very small battery as a source of power. There is no possibility for recharging in general, so once the battery dies out the sensor is of no use. Thus, it is very important to use the available battery power very efficiently to improve the lifetime of individual sensors and the WSN as a whole. The lifetime is defined to be the time during which the sink can gather data from the sensors.

A *data gathering schedule* specifies how the data packets from all sensor nodes are collected and routed to the sink in each round [120, 121, 123, 124, 139]. Given the locations of the sensors and the sink, and the energy level at each sensor, the problem of finding the optimal routing paths for efficient data gathering that maximizes the network lifetime is referred to as the Maximum Lifetime Data Aggregation (MLDA) problem [79, 82]. This problem is NP-hard [132]. Lot of efforts [29, 106] have been devoted to solve the problem, but a realistic solution with distributed computation using local information and minimum message communication is yet to come up.

1.3 Scope of the Thesis

From the above discussion, it is clear that whatever be the network, either a general ad hoc network, or a Bluetooth network, or a WSN, given any distribution of large number of nodes, it is a challenging problem to design efficient algorithms to resolve the problems related to media access control, routing, and topology control, and finally to build a network that can function appropriately to achieve its global objective. This thesis envisages the scenario of a large number of ad hoc nodes with built-in capability of message communication, distributed randomly over a region, where each node runs a series of localized but coordinated greedy algorithms to resolve the design issues namely the media access control, transmission power control, and routing backbone design, to culminate an efficient network topology

for regular operation of the network. This work focusses on these issues with special emphasis on optimal utilization of the limited resources, namely the battery power, bandwidth, computing power and storage capacity [24, 31]. To conserve power and bandwidth, our algorithms aim to keep the number of communications per node low. To keep the computing power and storage demands low, proposed techniques use distributed or localized light-weight algorithms based on very limited local neighbourhood information.

During this initialization process, the nodes are assumed to be static, however, each algorithm offers the scope of self-organization in case of limited perturbation in the network topology in the form of new nodes joining the network, or some nodes leaving the network. This work assumes that for the given devices with communication radios, the physical layer and the data link layer protocols are already built-in that offer an initial platform for error-free message exchange between devices.

The following subsections present the contribution of this thesis towards specific problems.

1.3.1 Distributed Time Slot Assignment for TDMA Scheduling

As discussed in section 1.2.1, TDMA is a reasonable choice for contention-free medium access control in energy-starved ad hoc networks where the nodes are synchronized. In TDMA, collision is avoided by assigning dedicated time slots to the nodes. For an ad hoc network with n nodes, a trivial TDMA solution [56] is to have a frame length n, i.e., n slots per round, that allows assigning a unique slot to each node avoiding any possibility of collision. Then, it is obvious that a node ready to transmit may have to wait, in the worst case, for $(n-1)$ slots. Also, in every round there may be a large number of idle slots, making inefficient utilization of channel. A better technique is to reuse the slots for nodes sufficiently separated in space, or beyond the range of interference, which is termed as STDMA (Spatial Time Division Media Access) [86, 117]. In STDMA, an important optimization objective is to minimize the frame length, assigning at least one slot to every node for contention-free transmission. The secondary objective is to allow as many nodes as possible to transmit simultaneously during a single slot without conflict for increasing the channel utilization. But even for static packet radio networks, the TDMA scheduling problem, either broadcast scheduling [56] or link scheduling [6, 57] is NP-complete. Even polynomial-time algorithms with constant approximation ratios appear to be

unlikely for general graphs.

In [133], it is shown that the worst-case number of slots required per frame is $\theta\Delta$ for broadcast scheduling and $\theta^2\Delta$ for link scheduling, where Δ is the maximum node-degree and θ (the thickness of the graph) is the minimum number of planar graphs into which the given graph can be partitioned. Also, different approximation algorithms have been proposed in the literature [23, 89, 133].

In this thesis, we established an upper bound $min\ \{(19\Delta - 18),\ (\Delta^2 + 1)\}$ on STDMA frame length for broadcast scheduling, assuming that the nodes are deployed over a two-dimensional region. It essentially shows that the minimum TDMA frame length is of linear order of Δ. It improves the earlier bound in [133] when the network is dense for which θ is of $O(\Delta)$. We developed a distributed algorithm for slot assignment for multi-hop ad hoc networks with the local knowledge of the topology. The time and message complexities of this algorithm are $O(n)$ and $O(\Delta n)$, respectively. We extended our algorithm to assign multiple time slots per node within each frame. Simulation study shows that it enhances the utilization of slots compared to earlier works [56]. We also developed distributed algorithm to adjust schedule locally for the incremental changes in the topology due to node movements and/or node failures. It is also shown that our algorithm is energy efficient as nodes can be put into sleep modes periodically during slot assignment process.

1.3.2 Connected Dominating Set Based Routing Backbone

Connected dominating set (CDS) is found to be a reasonable choice for hierarchical routing backbone, where dominating nodes form the backbone of the network [173, 163]. A dominating set (DS) of a graph is a subset of nodes such that each node that is not a member of the subset is adjacent to at least one node in DS. A CDS is a DS that induces a connected subgraph. It is desirable to find a minimum connected dominating set (MCDS) as a virtual hierarchical routing backbone. However, finding an MCDS in a graph is NP-hard [43].

In [18, 52, 151], three centralized CDS construction algorithms are proposed where all of them ensure $3H(\Delta)$ approximation of MCDS, H is the harmonic function and Δ is the maximum node degree. But, the distributed implementation of this greedy algorithm with the approximation factor lying between $\frac{log\Delta}{2} - \frac{1}{2}$ and $3H(\Delta)$ has very high time complexity $O(n^2)$ and message complexity $O(n^2)$ as shown in

[163], where n is the number of nodes in the graph. The approximation factor of the distributed algorithm proposed by [173] is exactly $\frac{n}{2}$. Its message complexity is $\theta(m)$, where m is the number of edges, and its time complexity is $O(\Delta^3)$ [163].

It has been shown in [163] that any distributed algorithm for nontrivial CDS requires at least $O(n \log n)$ bit messages. Authors also presented a distributed algorithm that has an approximation factor of at most 8. Its time and message complexities are $O(n)$ and $O(n \log n)$, respectively. It shows that this algorithm is a message-optimal distributed algorithm for nontrivial CDS, however it takes $O(n)$ time. The algorithm presented in [173] is an improvement in average size of CDS over the previous works.

In this thesis, we propose a distributed greedy algorithm for finding CDS for multi-hop networks with minimal information about its local neighbours, namely, the neighbours' identities and their connections. The time and message complexities of the algorithm are $O(1)$ and $O(m)$ respectively. The runtime of our algorithm is an improvement over earlier results [163]. It requires only 6 rounds of message exchanges, and hence is suitable for ad hoc networks with fast convergence. Simulation study shows that compared to earlier results [173], the sizes of the CDS reported by our algorithm are smaller. Even when the network is sparse with smaller cliques, our algorithm generates CDS of considerably smaller size.

1.3.3 Scatternet Formation in Bluetooth Networks

Given a collection of Bluetooth-enabled devices, the *scatternet formation problem*, with minimum number of piconets (or bridges) maintaining overall connectivity is NP-hard [110].

Some solutions for single-hop networks are presented in [94, 95] and [96]. For multi-hop networks, the method described in [180] generates a tree-like scatternet. However, it requires that the protocol to be initiated by a designated node *blueroot*, and hence it is not truely distributed. Some of the solutions that produce topologies different from a tree are those presented in [99, 16, 130, 165]. The protocols proposed in [99, 159] require that each Bluetooth device be accompanied with GPS (global positioning system) capability, which limits its applicability. Algorithms presented in [1, 140] depend on a single device to design the scatternet topology and to notify it to other devices. The distributed Tree Scatternet Formation (TSF) protocol proposed in [160] gives relatively short scatternet formation latency on an average,

but it does not minimize the number of piconets.

The time complexities of all these algorithms are $O(\lambda)$ where λ is the diameter of the topology graph. Moreover, none of the solutions attempts to minimize the number of *parked* nodes in the scatternet. Collision often arises when two or more would-be masters try to communicate with a common neighbour during scatternet formation. None of the algorithms has addressed this aspect as well.

In this thesis, we develop an energy-efficient distributed algorithm for Bluetooth scatternet formation for multi-hop networks without any global knowledge of the topology. It produces a mesh of piconets offering more robustness by providing multiple paths among the nodes. We implemented a synchronization mechanism at the first phase that ensures collision-free message exchange in the following phases of scatternet formation. It allows the nodes to remain into sleep mode periodically even during the scatternet formation making the procedure more energy-efficient. We also incorporated techniques to reduce the number of parked nodes in the resulting scatternet. Simulation studies show that our method generates scatternet with fewer bridges, and fewer masters and bridges compared to those of [16].

1.3.4 Power-Aware Topology Control

Topology control in ad hoc networks essentially means a coordinated control of transmission powers at individual nodes to achieve a network wide perspective like connectivity [142, 143], energy-efficient communication, coverage, etc. The power control mechanism can be either link based or node based. Since the link-based power control (LBPC), that selects power level based on the requirement of each link, demands excessive overhead to maintain the link status, as has been explained in Section 1.2.4, our work focuses on node based power control (NBPC) for configuring the network topology to support energy-efficient communication.

The centralized solutions to the topology optimization problem attempt to achieve an optimal network topology, assuming that all the information regarding node positions, link qualities, etc., is available at a central node for computation. However, accumulation of global network information requires excessive communication overhead and hence is not practical in ad hoc networks. Some solutions focused on establishing routes and maintaining these routes under frequent and unpredictable topology changes [12, 26, 48]. A topology control algorithm using heuristics based on a Delauney triangulation of graph is presented in [76]. A centralized spanning

tree algorithm, for achieving connected and bi-connected static networks while minimizing the maximum transmission power, is presented in [132].

A distributed connectivity preserving position-based topology control algorithm is proposed in [137]. This is further improved in [98]. A 2-approximation distributed algorithm is presented in [100].

Our approach is to propose a distributed protocol by which each node, using minimum amount of information (such as neighbours' identities and power demands of its links), can determine its power level in a coordinated way resulting a network configuration for energy efficient routing. Here, we developed two distributed NBPC algorithms for multi-hop wireless ad hoc networks without any global knowledge of the topology. The first algorithm takes $O(n)$ rounds (n is the total number of nodes), and results better optimization in terms of node power levels applicable to lifetime-critical networks with nodes having less mobility. The second algorithm takes only two rounds of message exchanges, but significantly reduces the node power level. This can be used in mobile networks due to its less computational complexity that allows faster convergence with frequent changes in topology. Simulation studies show that both the algorithms perform appreciably well than earlier algorithms.

1.3.5 Data Gathering in Sensor Networks

In [82], the Maximum Lifetime Data Aggregation problem is mapped to an integer linear programming (ILP) problem. But the number of independent variables is too large to be solved by sensor nodes. The linear relaxation to this ILP can be computed in polynomial time resulting a linear approximation to the optimal solution. The Low-Energy Adaptive Clustering Hierarchy (LEACH) [70] uses a clustering based protocol for data gathering and aggregation. The Power-Efficient GAthering in Sensor Information Systems (PEGASIS) [105] improves the lifetime over LEACH by forming a chain with the sensors where each sensor communicates with the base station in turn to deliver the aggregated data. The time and message complexities of both the algorithms are $O(n)$.

In this thesis, we developed a distributed algorithm for data gathering schedule for multi-hop WSN without any global knowledge of the topology. It is difficult to compare the time and message complexities of our proposed algorithm with PEGASIS or LEACH as we consider a multi-hop network, whereas earlier works assumed single-hop network. Both the time and message complexities of our algorithm are

$O(n)$, same as PEGASIS, if we assume that each sensor can directly communicate to the Base Station (BS). The actual worst case time and message complexities could be $O(n^2)$ if we consider that only multi-hop communication is possible to reach the BS. We have shown through simulation that the schedule produced by our proposed algorithm significantly improves the system lifetime compared to the schedule based on the Minimum Spanning Tree (MST) and the Shortest Path (SP) routing techniques. Performance comparison with PEGASIS [105] shows that our algorithm performs better as more and more nodes die out.

1.4 Thesis Organization

This thesis is organized in eight chapters. In Chapter 2, we discuss the basic terminologies and some of the well accepted models for the study of ad hoc networks. The following chapters and algorithms are based on the models presented in this chapter.

In each of the following chapters, we discuss one specific problem within the scope of this thesis. For each problem we briefly discuss some of the well accepted solutions from the literature and also our proposed solution. All the problems being NP-hard, the performance of our proposed algorithms have been compared with other well-known algorithms in this area through simulation.

Our study on TDMA scheduling is presented in Chapter 3. In Chapter 4, we discuss the connected dominating set based routing backbone design problem. We present our energy efficient Bluetooth scatternet formation protocol in Chapter 5. In Chapter 6, we develop node-based power control algorithms that reduce the power consumption per node. In Chapter 7, we present the distributed data gathering algorithm for multihop Wireless Sensor Networks. Finally, Chapter 8 summarizes our contributions and also outlines some of the future directions.

Chapter 2

Network Model

Given a distribution of ad hoc nodes over a two dimensional region, this thesis attempts to formulate some energy-efficient distributed algorithms for self-organization of the network. For that, it is essential to consider appropriate models for the wireless links, network topology, energy consumption at nodes and finally the distributed nature of functioning of ad hoc networks. In this chapter, we present the simplest model that has been widely accepted in the wireless ad hoc network community so far.

Since, both wireless sensor networks and Bluetooth networks may be considered as special cases of ad hoc networks, this model equally applies to those networks as well.

This chapter is organized as follows. Section 2.1 presents a brief description of the wireless channel models used in the literature. The graphical representation of the network topology with associated definitions and notations used in this thesis is introduced in Section 2.2. To make our design energy-efficient, the energy consumption model used here is presented in Section 2.3. Section 2.4 describes the philosophy and the goal of our algorithms, designed for ad hoc networks modeled as distributed systems with special assumptions. Since extensive simulation studies have been made to evaluate the performance of our algorithms, the simulation model followed in this thesis is presented in Section 2.5. This chapter concludes with a summary.

2.1 Radio Signal Propagation Model

Each node in an ad hoc network is a radio unit capable of transmitting and receiving messages. Depending on the model of the wireless channel, the transmission power of a node determines its transmission range (R). In the basic ideal model of a wireless link, two nodes are said to have a perfect link (with a 100% packet reception rate), if their distance $d \leq R$, and to have a non-existent link (0% reception rate) otherwise.

So, before modeling the ad hoc networks, it is important to decide on the wireless channel model. The model presented here is mainly based on the material contained in [136, 144].

A node u when transmits radio signal with a power P_t, another node v at a distance d from u can receive the signal successfully, if and only if the received power $P_r \geq \beta$ where β denotes the threshold power for successful reception. Then it is said that there exists a radio channel between u and v, or in other words, nodes u and v can communicate directly.

For a given value of transmitted power P_t, the received power P_r is determined by the signal propagation model, whereas the exact value of β depends on the properties of the transceivers and also on the data rate of communication. Therefore, the existence of a direct radio channel between two nodes of an ad hoc network can be predicted if the signal propagation model is already known.

Exact modeling of wireless radio signal propagation has been studied in-depth for several decades and it has been found to be one of the hardest tasks for system designers. Based on different environments, three propagation models have been accepted widely. The following subsections describe each of them briefly.

2.1.1 The Free-Space Propagation Model

It considers the case when the transmitter and receiver can always communicate along a line-of-sight (LOS) path. According to this model, the power received at a distance d is given by:

$$P_r(d) = C_f.P_t/d^2 \tag{2.1}$$

where P_t is the transmitted power and C_f is a constant dependent on transceiver characteristics and signal frequency.

Hence, the range R of a node u transmitting with a power P_t is:

$$R = \sqrt{(C_f.P_t/\beta)} \tag{2.2}$$

Then, in two dimensions, the radio coverage area of node u is a circle of radius R, centred at node u, indicating that any other node v at a distance d from u can receive the packet successfully from u, if and only if $d \leq R$.

Since the assumption of LOS path is not valid for most practical purposes, the free space propagation model seldom produces correct estimates.

2.1.2 The Two-Ray Ground Model

This model assumes that signal propagates along two paths from the transmitter to the receiver, one is along the LOS path, and the other is a ground-reflected path. Here,

$$P_r(d) = C_t.P_t/d^4 \tag{2.3}$$

Hence, the range R of a node transmitting with a power P_t, by this model is:

$$R = (C_f.P_t/\beta)^{1/4} \tag{2.4}$$

2.1.3 The Log-Distance Path Model:

Combining both analytical and empirical methods, this model estimates the received power as,

$$P_r(d) \propto P_t/d^{\alpha} \tag{2.5}$$

where α is called the *path loss exponent*, and depends on the environmental conditions. Hence, the range R of a node transmitting with a power P_t, by this model is:

$$R \propto (P_t)^{(1/\alpha)} \tag{2.6}$$

From experimental observations in different media conditions, like free space, urban area, indoor LOS and indoor non-LOS propagation, it has been found that $2 \leq \alpha \leq 6$.

So, from Eqns. (2.1) to (2.6), it is evident that the log-distance model is a generalization of the other two models. Therefore, in this thesis, we will follow *log-distance propagation model* only.

Also, it is to be noted that whatever be the propagation model, the range R of a node can be manipulated if we assume that the nodes can vary its transmission power P_t individually. Then, unlike the wired networks, the existence of links may vary with P_t that enables us in devising the technique called *topology control* in ad hoc networks even when the nodes are static. In case, P_t remains constant, it is obvious that a link may appear or disappear between nodes depending on their movements, making the network graph a dynamic one.

2.2 Network Topology Graph

In general, it is assumed that a node transmits with a power P_{max} which corresponds to its maximum transmission range R_{max}. Also, the P_{max} (and hence R_{max}) is equal for all nodes in a network, unless mentioned otherwise. So, when a node transmits with a power P_{max} all nodes within its transmission range R_{max} can listen to it. This is termed as single-hop communication. In ad hoc networks, two nodes, not within the range of each other may also communicate, forwarding the packet in multiple hops through intermediate nodes. This philosophy of possible communication is captured here in the form of a graph namely the topology graph.

Given a set of n ad hoc nodes distributed over a two dimensional region, represented by a set of n vertices $V = \{v_1, v_2, \cdots, v_n\}$, we can construct a topology graph $G(V, E)$, where two nodes u and v are connected by an edge $(u, v) \in E$ if and only if their distance $d \leq R_{max}$ (i.e., they are within the transmission range of each other). Nodes in ad hoc networks are assumed to have unique IDs [112]. We don't use the nodes' IDs in general in our algorithms, but only sometimes use it for breaking the symmetry (such as tie breaking). Thus, uniqueness in IDs in any form is sufficient for our discussion.

Since, R_{max} is equal for all nodes, the graph G is undirected. Also, it is to be mentioned that we limit our discussion on connected topology graphs only since there is no way for communication between disconnected components or nodes.

Remark: When a spatial distribution of ad hoc nodes on a two dimensional plane is modeled as a topology graph $G(V, E)$, it preserves only the link connectivity among the nodes, but looses the information about their coordinates, and hence the exact pair-wise distances as well.

Example 2.1 *Figure 2.1(a) shows a distribution of eleven ad hoc nodes on a two dimensional plane with their equal maximum transmission range R_{max}. The dotted circle around each node represents its transmission region. We note that the nodes b, j and k lie within the transmission region of node a. This fact is represented by adding three edges from node a to nodes b, j and k in Figure 2.1(b). Thus, Figure 2.1(b) represents the topology graph corresponding to the distribution of nodes of Figure 2.1(a).*

Within a topology graph $G(V, E)$, the degree d_i of a node v_i is the number of nodes adjacent to v_i. So, when v_i transmits all d_i nodes within its transmission range

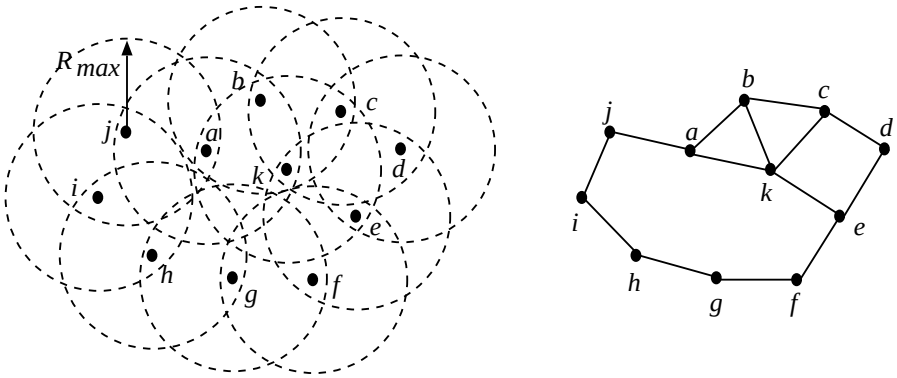


Figure 2.1: An ad hoc network: (a) a distribution of eleven nodes on a two-dimensional region and (b) corresponding topology graph

can listen to it. This is termed as single-hop communication. In ad hoc networks, two nodes, not within the range may also communicate, if the intermediate nodes agree to forward the packet in multiple hops. Without loss of generality, it is assumed that all nodes of any ad hoc network are by virtue, co-operative. The maximum node degree of a graph, denoted by Δ, plays an important role in our work.

Definition 2.1 *In $G(V, E)$, the hop-distance $\delta(v_i, v_j)$ between two nodes $v_i, v_j \in V$, is the length of the shortest path in hops between them in $G(V, E)$. Two nodes v_i and v_j are said to be h-hop away if $\delta(v_i, v_j) = h$.*

Definition 2.2 *The diameter, λ, of a graph is the maximum length of the shortest paths, in terms of number of hops, between two vertices in the graph, i.e., $\lambda = max\ \{\delta(v_i, v_j), \forall v_i, v_j \in V\}$.*

Definition 2.3 *$N^h(v_i)$, the set of h-hop neighbours of a node v_i in $G(V, E)$, is the set of nodes $V' \subseteq V$, $V' = \{v_{i1}, \cdots, v_{ik}\}$, such that $\delta(v_i, v_{ij}) \leq h$ for $1 \leq j \leq k$.*

Definition 2.4 *The h-hop partial graph of v_i, $PG^h(v_i)$ is a subgraph $G'(V', E')$ of $G(V, E)$ induced by the node set $V' = N^h(v_i) \cup \{v_i\}$ and deleting the edges between any two h-hop away nodes of v_i.*

The h-hop neighbours $N^h(v_i)$ along with the h-hop partial graph $PG^h(v_i)$ is called as h-hop partial information. The distributed self-organization algorithms discussed in this thesis do not require the global knowledge of the topology graph

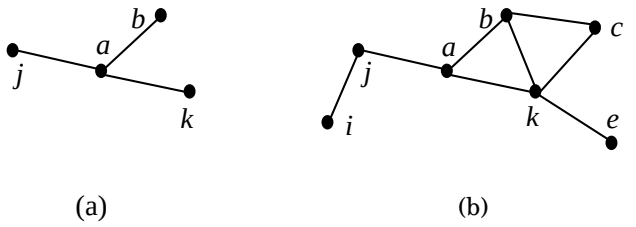


Figure 2.2: The partial graphs of the node a: (a) 1-hop partial graph $PG^1(a)$ and (b) 2-hop partial graph $PG^2(a)$

$G(V, E)$. They all work with a limited (partial) knowledge of the network. A node v_i performs various computations and hence takes various decisions based on the information of its neighbours reachable in h hops only, in most of the cases $h \leq 2$. All of our algorithms work with 2-hop partial information of the topology graph available in each node. So, each node will be dealing with $N^1(v_i)$, $N^2(v_i)$, $PG^1(v_i)$ and $PG^2(v_i)$ and additional information derived from them. Note that, $PG^{(\lambda+1)}(v_i)$ is the topology graph $G(V, E)$ itself.

It is evident that the topology graph $G(V, E)$ models an ad hoc network appropriately when we are interested only about the connectivity information among the ad hoc nodes. However, for developing algorithms where we need more information than mere connectivity, we associate a weight $w_{u,v}$ with each edge (u, v) to form the *weighted topology graph* $G(V, E, W)$. Depending on the objective, the weight may represent either the Euclidean distance between nodes u and v, or the transmission power required for the link, etc.

2.3 Energy Consumption Model

As has been mentioned in Chapter 1, the nodes of ad hoc / sensor networks are generally battery-driven and preferably unattended during the entire lifetime (in case of WSN), making energy conservation a vital issue in network design. Therefore, it is essential to model the energy dissipation in nodes realistically and accurately as far as possible.

Depending on the application, the nodes of an ad hoc network may be of diverse types like laptops, PDA's, cell phones, palm tops, sensor nodes, etc., and hence to have a uniform energy model, independent of nodes, we focus only on the energy consumed by the wireless transceiver of the nodes which handle the wireless com-

munication between nodes. It is not only our choice, but is followed extensively in the literature as well, motivated by the fact that the energy consumed by the wireless card is always a significant portion of the total power dissipation in an ad hoc node. For example, a PDA device equipped with an *IEEE 802.11* card consumes about 35% energy for wireless communications. For the sake of completeness, we briefly describe here the most conventional power consumption model used in wireless networks as mentioned in [104].

According to this model, the energy consumed by a node v_i in receiving a k-bit message is

$$Rx = \varepsilon_{elec} \times k.$$

The energy consumed by a node v_i to transmit a k-bit message to v_j is

$$Tx_{i,j} = Rx + \varepsilon_{amp} \times k \times d_{i,j}^{\alpha}, \text{ where } 2 \leq \alpha \leq 6.$$

Here, ε_{elec} is the constant energy required by the transmitter or receiver circuitry and ε_{amp} is that for the transmitter amplifier to transmit a single bit to *unit* distance, and $d_{i,j}$ is the Euclidean distance between v_i and v_j. It is assumed that the radio channel is symmetric, i.e., $Tx_{i,j} = Tx_{j,i}$.

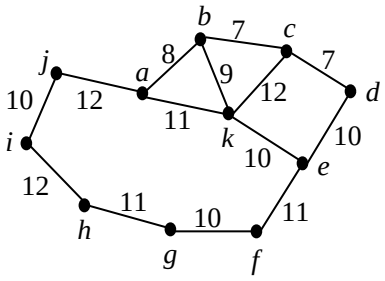


Figure 2.3: The weighted topology graph $G(V, E, W)$

Example 2.2 *Figure 2.3 shows the weighted topology graph corresponding to the topology graph shown in Figure 2.1 where the weight $w_{u,v}$ of an edge (u, v) denotes the minimum transmission power $Tx_{u,v}$ required for the link. The maximum transmission power of a node is assumed to be $P_{max} = 12$ units. The links are assumed to be symmetric.*

2.4 Distributed System Model

With the models for radio link, network topology and energy consumption already determined, now we are in a position to envision an ad hoc network as a distributed system with a set of independent processors that communicate and co-ordinate their actions only by passing messages to achieve a global objective. All the nodes are assumed to be of equal ability without any special leader. While developing the distributed algorithms [107] we assumed the following features of the distributed system:

- Each node has a unique node ID.
- The communication network is connected.
- The communication links are bidirectional and reliable.
- The nodes do not share memory, but each node has enough local memory.

All our algorithms follow the *partially synchronous timing model* that lies in between synchronous and asynchronous timing models. Here, the computations in different nodes do not proceed in lock step fashion as it is in the synchronous model. However, there exist some restrictions on the relative timings of events. We assume that nodes have some knowledge of time or has some type of a time-out facility. As for example, when an algorithm requires more than one pass or phase, we assume that the passes or phases are synchronized in all nodes.

Though the time synchronization issue is beyond the scope of this thesis, we can just mention that all our algorithms apply when there is very little or no relative drift between the local clocks of the nodes. If the nodes are assumed to be homogeneous there may be a very small drift termed as clock skew. Some of the approaches to address this problem may be to run any of the traditional clock skew algorithm [7, 50, 93] for distributed systems on top of our protocol.

Under this timing model, the *time complexity* of an algorithm is measured in terms of the number of *rounds* until all the required outputs are produced or until all the processes halt.

The *message complexity* is measured in terms of the total number of messages that are transmitted. It may help us to measure the average number of message transmission per node and hence the energy consumption per node in communication.

The *computational complexity* refers to the order of internal computation done by any node in the system to take any decision to achieve the global objective.

In this thesis, based on this distributed system model, we develop algorithms, primarily for a distributed computation of a broad range of initialization and topology configuration in ad hoc wireless networks. The algorithms are locally executed at each individual node, based on only locally available information (for example node (v_i) computes with the knowledge of $PG^2(v_i)$ only) about the network. Compared to the centralized algorithms, these distributed algorithms result in the following advantages:

(a) Fewer message exchanges with local neighbours to collect information.

(b) Less energy overhead in communication.

(c) Less computation distributed in each node.

It is to be mentioned that the goal of such a localized algorithm [145] is to produce a *reasonably good solution* as fast as possible, perhaps with a bit of compromise in the accuracy.

2.5 Simulation Model

We supplemented our theoretical study of the distributed algorithms by an extensive simulation study. We framed the simulation environment to represent the ad hoc networks realistically. A two-dimensional rectangular region of length L and breadth B is first identified. We then set the number of ad hoc nodes n and the maximum transmission range R_{max}. The nodes are then distributed randomly over this two-dimensional region [8]. Based on the distribution of the nodes and R_{max}, a topology graph is constructed according to the model described earlier in this chapter. For a given set of values of n and R_{max} we have generated more than 100 random distributions of the nodes and hence more than 100 topology graphs. For a specific value of n we varied R_{max} to generate corresponding topology graphs to see the performance of our proposed algorithms. Additionally, we also varied the number of ad hoc nodes n from very small value of about 30 to around 500 to observe the performance of our algorithms. Since the performance of our algorithms largely depends on Δ, the maximum degree of the topology graph, we generated random graphs with a given value of Δ.

We simulated our proposed algorithms as well as the previously known algorithms on those randomly generated topology graphs. Depending on the specific

problem, various metrics are determined for comparing the performance of the algorithms. The performance metrics are compared with the centralized algorithms and/or distributed algorithms reported in earlier works.

Summary

A simple and well-accepted realistic graph theoretic model of ad hoc networks is presented for the purpose of our study of the distributed self-organizing algorithms. We studied the topic following the philosophy of distributed algorithms based on local information, suitable for mobile ad hoc networks where the resources are limited. This localized approach is used throughout the thesis and the objective behind the use of it is demonstrated here. In the following chapters we focus on developing localized distributed algorithms where the fast convergence is more important than the accuracy. The distributed algorithms are analyzed on the basis of the quasi-synchronous distributed system model described above. We also described the simulation environment for the experimental study of the algorithms. We have defined the various notations and terminologies of the model which will be used in the following chapters.

Chapter 3

TDMA: Upper Bound and Scheduling Algorithm

For ad hoc wireless networks, Time Division Multiple Access (TDMA) is a reasonable technique for managing wireless media that substantially reduces collisions, avoids wastage of energy in retransmission, and also improves energy conservation in battery-powered nodes by keeping radios off during the idle time. Here, the single communication channel is shared by nodes during a fixed number of time slots in a *round*, usually termed as *frame*, that is repeated periodically. At least one collision-free time slot is to be assigned to each node in a frame that essentially dictates a transmission schedule for the nodes in the network. The frame length, i.e., the number of slots in a frame, typically depends on the application, the number of shared users of the channel and the interference model.

Traditionally, TDMA has been proved to be a very effective MAC protocol and is being used in numerous applications such as GSM digital cellular systems, Personal Digital Cellular (PDC), Digital Enhanced Cordless Telecommunications (DECT), satellite systems, and many other systems. Spatial reuse TDMA (STDMA) is a fixed assignment access protocol for multi-hop radio networks. The idea is to increase the network capacity by allowing several radio terminals to use the same time slot when they are spatially separated in such a way that the interference caused is not too severe.

In radio networks, two types of scheduling are common. One is *broadcast scheduling*, in which each node is assigned a time slot during which it transmits and all the nodes within its range can listen to it. The other type is *link scheduling* where each link (u, v) of the network is assigned a slot, during which node u transmits and only its neighbour v listens to it, keeping all other nodes within its range sleeping. Link scheduling is more power conserving for point-to-point communication. But on the contrary, broadcasting a message to n neighbours, a node should transmit n times, if it follows link scheduling, whereas it is done just by one transmission in case of

broadcast scheduling. However, in case of ad hoc networks, broadcasting is often required for topology discovery, routing, etc. Especially, since there is no centralized control, broadcasting is an essential tool for each node to maintain the network in a distributed fashion. Therefore, we focus on the problem of *broadcast scheduling* for ad hoc nodes using STDMA.

The frame length plays an important role in STDMA. The frame length should be sufficiently large that will allow to assign at least one slot to each node in the network. On the contrary, the larger the frame length the larger is the latency for nodes to wait for its assigned slot. However, finding the optimal frame length for STDMA is an *NP-hard* problem. Along with fixing the frame length in STDMA, the challenges are to develop distributed algorithms to assign time slots to each node to maximize the bandwidth utilization. Moreover, due to inherent energy limitation and mobility constraints the distributed algorithms should be localized and easily adaptable to the changes in network topology.

Chapter organization

The rest of this chapter is organized as follows. Section 3.1 outlines the previous works on STDMA. In Section 3.2, we discuss the distinct features of our study on STDMA. In Section 3.3, we prove a new upper bound on the frame length for STDMA. The STDMA scheduling algorithms for single and multiple slot assignment for static ad hoc networks are presented in Section 3.4. In Section 3.5, we describe our simulation study and the results. Section 3.6 describes the rearrangement procedure of STDMA scheduling when topology changes due to node mobility. We also include a summary at the end of the chapter.

3.1 Background

Extensive studies have been done so far on both broadcast and link scheduling [133, 131, 150, 25, 37, 183] using TDMA for radio networks. The performance comparison in terms of throughput and delay for broadcast and link scheduling in STDMA presented in [67] interestingly shows that the link scheduling results a higher delay for low traffic loads, but can achieve much higher throughput than that for broadcast scheduling.

In TDMA scheduling [21, 39, 40, 65, 153, 182], the basic optimization objective is to minimize the length of the TDMA frame, such that each node gets at least

one slot for conflict-free transmission. This problem is NP-complete, even for a packet radio network where the nodes are static, and centralized control is feasible. For ad hoc networks, the problem is more complex due to its dynamic topology, and distributed nature of functioning. The secondary objective is to allow as many nodes as possible to transmit simultaneously during a single slot without conflict for increasing the channel utilization.

Earlier works on TDMA for packet radio networks, in general, keeps the frame length quite large [5, 21, 23, 39, 40, 51, 71, 133, 134, 135, 147, 153], so that at least one slot is ensured per node. For example, if N is the total number of nodes, a trivial TDMA solution is to have a frame length of N, where a unique slot is assigned to each individual node. However, it may cause a large number of free unutilized slots per frame, making inefficient utilization of channel. Also, a node may have to wait for a long period before getting its turn. The problem of minimizing the frame length in broadcast scheduling is solved by mapping it to the standard graph coloring problem [134] with appropriate coloring constraints to tackle the communication interferences. However, in [71], it has been shown that this simple reduction of TDMA time slot assignment problem actually does not consider some additional opportunities of time division, and hence even a solution to minimum coloring does not necessarily produce the best result for TDMA. In that sense, the problem of optimum TDMA time slot assignment is harder than optimizing the number of colors. Many such problems are NP-complete and the approximation algorithms proposed so far are mostly not distributed [23, 89, 133].

For static wireless networks, various centralized algorithms for scheduling have been proposed in [21, 51, 133, 147]. In [133], it was shown that the worst-case number of slots required per frame is $\theta\rho$ for broadcast scheduling and $\theta^2\rho$ for link scheduling, where ρ is the maximum node-degree and θ (thickness of the graph) is the minimum number of planar graphs into which the given graph can be partitioned. It actually improves the bounds presented in [5] and [135]. In [147], a broadcast scheduling algorithm is presented that attempts to maximize the network throughput. In [51], a joint routing, link-scheduling and power control scheme has been presented. A mathematical programming formulation for resource optimization in STDMA has been presented in [21].

The broadcast scheduling methods proposed in [66] shows that in terms of average delay and throughput the interference-based scheduling with global interference knowledge performs better than graph-based scheduling with limited interference

knowledge. The problem of throughput-optimal link scheduling subject to K-hop interference models (i.e., no two links within K hops can transmit simultaneously) is studied in [149]. This problem can be mapped to a Maximum Weighted Matching (MWM) problem subject to the K-hop interference constraints. The classical MWM problem is polynomial for $K = 1$ and NP-Hard for $K > 1$, and cannot be approximated within a factor that grows polynomially with the number of nodes. Interestingly, it is shown that the polynomial time approximation is possible for a wide range of wireless networks. The greedy matching algorithm in [149] provides a constant factor approximation to the scheduling problem for all K. But all these works present centralized algorithms, and hence are not suitable for ad hoc networks.

In fact we need distributed algorithms based on local information only to cope up with the mobility, transient fault tolerance, scalability and severe energy crunch in ad hoc networks. Some distributed STDMA algorithms have been presented in [71, 90, 91]. Authors in [90, 91] developed a self-stabilizing TDMA algorithm for a restricted class of network topology, namely the grid topology, where each node knows its location. In [71], authors present a self-stabilizing algorithm for renaming and coloring of a base model that ensures only probabilistically conflict-free communication using a constant number of colors ($\leq \Delta^6$), where Δ is the maximum degree of a node in the network graph. In [41], the STDMA broadcast scheduling is proposed using the decentralized dynamic channel allocation.

However, the use of TDMA, in fact is targeted to guarantee conflict-free communication in ad hoc networks to conserve energy. So, developing a localized algorithm for TDMA scheduling to keep the frame length minimum is still a challenging task to the network designers.

3.2 Our Contributions

Assuming that the nodes are distributed over a two-dimensional region, and the interference range is the same as the transmission range, we establish an upper bound (L) on the length of a frame for STDMA scheduling. The upper bound is of linear order of the maximum degree Δ of the topology graph. We presented a distributed algorithm to assign time slots to the nodes for broadcast scheduling following STDMA technique. We improve our slot assignment algorithm to allocate multiple slots to nodes within a frame to enhance the utilization of slots. Also, we propose algorithms to adapt with incremental changes in the topology due to node

movements and/or node failures. Simulation studies have been done to show that our algorithm of assigning multiple slots enables the nodes to transmit at higher rate when data is available, else they can remain in the sleep mode in idle slots, saving energy significantly.

3.3 New Upper Bound on Frame Length

Applying the concepts of simple geometry over the random distribution of ad hoc nodes, here we establish an upper bound on the cardinality of $N^2(v_i)$, $\forall v_i \in V$, based on the node-degree bound Δ. This, in turn, limits the maximum number of slots required in a frame for broadcast scheduling.

Lemma 3.1 *Given a topology graph $G(V, E)$, any two nodes v_i and v_j of V can transmit during the same time slot without any collision if $d(v_i, v_j) \geq 3$.*

Proof: Let the nodes v_i and v_j transmit during the same time slot. If $d(v_i, v_j) = 1$, both v_i and v_j will detect collision (interference range is equal to the transmission range).

If $d(v_i, v_j) = 2$, there exists at least one node v_k, where $v_k \in N^1(v_i) \cap N^1(v_j)$. So, v_k will detect a collision.

If $d(v_i, v_j) \geq 3$, any path between these two nodes will contain at least two intermediate nodes, say, v_{ki} and v_{kj} such that $d(v_i, v_{ki}) = d(v_j, v_{kj}) = 1$. If v_i and v_j transmit during the same time slot, v_{ki} can receive the packet from v_i and v_{kj} can receive the packet from v_j without any collision. If there exist more than two intermediate nodes, then also there will be no collision. This proves that v_i and v_j can transmit during the same slot without collision if $d(v_i, v_j) \geq 3$. ■

Lemma 3.2 *For any topology graph $G(V, E)$, $|N^2(v_i)| = min\{(19\Delta - 18), (\Delta^2 + 1)\}$, for the maximum node degree $\Delta \geq 2$.*

Proof: Let a node v_c be placed at the origin $C(0, 0)$ as shown in Figure 3.1(a). Two circles A and B are drawn with radius r and $2r$, respectively, centred at C. r is the range of each node. Assuming that all the nodes are distributed over a 2-D region, it is evident that all 1-hop away neighbours of v_c will lie within the circle A, and all 2-hop away neighbours will lie within the annular region bounded by the circles A and B. It is also assumed that the degree of each node is bounded by $\Delta \geq 2$.

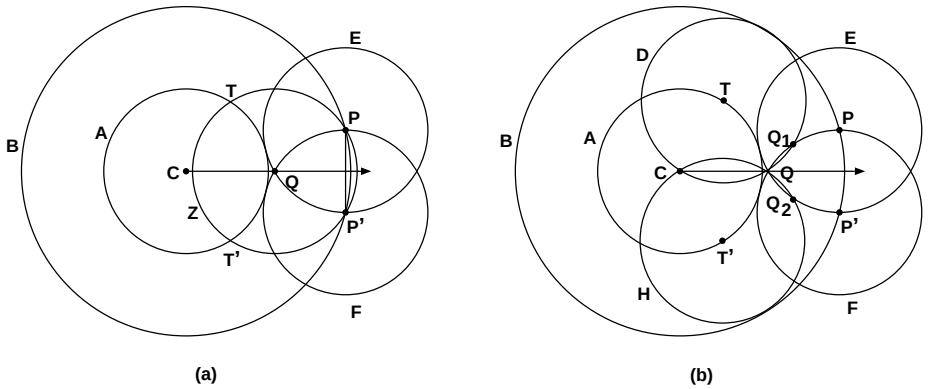


Figure 3.1: Region boundaries of Lemma 3.1

PP' is a chord of circle B of length r perpendicular to x-axis such that x-axis bisects the line PP'. Hence, the co-ordinates of P and P' are $(\frac{\sqrt{15}}{2}r, \frac{r}{2})$ and $(\frac{\sqrt{15}}{2}r, -\frac{r}{2})$ respectively. Two circles E and F each with radius r are drawn centred at P and P' respectively. E and F intersect each other on the x-axis at $Q = (\frac{\sqrt{15}-\sqrt{3}}{2}r, 0)$ inside the circle B. Consider the circle Z of radius r centred at Q intersecting the circle A at two points T and T' with coordinates $(\frac{\sqrt{15}-\sqrt{3}}{4}r, \sqrt{\frac{\sqrt{45}-1}{8}}r)$ and $(\frac{\sqrt{15}-\sqrt{3}}{2}r, -\sqrt{\frac{\sqrt{45}-1}{8}}r)$, respectively, as shown in Figure 3.1(a).

Another circle D of radius r is drawn centred at T. It intersects the circle F at two points Q and $Q_1(\frac{\sqrt{15}+\sqrt{3}}{4}r, \frac{\sqrt{3\sqrt{5}-1}-\sqrt{2}}{2\sqrt{2}}r)$ shown in Figure 3.1(b). Similarly, the circle H of radius r centred at T' intersects the circle E at two points Q and $Q_2(\frac{\sqrt{15}+\sqrt{3}}{4}r, -\frac{\sqrt{3\sqrt{5}-1}-\sqrt{2}}{2\sqrt{2}}r)$ as shown in Figure 3.1(b). The angle α subtended by ray $\overrightarrow{CQ_1}$ with the x-axis equals $\tan^{-1}\frac{\sqrt{2}(\sqrt{\sqrt{45}-1}-\sqrt{2})}{\sqrt{15}+\sqrt{3}} \approx 13.82°$.

Knowing the points Q, Q_1 and the angle α, the annular area between the circles B and A is partitioned into a set of regions as shown in Figure 3.2. Two circles C_I and C_O of radius CQ and CQ_1 are drawn centred at C. Taking a point G_0 on the circle A, the line segment $\overrightarrow{CG_0}$ is drawn that intersects the circles C_I, C_O and B at points K, L and M respectively. Consecutively, 26 rays are drawn $CG_1, \cdots, CG_{26}$ on both sides of CG_0, each making an angle α with the previous one as shown in Figure 3.2. Using the points of intersection of the rays with the circles A, C_I, C_O and B, we mark 13 regions of E type and 13 regions of I type as shown in Figure 3.2. Let these regions be denoted as E_i and I_i, $0 \leq i \leq 12$ respectively. Finally there will be a residual area R making an angle β of approximately 0.82° (less than 1°) as shown in Figure 3.2. These E and I regions together with R cover the whole region where the 2-hop away nodes of v_c may lie. Let $n(x)$ denote the number of nodes in

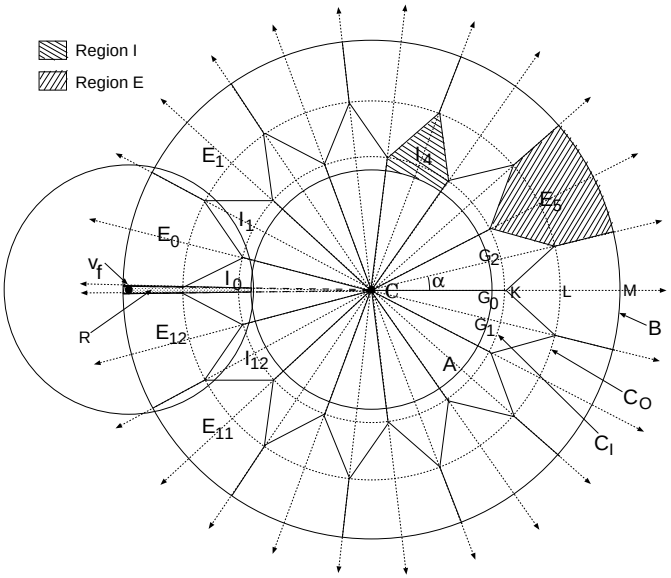


Figure 3.2: The regions E and I to partition the annular region between circles A and B

a region x.

Based on the construction of E and I regions, following two observations can be made:

(a) all nodes in any E_i (I_i) region are connected to each other, and

(b) a node in I_i is connected to all nodes in the neighbouring regions I_{i-1} and I_{i+1}, $\forall i = 0, \cdots, 12$ ($I_{-1} \equiv I_{12}$ and $I_{13} \equiv I_0$). It implies that $\sum_{i=0}^{12} n(I_i) < 6(\Delta-1)$, as degree of any node is bounded by Δ, assuming that each node in I regions is a 2-hop away node of v_c, i.e., connected to at least one node in $N^1(v_c)$.

Let v_f be one of the farthest 2-hop away nodes of v_c. Without loss of generality, let us assume that v_f is placed anywhere in the residual region R, as shown in Figure 3.2. Then, node v_f remains connected to all nodes within the regions E_0, E_{12} and R. It implies that $n(E_0)+n(E_{12})+n(R) \leq (\Delta-1)$. So, $\sum_{i=0}^{12} n(E_i) \leq 12(\Delta-1)$.

Hence, at most $18(\Delta-1)$ nodes can exist as 2-hop away neighbours of v_c assuming that each one is connected to v_c via at least one 1-hop neighbour within circle A. Therefore, $|N^2(v_c)| = (19\Delta - 18)$. It will be true for any node. However, for any network with a node-degree bound Δ, it is evident that $|N^2(v_i)| \leq (\Delta^2+1)$. Hence, it proves that $|N^2(v_i)| = min\{(19\Delta - 18), (\Delta^2 + 1)\}$, for any $v_i \in V$, with a node degree bound $\Delta \geq 2$. ■

Theorem 3.1 *For a topology graph with a node degree bound $\Delta \geq 2$, min $\{(19\Delta -$*

18), $(\Delta^2+1)\}$ *slots are sufficient in a frame to assign at least one slot to each node for collision-free transmission.*

Proof: From Lemmata 3.1 and 3.2, it is evident that if each 2-hop neighbour of any node v_i is assigned a distinct slot in the frame, collision-free communication can be ensured. Hence, from Lemma 3.2, $min\{(19\Delta-18), (\Delta^2+1)\}$ slots are sufficient in a frame. ■

The *arboricity* of an undirected graph is the minimum number of forests into which its edges can be partitioned. Equivalently it is the minimum number of spanning forests needed to cover all the edges of the graph. The arboricity of a graph is a measure of how dense the graph is: graphs with many edges have high arboricity and graphs with high arboricity must have a dense subgraph. The *thickness* (θ) of a graph is the minimum number of planar subgraphs into which its edges can be partitioned.

Theorem 3.2 *For broadcast scheduling, the upper bound on the STDMA frame length* $min\{(19\Delta-18), (\Delta^2+1)\}$ *proposed here is better than bound* $\theta\Delta$ *proposed in [133].*

Proof: As any planar graph has arboricity three, the thickness θ of any graph is at least equal to a third of the arboricity, and at most equal to the arboricity. The arboricity of a graph with n vertices and m edges is at least $\lceil \frac{m}{n-1} \rceil$. So, the arboricity of a dense graph can be $O(\Delta)$. So, θ is $O(\Delta)$ for dense graph. Thus, the bound $\theta\Delta$ can be $O(\Delta^2)$ for dense graph. This proves the theorem. ■

3.3.1 Time Frame Structure

The time is divided into slots of a fixed interval of τ units. Each slot contains a short header H of τ_h units ($\tau_h \ll \tau$). The header is generally used for control purposes. The rest of the slot of duration $T = (\tau - \tau_h)$ is used for message transmission. We need to synchronize the slot boundaries, but not the frame boundaries. The synchronization is to be done by using some distributed synchronization techniques [55] at regular intervals. Figure 3.3 shows the time frames at two different nodes v_i and v_j with different frame boundaries. It has been assumed that a frame contains exactly $L = min\ \{(19\Delta-18), (\Delta^2+1)\}$ slots.

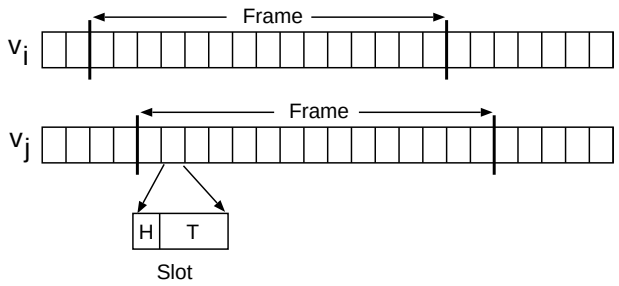


Figure 3.3: The time slots and frames in two nodes v_i and v_j

3.4 Scheduling Algorithm for Static Networks

The proposed methodology consists of two phases. First phase is initialization where the nodes discover their 2-hop neighbours. On completion of the initialization procedure each node starts slot assignment procedure.

3.4.1 Neighbourhood Discovery

As the ad hoc nodes wake up they try to discover their neighbours for a predefined time t_d. Any node v_i may transmit a HELLO message with a small probability p, during the slot-header H of any time slot and listens for the remaining duration T of each slot. In the HELLO message, node v_i sends the IDs of all its 1-hop neighbours it has already discovered. Each node v_i repeatedly switches between the roles of sender and receiver randomly during this phase and gather knowledge about its 2-hop partial graph $PG^2(v_i)$.

Procedure NeighbourhoodDiscovery(v_i, t_d)
begin
$State = INIT$; $N^1(v_i) = \phi$
for slot=1 to t_d
  if $rand(0,1) < p$
    transmit HELLO message in header
  else wait to receive message
    if HELLO message received from v_j
      $N^1(v_i) = N^1(v_i) \cup \{v_j\}$; update $PG^2(v_i)$
if No HELLO message is received during t_d
  $State = NEW$
  Call Procedure New-Node()
end

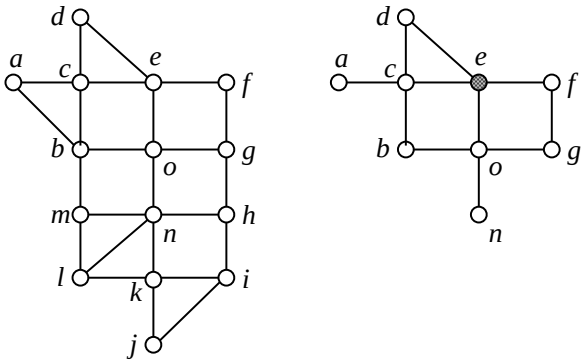


Figure 3.4: (a) Topology graph and (b) 2-hop partial graph $PG^2(e)$

3.4.2 Single Slot Assignment

Definition 3.1 *The slot assignment vector S_i of a node v_i is a vector of length L where, $S_i(j) = ID(v_k)$, if node $v_k \in (N^2(v_i) \cup \{v_i\})$ uses slot j for transmission, otherwise $S_i(j) = \phi$, $1 \leq j \leq L$.*

Example 3.1 *Let us consider the node e of the network shown in Figure 3.4(a). Its 2-hop neighbour set is $N^2(e) = \{a, b, c, d, e, f, g, n, o\}$. Assuming $L = 16$, a possible slot assignment vector S_e is shown in Figure 3.5. It indicates that, node o transmits during the time slot 2 and so on.*

Definition 3.2 *The offset vector F_i^s with respect to a time slot s at a node v_i is a vector of length at most $(\Delta + 1)$ where each element of F_i^s is a tuple of the form (node, offset) computed from S_i and $N^1(v_i)$ as follows: $\forall v_j \in (N^1(v_i) \cup \{v_i\})$, if $\exists k$, such that $S_i(k) = v_j$ then $(v_j, k - s) \in F_i^s$.*

Example 3.2 *Figure 3.5 shows F_e^{12}, the offset vector at node e with respect to current time slot 12. The F_e^{12} contains 5 elements, one for each neighbour of e. The element $(o, -10)$ in F_e^9 denotes that node o transmits on a slot which precedes 10 slots from the current slot 12 in node e.*

Definition 3.3 *A slot assignment is consistent at node v_i if the slot selected by node v_i (i.e., slot(v_i)) is not used by any node $v_i \in N^2(v_i)$. The slot assignment of topology G is consistent if the slot assignment is consistent at every node in G.*

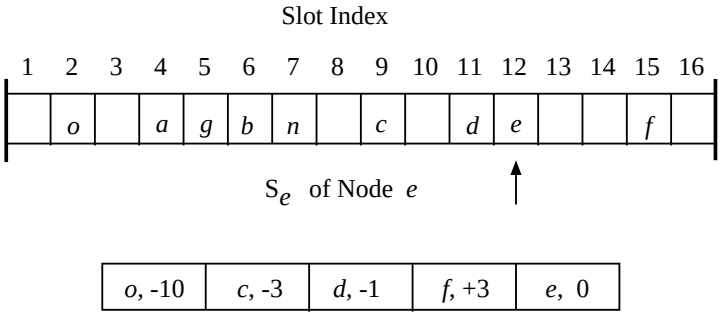


Figure 3.5: The slot assignment vector and offset vector at node e

During this phase, each node v_i can assign a slot if and only if all $\{v_j : ID(v_j) > ID(v_i), v_j \in N^2(v_i)\}$ have already selected their slots. After a node selects its slot, it broadcasts its offset vector and a *Forward-Schedule Message (FSM)* asking a subset of its 1-hop neighbours in ascending order of their IDs to broadcast their offset vectors in successive slots, to inform the 2-hop neighbours who are still to select their slots. The details of the procedure are given below.

Procedure Single-Slot-Assignment(v_i)
begin
$assigned = false$; $ready = false$;
$hN = \{v_j : v_j \in N^2(v_i)$ and $ID(v_j) > ID(v_i)\}$ // List of higher neighbours of v_i
do in each slot s
  if $assigned = false$ and $hN = \phi$ then
    do ASSIGNING - select free slot, update S_i, F_i^s, FSM
      send *offset vector* and Forward-Schedule message
    $assigned = true$
  else
    if scheduled to FORWARD at slot s
      do FORWARDING - send *offset vector* F_i^s
    else
      do WAITING - listen for incoming message
      if received *offset vector* message from v_j
        update S_i and hN
      if received Forward-Schedule message
        schedule to send *offset vector* message if required
until S_i contains an entry for each $v_j \in N^2(v_i)$
$ready = true$;
end

Lemma 3.3 *A node sends at most* $(\Delta + 2)$ *messages each of length* $O(\Delta)$ *during slot assignment.*

Proof: The ASSIGNING operation is performed at any node only once, so at most one Forward-Schedule-Message (FSM) is generated by any node. Each FSM message contains an ordered set of at most Δ nodes who need to forward their *offset vector*s. Thus, any node sends its *offset vector* at most $(\Delta + 1)$ times; just after its own slot assignment and at most Δ times after slot assignment of each of its neighbours. Again, both the *offset vector* and FSM message contain $O(\Delta)$ elements. So, each node sends at most $(\Delta + 2)$ messages of length $O(\Delta)$. ■

Theorem 3.3 *The time and message complexities of the single slot assignment algorithm are $O(n)$ and $O(\Delta n)$ respectively.*

Proof: The proof directly follows from steps of the algorithm and Lemma 3.3.

3.4.3 Multiple Slot Assignment

Since the frame length is equal to the maximum number of 2-hop neighbours possible for a node, it is obvious that after single slot assignment, each node will find many free slots in a frame, which are not assigned to any of its 2-hop neighbours. It will result in poor utilization of slots. To improve it, on completion of *Single Slot Assignment* algorithm each node executes the *Multiple Slot Assignment* procedure, exactly in the same way as it had been done in the *single slot assignment algorithm*, with the difference that in its turn, depending on the number of its 2-hop neighbours each node v_i may reserve at most $\lfloor \frac{L}{|N^2(v_i)|+1} \rfloor$ slots, if available.

However, the number of extra slots may be determined following some other criteria also, such as demand of a node, QoS requirement, or time delay, etc. It will obviously help to enhance the throughput of the network [147].

3.5 Simulation Studies

Simulation studies have been done to measure the performance of multiple slot assignment algorithm under the simulation environment presented in Section 2.5. The performance is represented in terms of a parameter called *frame utilization (U)*. For each node v_i a frame utilization factor u_i is defined as the ratio of the number of distinct slots reserved by itself and its 2-hop neighbours in the frame. For a given topology, the percentage frame utilization factor is defined as, $U =$

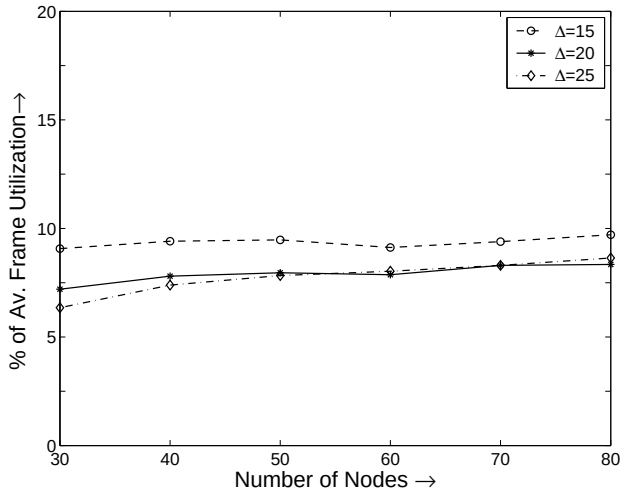


Figure 3.6: Single slot assignment: variation of frame utilization with number of nodes

$\frac{\sum u_i}{n} \times 100$. Simulation is carried on random graphs to show the variation of average percentage of frame utilization with number of nodes (n) using the degree bound Δ as a parameter. The results shown in Figure 3.7 reveals that about $80 - 94\%$ slots are occupied on an average following the technique of multiple slot assignment whereas for single slot assignment the frame utilization factor varies in the range of $6 - 10\%$, as shown in Figure 3.6.

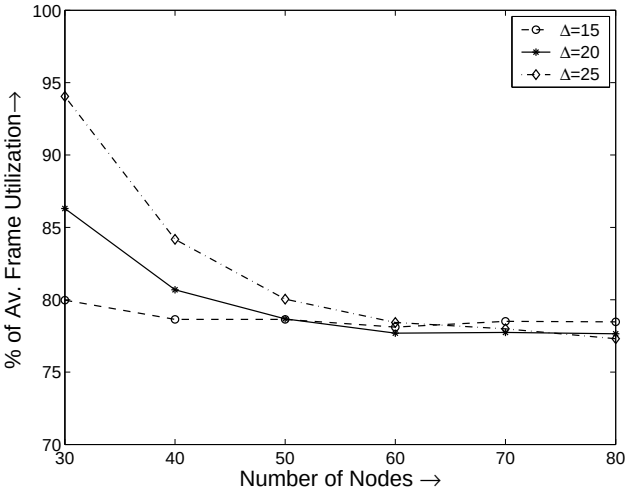


Figure 3.7: Multiple slot assignment: variation of frame utilization with number of nodes

Also, with multiple slot assignment, the variation in the number of nodes using a given slot (with slot-ID's) is shown in Figure 3.8 for a typical case with $n = 90$ and

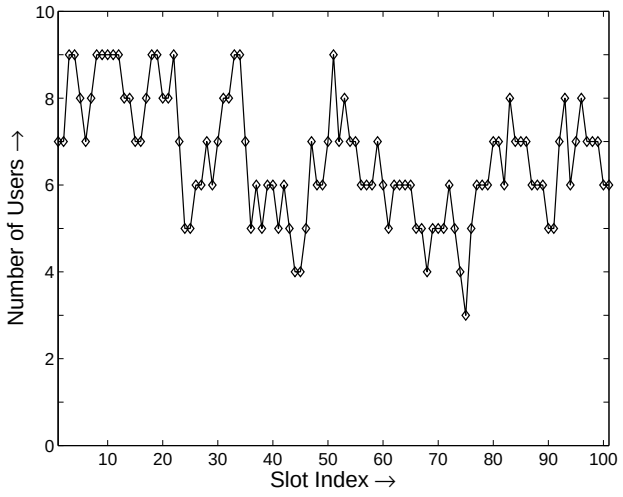


Figure 3.8: Number of users assigned on different slots for $n = 90$ and $\Delta = 10$

$\Delta = 10$. It shows that for the topology under consideration the number of nodes using same slot varies from 3 to 9. Figure 3.9 shows the distribution of number of slots vs. node ID, for a typical graph with $n = 100$ and $\Delta = 25$. It shows that the number of slots per node varies in the range of 6 to 22.

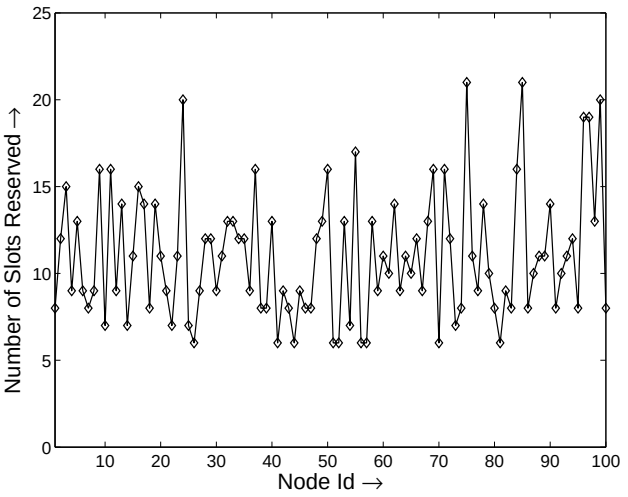


Figure 3.9: Number of slots reserved by different users for $n = 100$ and $\Delta = 25$

Therefore, the simulation results reveal the fact that multiple slot assignment can improve the network throughput significantly if we can distribute the free slots appropriately according to the demand.

3.6 Scheduling for Dynamic Topology

In ad hoc networks, the topology may change due to the following reasons: (a) a new node is switched on, (b) a node is switched off, or (c) a node moves from one location to another. Case (c) can be considered as a combination of (b) and (a).

Obviously, if a node is switched off, its neighbours just make its reserved slots free, keeping the remaining assignments unaltered. But if there is an addition of a new node, slot assignments may have to be rearranged in the 2-hop neighbourhood of the new node to avoid collision. The proposed algorithm is augmented to adapt with limited perturbations in the topology assuming that any node may have at most one perturbation (addition of a single new node) at a time within its 2-hop neighbourhood. Some collisions may occur during the rearrangement phase. However, after some message exchanges among the 2-hop neighbours of the new node, finally the slots are rearranged, if necessary, and makes all communications conflict-free again.

A single slot assignment always guarantee an available slot for a new node. However, in case of multiple slot assignments, it is not guaranteed that a free slot will be always available. Therefore, as soon as a new node wants to join the STDMA schedule, all its 2-hop neighbours should release excess slots, keeping only a single slot for each. After the new node occupies a single slot, all the perturbed nodes are allowed to execute the *Multiple-slot-assignment* procedure to reserve additional slots again to improve the slot utilization, and to enhance the throughput.

The detailed procedures are described below.

3.6.1 Node Addition

Let the new node be v_{new}. It proceeds in the following way:

- v_{new} sends a NEW message in the header of each slot during its first frame, and listens during the rest periods of the slots.
- on receiving a NEW message, each static neighbour $v_i \in N^1(v_{new})$ withholds its normal operation; modifies its *offset vector* keeping only the slot-ID's reserved by itself and each of its 2-hop neighbours by the *single slot assignment* algorithm, and broadcasts a HOLD message.
- on receiving HOLD message from its 1-hop neighbour v_i,

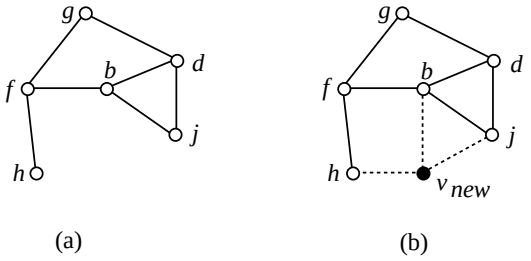


Figure 3.10: Collision at new node v_{new}

a) v_{new} updates S_{new} and PG^2_{new}, and

b) each static neighbour $v_j \in N^1(v_i)$ withholds its normal operation and modifies its own *offset vector*.

- if v_{new} does not detect any collision in the whole frame, it selects a free time slot, else, it waits for the colliding nodes to reassign their slots, and finally reserves a single free slot for itself, and intimates all its 2-hop neighbours.

The situation becomes complex when collision is detected at v_{new} while receiving the *offset vector* from any neighbour. One such case is shown in Figure 3.10. Let in the initial topology graph shown in Figure 3.10(a), $\delta(h, j) > 2$, and they use same time slot for transmission. Now a node v_{new} appears and the nodes h and j becomes 2-hop neighbours of each other through v_{new} as shown in Figure 3.10(b). Therefore, when the two nodes transmit simultaneously, v_{new} will detect collision. So, the slots of h and j become inconsistent, and rearrangement is necessary.

The new node v_{new} on detecting some collisions modifies the subsequent NEW messages by adding the offset of the slots where it has detected collisions. This allows the set of colliding nodes to identify themselves. Each colliding node $v_i \in N^1(v_{new})$ proceeds in the following way:

- sends INCONSISTENT message to all nodes in $N^2(v_i)$. The slot assignment of all nodes in $N^2(v_i)$ should remain unaltered until slot reassignment of v_i is over.
- broadcasts a SEARCH message with the ID of v_i and the ID of v_{new}
- waits for sufficient time to listen to all incoming SEARCH messages
- collects the IDs of all colliding nodes from SEARCH messages, arranges in descending order, and executes *single slot assignment* algorithm to reserve single slot in its appropriate turn

- sends CONSISTENT message to all nodes in $N^2(v_i)$ on completion of reassignment

The procedures followed at the new node v_{new} and at any node $v_i \in N^1(v_{new})$ are given below.

Procedure New-Node(v_{new})
begin
$C = \phi$ // C: set of colliding slots
for *slot* = 1 to L
  send NEW and C in *header*
  listen during T of each slot
  if collision is detected
    $C = C \cup slot$
  else
    offset vector is received from v_i
    update $N^1(v_{new})$, $PG^2(v_{new})$ and S_i
if $C \neq \phi$ // collision is detected
  wait until colliding neighbours reassign slots
do ASSIGNING and send *offset vector* to neighbours
end

Theorem 3.4 *For addition of a new node, single slot reassignment procedure completes in (i)* $(L+1)$ *slots if no collision is detected at the new node, and (ii) at most* $((\lambda+1)L + 2\Delta + 2)$ *slots when collision is detected, where* λ *is the diameter of the topology graph* G *in hops.*

Proof: First of all, it is guaranteed that the NEW messages are received by all nodes within the range of the new node, v_{new}, as each of them listens during the H period of the slot assigned to it, in every frame. This also ensures that v_{new} will eventually receive *offset vectors* from all nodes in $N^1(v_{new})$.

Let us assume that there is no collision during the first L slots of new node v_{new}. This implies that the slot assignment in G is consistent in presence of new node v_{new}. Also, v_{new} has received the slot assignment of nodes in $N^2(v_{new})$ via *offset vector* from $N^1(v_{new})$. Then, v_{new} can immediately select a free slot and can finish the slot assignment procedure in the next time slot. Therefore, the whole procedure completes in $(L+1)$ slots.

When collisions are detected at v_{new}, the colliding nodes identify themselves within the first $(L+1)$ slots. The SEARCH message initiated at a node v_i is received by all other colliding nodes within at most λL slots, where λ is the diameter

of the topology graph. Therefore, the colliding nodes can reassign their time slots after $((\lambda+1)L+1)$ slots. Even if all 1-hop neighbours (at most Δ) of v_{new} need reassignment, their assignment can be finished within 2Δ slots since all other 1-hop neighbours of any colliding node v_i remain silent until v_{new} selects its own slot. IT implies that the slot reassignment procedure completes within $((\lambda+1)L+2\Delta+2)$ slots in the worst case. ■

Procedure OnReceivingNewMsg(v_i)
begin
$ready = false$; $N^1(v_i) = N^1(v_i) \cup \{v_{new}\}$
during T of $slot(v_i)$
  send *offset vector* to v_{new} and HOLD message to $N^1(v_i)$
wait to listen from v_{new} for next L slots
if collision at $slot(x_i)$
  $assigned = false$; $consistency := false$;
  $R = \{v_i\}$ // R: set of nodes to reassign slots
  send INCONSISTENT to v_j, $\forall v_j \in N^2(v_i)$
  wait until no NEW message
  broadcast $SEARCH(v_i, v_{new})$ message
  do for next D frames
    if received SEARCH message from v_j
    $R = R \cup \{v_j\}$
    update $PG^2(v_i)$ by adding v_j as a neighbour of v_{new}
  $hN = \{v_j : v_j \in R,\ ID(v_j) > ID(v_i)\}$
  do in each slot of a frame
    if $hN = \phi$
      do ASSIGNING - select first free slot and send *offset vector*
      $assigned = true$; $consistency := true$;
      send CONSISTENT message to all nodes in $N^2(v_i)$
    else
      do WAITING - listen for incoming message from v_{new}
      if received *offset vector* from v_{new}
        update S_i^2 and hN
wait until v_{new} assigns slot
$ready = true$ and send READY to $N^1(v_i)$
end

3.6.2 Node Deletion

Before the node v_i is switched off, it sends a *switching off* message in its slot. Knowing that v_i is getting switched off all nodes in $N^1(v_i)$ mark the slots assigned to v_i as available and inform their 1-hop neighbours to do the same thing, so that all 2-hop neighbours of v_i can make those slots free.

Summary

For a random distribution of ad hoc nodes over a two dimensional region, we proved that the upper bound on frame length (L) is of linear order of Δ, where Δ is the maximum node degree. This frame length guarantees at least one conflict-free slot to each node, but in practice it allows multiple slots for each node. The proposed slot assignment algorithm that requires only Δ messages per node to converge to a schedule can be used even in energy starved condition of the network. An $O(\lambda L)$ distributed algorithm is developed to adjust the time slots locally when topology changes due to mobility. As L reduces, such perturbation in the network will converge faster than expected.

Simulation results show that the use of multiple slot assignment enhances the slot utilization appreciably.

It is to be noted that all these activities like single and multiple slot assignments, and the readjustment of slots due to node mobility are done based on limited local knowledge of 2-hop neighbourhood only.

Chapter 4

Fixed Round Connected Dominating Set Construction

In wireless ad hoc networks, node mobility in absence of any hierarchy causes occasional route breakages and hence requires frequent topology discovery making it difficult to devise energy efficient routing algorithms. Hierarchical routing backbone can play an important role in making the maintenance and scaling of the network easy [102]. Moreover, introducing hierarchy, although virtual, makes the location update and location finding in ad hoc networks, easier in general. Connected Dominating Set (CDS) has been emerged as an effective hierarchical routing backbone in mobile ad hoc networks.

A Dominating Set (DS) of a graph $G = (V, E)$ is a subset of nodes V', $V' \subseteq V$, such that each node in $V'' = V - V'$, termed as a non-dominating node, is adjacent to some dominating node in V'. A CDS is a DS that induces a connected subgraph of G. A minimum DS or CDS (MDS or MCDS) is a DS or CDS with the minimum cardinality. The (connected) domination number (d) of G is the number of vertices in the MCDS (MDS) [141]. Figure 4.1 shows an example of a topology graph $G(V, E)$, and a dominating set, a minimum dominating set and a connected dominating set of $G(V, E)$ respectively (shaded nodes are the dominating ones).

It is desirable to find an MCDS for the underlying topology to simplify the topology and routing management. However, finding an MCDS in a graph is NP-hard [43], and thus distributed algorithms for suboptimal solutions with faster convergence may be in demand for wireless ad hoc networks.

There are many advantages of using CDS as routing backbone. First it partitions the set of nodes into exactly two sets V' and V''. This limited number of sets permit an easier way to build and maintain the hierarchy. Secondly, it maps the routing process for the whole network to a smaller sub-network induced by the CDS

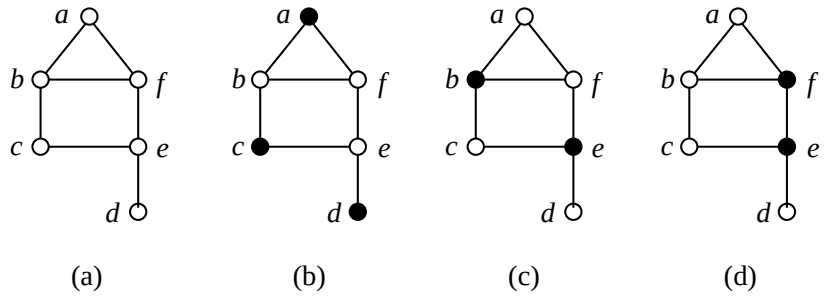


Figure 4.1: Various configurations: (a) a topology graph, (b) a dominating set, (c) a minimum dominating set and (d) a minimum connected dominating set

only. It also implies that only dominating nodes need to keep routing information. Most important is that any node in the network can reach any other node via the dominating nodes only. Moreover, there is no need to recalculate routing tables as long as node mobility do not alter the sub-network induced by the CDS. Additionally, it can also be used for multi-casting [171], broadcasting [181] and connectivity management.

Chapter Organization

This chapter is organized as follows. Section 4.1 overviews the related works on CDS construction and its use as a routing backbone in ad hoc networks. In Section 4.2, we discuss the distinct features of our study on CDS. In Section 4.3, we prove some interesting properties of dominating sets which form the basis for our CDS construction algorithm. Section 4.4 describes the CDS construction algorithm with the details of the three phases of our algorithm including the completeness proof and complexity analysis. Simulation results are discussed in Section 4.5. The chapter ends with a brief summary.

4.1 Background

Traditionally, the CDS problem is solved by approximating it to a *maximum leaf spanning tree* (MLST) problem [58]. An MLST is a spanning tree that has the maximum number of leaves among all spanning trees of G. The *maximum leaf number* (l) of G is the number of leaves in the MLST. It has been shown in [53],

that $n = d + l$, where d is the dominating number. The maximum leaf spanning tree problem is NP-hard [61]. However, it can be approximated to within a factor of 2 in polynomial time [155]. Several NP-hard optimization problems, such as CDS, can be solved in polynomial time for graphs with bounded maximum leaf number [58]. So, finding the minimum connected dominating set is equivalent to finding a maximum leaf spanning tree. However, connected domination and MLST are not the same in terms of approximation algorithms. There exists an approximation for the minimum connected dominating set that achieves a factor of $2\log(d) + O(1)$, where d is the connected dominating number [69].

In [18] [52] [151], three-stage centralized CDS construction algorithms are presented. They first compute an approximation to Minimum Dominating Set (MDS) based on the generalized set cover problem. The three algorithms differ only in the weight parameter chosen to be used in the greedy heuristic to find the MDS. The second stage constructs a spanning forest where each tree is a union of stars centred around the nodes selected in the first stage. The third stage expands the spanning forest to form a spanning tree whose internal nodes are selected as a CDS. The approximation factor of this algorithm is $3H(\Delta)$, (where Δ is the maximum node degree and H is the harmonic function), which stems naturally from the use of set cover heuristics [42] in the first stage. However, these approximation algorithms are centralized and so they are not suitable for ad hoc networks.

The distributed algorithms proposed in [18] [52] [151] are of very high time and message complexities $O(n^2)$. In [163], it is also shown that the approximation factors of those algorithms lie between $(\frac{\log \Delta}{2} - \frac{1}{2})$ and $3H(\Delta)$.

The distributed greedy algorithm proposed in [173, 172] first finds a CDS and then prunes the set by removing locally redundant nodes preserving connectivity. Its time and message complexities are $O(\Delta^3)$ and $\Theta(m)$, where m is the number of edges in the topology graph, respectively as shown in [163]. The approximation factor of this algorithm is as high as $\frac{n}{2}$.

In general, starting from a minimal DS, if it is not connected, we can generate a CDS by adding new nodes to make the DS connected. An interesting question is when a minimal connected dominating set becomes a minimal dominating set. Let for a given graph G, the domination number (denoted by $\gamma(G)$), and the connected domination number (denoted by $\gamma_c(G)$), be the cardinalities of the smallest possible

DS and CDS, respectively. It has been shown in [164] that for general graph, $\gamma(G) \leq \gamma_c(G) \leq 3\gamma(G) - 2$.

It is evident that for a given graph $G(V, E)$, a minimal CDS will be a minimal DS, if and only if for each node v_i in CDS there exists at least one non-dominating node v_j which is not dominated by any node in the CDS other than v_i.

As for example, in Figure 4.1(d), $\{e, f\}$ is a minimal CDS which is also a minimal DS. For the dominating node e, the non-dominating node d is dominated by e only. Similarly for the other dominating node f, there exist two nodes a and b, which are dominated by node f only.

Instead of starting from a DS, another approach for finding a CDS is to approximate it by a maximal independent set (MIS) problem and then connecting the nodes using some heuristics [34, 64, 103, 159]. The MIS is obviously a DS since addition of any node to the set implies appearance of an edge in the MIS, i.e. each remaining node has at least one adjacent node in the MIS. Also, interestingly, the MIS is a minimal DS, since if we remove any node v_i from the MIS, there will be no node in the MIS that can dominate v_i. Hence the MIS is an MDS. For a given graph $G(V, E)$ the relation between the cardinality of the MIS ($\alpha(G)$) and the connected domination number ($\gamma_c(G)$) plays an important role to evaluate the performance of the heuristics applied to connect the nodes of the MIS. It has been proved that for unit disc graphs, $\alpha(G) \leq 3.8\gamma_c(G) + 1.2$. But unfortunately, for the MIS, the nodes are isolated. Therefore, to generate a CDS from it, additional nodes are always to be included and the minimal DS will not be a minimal CDS, excepting some trivial cases. For example, for a complete graph, or for a star graph, the MIS contains a single node and it can serve as a CDS also.

However, since every maximal independent set is a minimal DS which is easy to construct, several attempts have been made to construct CDS starting from MIS. There are three such notable contributions in the literature. In all these algorithms a maximal independent set (MIS) is computed, which acts as the set of cluster-head nodes. A node is a *border* node if it is not a cluster head and there are at least two cluster-heads within its 2-hop neighbourhood. The cluster-heads are selected by some heuristic ranking of nodes that gives rise to a total ordering of all nodes. Three ranking heuristics are used: the ID only [64], an ordered pair of degree and ID [34], and an ordered pair of degree and location [158]. All these algorithms are of

high implementation cost in terms of message complexity and/or time complexity with logarithmic approximation factor [64], and linear approximation factors [103], [158] respectively.

In [163], it has been shown that any distributed algorithm for nontrivial CDS requires at least $O(n \log n)$ messages, where the message length is of the order of the number of bits representing the node IDs. In [163], a distributed algorithm is also presented that has an approximation factor of at most 8. The time and message complexities of the algorithm are $O(n)$ and $O(n \log n)$ respectively. The $O(n)$ time complexity limits its applicability to ad hoc networks with large n.

4.2 Our Contributions

We first prove some interesting properties of dominating sets, which are in general useful and form the basis of our CDS construction algorithm. We propose a three phase distributed algorithm to compute a CDS in an ad hoc network with the objective to minimize the size of the CDS. In this process, we explore some interesting properties of dominating sets in general. Our strategy of selecting a dominating node is based on coordinated decision of its neighbours and not just on the node's local computation. Through simulation it is found that this approach produces CDS that are, in general, smaller in size, than that achieved by previously known algorithms. Moreover, these properties also help us in choosing a better candidate for CDS. It also helps the algorithm to converge faster. The algorithm has $O(1)$ time complexity (exactly 6 rounds of message exchanges) and $O(n)$ message complexity.

4.3 Properties of Dominating Set

We first present some definitions and prove relevant properties of DS and CDS which are used in our proposed distributed algorithm. Since we are interested in distributed algorithms based on local information of the network, we develop some new concepts on DS and CDS based on 2-hop neighbourhood information of a graph.

Definition 4.1 *Given a (connected) dominating set V' of a graph $G(V, E)$, the corresponding (connected) dominating graph $DG_{V'}(V, E')$, $E' \subseteq E$, induced by V'*

is a subgraph of G containing all the edges of G excluding those between any two non-dominating nodes in $V - V'$.

A dominating graph may be either connected or disconnected. However, when V' is a CDS of the graph $G(V, E)$, $CDG(V')$ is always a connected subgraph of G. Figure 4.2 shows a dominating graph and a connected dominating graph corresponding to the topology graph of Figure 4.1(a).

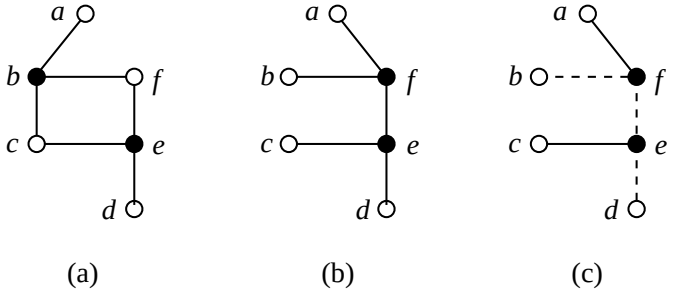


Figure 4.2: Various graph configurations: (a) a dominating graph, (b) a connected dominating graph and (c) *D-path* between the nodes b and d

Definition 4.2 *Given a graph $G(V, E)$ with a dominating set V', a path P between two nodes v_i and v_j is called a D-path if all intermediate nodes are dominating nodes in V'. The nodes v_i and v_j are said to be D-path connected (or, v_i is D-path connected to v_j). We also define the D-distance, $dist_{V'}(v_i, v_j)$ between two nodes v_i and v_j as the length of the shortest D-path between them. The D-diameter, $\lambda_{V'}$, of G is the maximum length of the shortest D-paths between all pairs of vertices in G, i.e., $\lambda_{V'} = max\ \{dist_{V'}(v_i, v_j), \forall v_i, v_j \in V\}$.*

Corresponding to connecting dominating graph (CDG) shown in Figure 4.2(b), Figure 4.2(c) shows a *D-path* between two nodes b and d. By definition, a CDG corresponding to a CDS, ensures that there exists a *D-path* between any pair of nodes. However, this is not always true for any DG. For example, we see in Figure 4.2(a) that although the dominating graph is a connected subgraph of the corresponding topology graph in Figure 4.1(a), there exists no *D-path* between the nodes b to e or b to d. So, the *D-path* connectivity among all pairs of nodes can serve as a terminating criterion for the construction of a CDS.

Lemma 4.1 *For a dominating set V' of $G(V,E)$, if each node $v_i \in V'$ is D-path connected to all other nodes $v_j \in V$ then V' is a connected dominating set.*

Proof: Let V' be a dominating set of $G(V,E)$ such that each node in V' is *D-path* connected to all other nodes in V. To prove that V' is a CDS, we just need to show that each v_k in V is *D-path* connected to each node $v_i \in V$. If $v_i \in V'$, it is true. If $v_j \notin V'$ then there exists at least one node $v_j \in V'$ such that $v_j \in N^1(v_i)$. Then, v_i is *D-path* connected to all nodes in V through v_j (i.e., v_i to v_j and then from v_j to all other nodes in V). This proves the lemma. ■

Definition 4.3 *Let V' be any DS of $G(V,E)$ and $v_i \in V'$. The set of 1-hop dominating neighbours of v_i, denoted as $DN_D^1(v_i)$, is the subset of V', contained in $N^1(v_i)$ (i.e., $DN_D^1(v_i) = N^1(v_i) \cap V'$). Similarly, the 2-hop dominating neighbours of v_i, denoted as $DN_D^2(v_i)$, is the subset of V' which is the union of 1-hop dominating neighbours of the neighbouring dominating nodes of v_i, i.e.,*

$$DN_D^2(v_i) = \{ \bigcup_{\forall v_j \in DN_D^1(v_i)} DN_D^1(v_j) \}$$

Definition 4.4 *Let V' be any DS of $G(V,E)$ and v_i be any node in V'. The 2-hop reachable neighbours of v_i, denoted as $RN_D(v_i)$, is the subset of nodes in $N^2(v_i)$ where each node is D-path connected to v_i. The set of unreachable neighbours of v_i, denoted as $UN_{V'}(v_i)$, is the set where $UN_{V'}(v_i) = N^2(v_i) - RN_D(v_i)$.*

Let $V' = \{b, e\}$ be the DS shown in Figure 4.2(a). Then, $DN_D^1(e) = \{e\}$, $DN_D^2(e) = \{e\}$, $RN_D(e) = \{c, d, e, f\}$ and $UN_{V'}(e) = \{b\}$. However, if we consider the DS of Figure 4.2(b), we see that $DN_D^1(e) = \{e, f\}$, $DN_D^2(e) = \{e, f\}$, $RN_D(e) = \{a, b, c, d, e, f\}$ and $UN_{V'}(e) = \phi$.

Theorem 4.1 *For a dominating set V' of $G(V,E)$, if $UN_{V'}(v_i) = \phi, \forall v_i \in V'$ then V' is a connected dominating set of G.*

Proof: For a given dominating set V' of $G(V,E)$, let us assume that $UN_{V'}(v_i) = \phi, \forall v_i \in V'$. To prove that V' is a CDS, by Lemma 4.1, we just need to show that any $v_i \in V'$ is *D-path* connected to all other nodes in V. We will prove this by induction on hop distance h.

Let v_i be any node in V'. From the fact that $UN_{V'}(v_i) = \phi$, it is apparent that v_i is *D-path* connected to all nodes in $N^2(v_i)$.

Let v_i be *D-path* connected to all nodes in $N^h(v_i)$, where $h \geq 2$. Let v_j be any node in V such that, the hop distance, $\delta(v_i, v_j) = h+1$. Let $V'' = N^h(v_i) \cap N^1(v_j)$. First note that $V'' \neq \phi$, otherwise G is disconnected and $V'' \subset N^h(v_i)$. Moreover, for any node v_k, if $v_k \in N^{h-1}(v_i)$ then $N^1(v_j) \cap N^1(v_k) \subseteq V''$. Also, it is to be noted that all paths from any node in $N^h(v_i)$ (including v_i) to v_j in G go through the nodes in V''. Now if we can show that there exists at least one node $v_d \in V''$ such that $v_d \in V'$, we actually prove v_i is *D-path* connected to v_j via dominating node v_d.

This can be shown easily when v_j is a dominating node (i.e., $v_j \in V'$). As $h \geq 2$ and $\delta(v_i, v_j) = h+1$, there must exist a node $v_k \in N^{h-1}(v_i)$ such that $\delta(v_j, v_k) = 2$. As $v_j \in V'$ and $UN_{V'}(v_j) = \phi$ there exists a *D-path* from v_j to v_k. But all paths from v_k to v_j or vice versa pass through some nodes in V''. So, there exists a dominating node v_d on the *D-path* from v_j to v_k where v_d is in V''.

Let us assume that v_j is a non-dominating node $v_j \notin V'$. Also, assume that V'' does not contain any dominating node. Then, there must exist at least one dominating neighbour $v_d \in N^h(v_i)$ for each node in V'', as v_i is *D-path* connected to all nodes in $N^h(v_i)$. Definitely $v_d \notin V''$ as we assumed that V'' does not contain any dominating node. But, $\delta(v_k, v_j) = 2$, so $v_j \in N^2(v_k)$. As all paths from v_k to v_j go through some nodes in V'', it must have at least one dominating node on the path from v_k to v_j, otherwise $v_j \in UN_{V'}(v_k)$, contradicting our assumption that $UN_{V'}(v_k) = \phi, \forall v_k \in V'$. ■

It is to be noted that the property $UN_{V'}(v_i)$ of a dominating node is defined within the 2-hop neighbourhood of v_i which can be locally computed at each node knowing only neighbouring dominating nodes. Such local computation does not, however, restrict to verify the overall connectivity of the dominating set V'. The next theorem is a stronger version of the Theorem 4.1 that is used to prove that the set of nodes selected by our algorithm is a CDS.

Theorem 4.2 *Let V' be a dominating set of $G(V, E)$. Let V'' be any subset of V' such that V'' is also a dominating set of G. We compute $UN_{V'}(v_i), \forall v_i \in V''$. If $UN_{V'}(v_i) = \phi, \forall v_i \in V''$ then V' is a connected dominating set of G.*

Proof: Let $V''' = V' - V''$. We see, by Theorem 4.1, that each node $v_i \in V''$ is *D-path* connected to all nodes in V and may use some dominating node from the set V'''. However, V'' being a DS of G, each node $v_j \in V'''$ must have a neighbouring dominating node $v_k \in V''$. So, each node $v_j \in V'''$ is also *D-path* connected to all nodes in V via its dominating neighbour in V''. This proves the theorem. ■

We additionally prove the following interesting properties of dominating sets which may be of general use.

Property 4.1 *Let V' be a dominating set of a graph $G(V, E)$ with $|V'| > 2$. Let v_i and v_j be any two nodes of V' such that $\delta(v_i, v_j) > 3$. Then, there always exist two nodes v_k and v_l in V' such that $\delta(v_i, v_k) \leq 3$ and $\delta(v_j, v_l) \leq 3$, and v_k and v_l may be the same node.*

Proof: We prove this by induction. Given a dominating set V', let us assume $\delta(v_i, v_j) = 4$. Then, there exists a path, say, $v_i - v_a - v_b - v_c - v_j$. Assume, all these intermediate nodes on the path are non-dominating nodes, otherwise, the property is already satisfied. As V' is a dominating set of G, then the node v_b must have at least one neighbouring node $v_k \in V'$. So, $\delta(v_i, v_k) = \delta(v_j, v_k) = 3$. In this case, $v_l = v_k$. So, the property is satisfied for $\delta(v_i, v_j) = 4$. However, when $\delta(v_i, v_j) = 5$, v_k and v_l can be two distinct nodes.

Let us assume that this is true for $\delta(v_i, v_j) = h$ where $h > 4$. Let us consider the case where $\delta(v_i, v_j) = h + 1$. The h non-dominating nodes on the path (say in order v_{a_1}, v_{a_2}, $\cdots$, v_{a_h}) are in between v_i and v_j, with v_{a_1} adjacent to v_i and v_{a_h} adjacent to v_j. V' being a dominating set of G, node v_{a_2} must have at least one adjacent node v_{k_1} in V'. So, $\delta(v_i, v_{k_1}) = 3$ and $\delta(v_j, v_{k_1}) = h$, so the property is true by induction. ■

Property 4.2 *Given any dominating set V' of G, for a dominating node $v_i \in V'$ there must exist another node $v_j \in V'$ which is no more than 3-hop away from v_i.*

Proof: It follows directly from Property 4.1. ■

Theorem 4.3 *Let V' be a dominating set of a graph $G(V, E)$ with $|V'| > 1$. V' is a CDS if and only if every node $v_i \in V'$ is D-path connected to all those nodes of V' which are at most 3-hop away from v_i in G.*

Proof: [*If*] Without loss of generality, let us assume v_i and v_j be two nodes of V' such that $\delta(v_i, v_j) > 3$. It is to be proved that v_i and v_j are *D-path* connected. We are given that, for every node $v_i \in V'$, v_i is *D-path* connected to all those nodes of V' which are at most 3-hop away from v_i in G. Now, according to the Property 4.2 there must exist at least one node $v_{k_1} \in V'$ such that $\delta(v_i, v_{k_1}) \leq 3$. Let S_i^1 be the set of all such nodes in V'. But, then v_i is *D-path* connected to all nodes of S_i^1. Now, if for any node v_{k_1} of S_i^1, $\delta(v_{k_1}, v_j) \leq 3$ then v_{k_1} is *D-path* connected to v_j, and it implies that v_i is *D-path* connected to v_j. When all the nodes of S_i^1 are more than 3-hop away from v_j then from every node of S_i^1 we can again find some other set of nodes of V' such that they are at most 3-hop away from the members of S_i^1. This process is repeated until we reach a node v_{k_p} such that $\delta(v_{k_p}, v_j) \leq 3$. We must find such a node v_{k_p}, otherwise, v_j will be more than 3-hop away from all the nodes of V', which contradicts the Property 4.1.

[*Only if*] When V' is a CDS, all members of V' are *D-path* connected. It implies that every node $v_i \in V'$ is also *D-path* connected to all other nodes of V', including those at most 3-hop away in G. ■

4.4 CDS Construction Algorithm

We assume that each node v_i is already aware of its 2-hop partial graph $PG^2(v_i)$ for the underlying topology graph $G(V, E)$. The CDS construction is implemented in the following three phases executed in each node in distributed fashion:

- Phase 1: each node computes an *estimate* (ξ) (defined below) and broadcasts it to its neighbours. The node with highest estimate among its neighbours is recommended to be a dominating node. All such nodes constitute the initial dominating set (D_I).
- Phase 2: the connectedness among the nodes in D_I is checked (using Theorem 4.1).
- Phase 3: if D_I is not connected, some additional nodes are recommended to become dominating nodes by the nodes in D_I. Set D_I along with the additional nodes selected in this phase form the final dominating set D_F which is the desired CDS of the underlying topology graph G.

4.4.1 Initial Dominating Set Selection

During the first phase of CDS construction algorithm, each node checks its suitability for acting as a dominating node using the following estimate.

Definition 4.5 *The estimate of an edge between two nodes v_i and v_j for the underlying topology graph $G(V, E)$, denoted as $\xi(v_i, v_j)$, is the total number of nodes, adjacent to either v_i or v_j or both. That is, $\xi(v_i, v_j) = |N^1(v_i) \cup N^1(v_j)|$. The estimate of a node v_i is the maximum among the estimates of all the edges incident on it and denoted as $\xi(v_i)$, i.e., $\xi(v_i) = max\ \{\xi(v_i, v_j), \forall v_j \in N^1(v_i)\}$.*

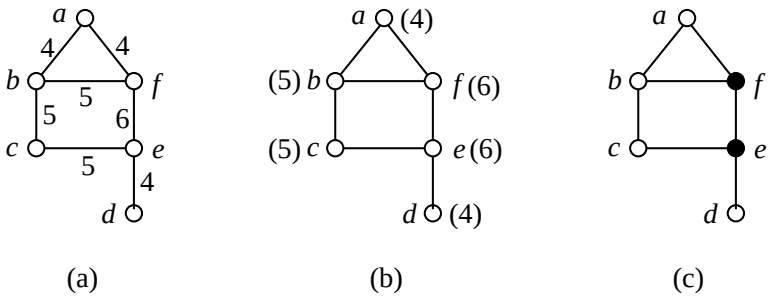


Figure 4.3: Selection of initial dominating nodes (a) edge estimates, (b) node estimates and (c) initial dominating set (D_I)

Figure 4.3 shows an example of node and edge estimates for the topology graph shown in Figure 4.1(a). Figure 4.3(c) shows the initial dominating set $D_I = \{e, f\}$ selected by this procedure for the topology graph shown in Figure 4.1(a).

Each node v_i first informs its estimate $\xi(v_i)$ to all its neighbours. After knowing the node estimates of the neighbours, each node v_i recommends exactly one of its neighbouring nodes, the one with the highest estimate (tie is broken by node ID), as a dominating node. Each recommended node marks itself as an initial dominating node forming the set D_I. The fact that a node is elected by its neighbours, instead of selecting itself as a dominating node, makes a better selection of the initial dominating set D_I. The details of the procedure *SelectingInitialDS* are given below.

Procedure SelectingInitialDS
begin
Initialization for each node v_i:
  For each edge (v_i, v_j)
    Compute edge estimate $\xi(v_i, v_j)$
  Compute node estimate $\xi(v_i)$

First round for each node v_i:
  Exchange $\xi(v_i)$ with nodes in $N^1(v_i)$

Second round for each node v_i:
  Let $\xi(v_j) > \xi(v_k), \forall v_k \in N^1(v_i)$
  Recommend v_j to be a dominating node

Finalizing for each node v_i:
  If v_i is recommended by some $v_j \in N^1(v_i)$
    Mark v_i as dominating node (i.e., $D_I = D_I \cup \{v_i\}$)
end

The following Lemma 4.2 and Theorem 4.4 prove that the set D_I constructed in this way is really a dominating set.

Lemma 4.2 *Let* $v_i \in D_I$ *and* $|D_I| > 1$, *then there exists a node* $v_j \in N^1(v_i)$ *such that* $v_j \in D_I$.

Proof: Each node v_i recommends exactly one node $v_j \in N^1(v_i)$ to become a dominating node irrespective of whether v_i belongs to D_I or not. So, for any node $v_i \in D_I$ there exists another node $v_j \in N^1(v_i)$ in D_I. ■

Theorem 4.4 *The selected set of nodes,* D_I, *is a dominating set of the topology graph* $G(V, E)$.

Proof: Each node v_i recommends exactly one of its neighbours as a dominating node and every recommended node is included in D_I. So, for every node $v_i \in V$ (whether $v_i \in D_I$ or $v_i \notin D_I$) there is at least one neighbouring node $v_j \in D_I$ (node v_j is recommended by v_i). Thus, according to the definition of dominating set, D_I is a DS of G. ■

Lemma 4.3 *The time and message complexities of initial dominating set selection procedure are* $O(1)$ *and* $O(n)$, *respectively.*

Proof: Each node sends exactly one *recommend* message in the first round and at most one *declared dominating* message in the second round. So, at most $2n$

messages are exchanged for selecting the initial dominating set proving the message complexity as $O(n)$. This also shows that exactly two rounds of message passing is required proving the time complexity as $O(1)$. ■

4.4.2 Connectedness Verification

Our simulation studies on large number of random graphs show that D_I is a CDS in most of the cases. However, the construction procedure does not ensure that D_I is either minimal or connected. For example in Figure 4.4(a), we see that $D_I = \{a, b\}$ is a CDS. However, only one of them would suffice, so it is not minimal. In Figure 4.4(b), we see that $D_I = \{a, b, d, f\}$ forming two disconnected components, $\{a, b\}$ and $\{d, f\}$.

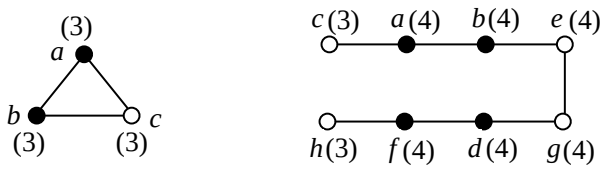


Figure 4.4: Initial dominating set D_I (a) not minimal and (b) not connected

The purpose of this phase is to verify the connectedness among the nodes in D_I by checking it locally at each dominating node or at its neighbouring nodes. Some definitions are given below that help in the verification procedure.

Definition 4.6 *Let V' be a DS of $G(V, E)$ and v_i be any non-dominating node (i.e., $v_i \notin V'$). The set of 1-hop dominating neighbours of v_i is $DN_N^1(v_i) = \{S_1^1, S_2^1, \cdots, S_{k_1}^1\}$, where S_1^1, S_2^1, $\cdots$, $S_{k_1}^1$ are disjoint sets of dominating nodes from $N^1(v_i) \cap V'$ such that all nodes in each S_i are D-path connected in $PG^2(v_i)$. The 2-hop dominating neighbours of v_i is $DN_N^2(v_i) = \{S_1^2, S_2^2, \cdots, S_{k_2}^2\}$, where S_1^2, S_2^2, $\cdots$, $S_{k_2}^2$ are disjoint sets computed in the following way. We first extend each set S_i^1 to S_i' where,*

$$S_i' = S_i^1 \cup \{ \bigcup_{\forall v_j \in S_i^1} DN_D^1(v_j) \}.$$

If an extended set S_i' overlaps with another extended set S_j', we compute $S_i' = S_i' \cup S_j'$ and remove S_j'. All these remaining S_i' form S_i^2.

At the start of this phase, each dominating node v_i computes $DN_D^2(v_i)$ and $UN_{V'}(v_i)$ and each non-dominating node v_j, (i.e., $v_j \notin D_I$), computes $DN_N^2(v_i)$. Each dominating node v_i, for which $UN_{V'}(v_i) = \phi$, remains inactive during this phase. However, a dominating node v_i, for which $UN_{V'}(v_i) \neq \phi$, requests all nodes in $N^1(v_i)$ to verify its *D-path* connectivity with its $UN_{V'}(v_i)$. Node v_i sends a message containing $DN_D^2(v_i)$ and $UN_{V'}(v_i)$ to nodes in $N^1(v_i)$. Each node, on receiving this message from v_j, computes the extended connected set of dominating nodes (denoted as D_E) from its own 2-hop dominating neighbours and v_j's 2-hop dominating neighbours. It then verifies if at least one node from the dominating neighbour set of each node in $UN_{V'}(v_j)$ lies in D_E. By this procedure, some or all of the nodes in $UN_{V'}(v_j)$ may be found to be already connected. The details of the procedure *VerifyConnectivity* are given below.

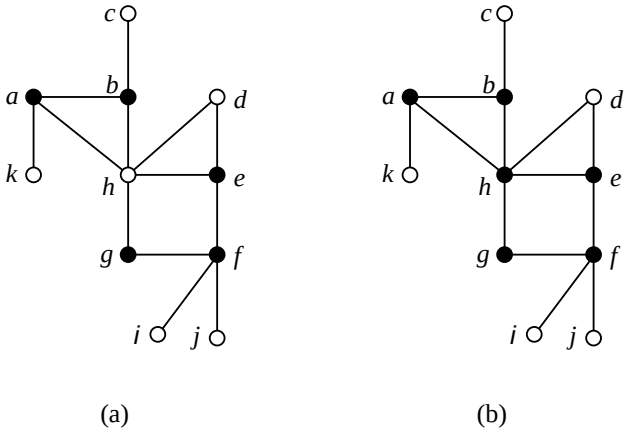


Figure 4.5: Selecting dominating nodes (a) initial dominating set D_I and (b) final connected dominating set D_F

Let us consider the initial dominating nodes reported in Figure 4.5(a). We compute $DN_N^1(h) = \{\{a, b\}, \{e\}, \{g\}\}$. Note that the nodes e and g are not directly connected, so they have been placed in two different sets in $DN_N^1(h)$. However, when we go up to 2-hop neighbours and compute DN_N^2, we first extend each set of $DN_N^1(h)$ forming sets of dominating nodes $\{a, b\}$, $\{e, f\}$ and $\{f, g\}$. Sets $\{e, f\}$ and $\{f, g\}$ are then merged (f being common) to form the set $\{e, f, g\}$. Thus, $DN_N^2 = \{\{a, b\}, \{e, f, g\}\}$.

Procedure VerifyConnectivity
begin
Initialization for each node v_i:
  If $v_i \in D_I$
    Compute $DN_D^2(v_i)$
    Compute $RN_D(v_i) = \{v_j | v_j \in N^1(v_i)$ or $(v_k \in DN_D^1(v_i)$ and $v_j \in N^1(v_k))\}$.
    Compute $UN_{V'}(v_i) = N^2(v_i) - RN_D(v_i)$
  If $v_i \notin D_I$
    Compute $DN_N^2(v_i)$
  $D_E = R_E = \phi$

First round for each node v_i:
  If $v_i \in D_I$ and $UN_{V'}(v_i) \notin \phi$
    Inform $DN_D(v_i)$ and $UN_{V'}(v_i)$ to nodes in $N^1(v_i)$

Second round for each node v_i:
  If v_i receives $DN_D(v_j)$ and $UN_{V'}(v_j)$ from $v_j \in N^1(v_i)$
    If $v_i \in D_I$
      $D_E = DN_D(v_i) \cup DN_D(v_j)$
    If $v_i \notin D_I$
      foreach $S_i^2 \in DN_N^2(v_i)$
        if $S_i^2 \cap DN_D(v_j) \neq \phi$
          $D_E = D_E \cup S_i^2$
      if $D_E \neq \phi$
        $D_E = D_E \cup DN_D(v_j)$
    foreach $v_k \in UN_{V'}(v_j)$ and $v_k \in N^1(v_i)$
      if $DN_D^1(v_k) \cap D_E \neq \phi$ or $DN_N^1(v_k) \cap D_E \neq \phi$
        Add v_k into $R_E(v_j)$
    Send $R_E(v_j)$ to v_j

Finalizing for each node v_i:
  If v_i receives R_E from $v_j \in N^1(v_i)$
    Delete all nodes in R_E from $UN_{V'}(v_i)$
end

Hence this phase just checks the connectivity among the nodes in D_I, either locally at each dominating node or by computation on its neighbouring nodes. If it finds that $UN_{V'}(v_i) = \phi$ for all $v_i \in D_I$, it concludes that the initial dominating set D_I is a CDS. Otherwise, it goes through the next phase, presented below.

4.4.3 CDS Finalization

However, after the verification phase, it may be the case that for some node $v_i \in D_I$ the $UN_{V'}(v_i) \neq \phi$. In this phase, each such node selects some additional nodes from $N^1(v_i)$ as dominating nodes resulting in the final dominating set D_F which forms a CDS of the topology graph $G(V, E)$. This problem of finding additional nodes at each node $v_i \in D_I$, is mapped to the classical set cover problem and solved at

each node using greedy heuristic [68]. The universal set of the set cover problem is $UN_{V'}(v_i)$. The family of subsets $\mathcal{S} = \{S_1, S_2, \cdots, S_k\}$ is constructed as follows. For each $v_j \in N^1(v_i)$ and $v_j \notin D_I$ we construct a set $S_j = N^1(v_j) \cap UN_{V'}(v_i)$ and place S_j in $\mathcal{S}$ if $S_j \neq \phi$. The details of the procedure *ConnectingDS* are given below.

Procedure ConnectingDS
begin
Set $D_F = D_I$
Initialization for each node v_i:
  If $v_i \in D_I$ and $UN_{V'}(v_i) \neq \phi$
    foreach $v_j \in N^1(v_i)$ and $v_j \notin D_I$
      If $(S_j = N^1(v_j) \cap UN_{V'}(v_i)) \neq \phi$
        $\mathcal{S} = \mathcal{S} \cup S_j$
    Let C be the solution of the set cover problem
    $A_D = \{v_j | S_j \in C\}$

First round for each node v_i:
  If $v_i \in D_I$ and $UN_{V'}(v_i) \notin \phi$
    Send A_D to all nodes in $N^1(v_i)$

Finalizing for each node v_i:
  If A_D is received and $v_i \in A_D$
    Mark v_i as dominating node (i.e., $D_F = D_F \cup \{v_i\}$)
end

Figure 4.5(b) shows the final dominating set $D_F = \{a, b, e, f, g, h\}$ generated by this procedure for the initial dominating set shown in Figure 4.5(a). The dominating node h is included in the final dominating set D_F which was missing in the D_I as shown in Figure 4.5(a).

Theorem 4.5 *The selected set of nodes, D_F, is a connected dominating set of the topology graph $G(V, E)$.*

Proof: It is evident that $D_F \supseteq D_I$ as D_F contains all nodes of D_I and may have some additional nodes selected during the *Connecting DS* phase of the algorithm. As D_I is a dominating set of G (proved in Lemma 4.4), so D_F is also a dominating set of G. The selection of additional nodes (if required) in *connecting* phase ensures that $UN_{V'}(v_i) = \phi$ for all $v_i \in D_I$. Setting $V' = D_F$ and $V'' = D_I$ in Theorem 4.2 proves that D_F is a CDS. ■

Lemma 4.4 *The time and message complexities of the Verify Connectivity and the Connecting DS procedures together are $O(1)$ and $O(n)$, respectively.*

Proof: The proof comes trivially from the steps of the algorithms. The *Verify Connectivity* phase finishes in two rounds of message passing and in each round at most one message is sent by any node resulting in at most $2n$ messages. The *Connecting* phase finishes in one round of message passing and at most one message is sent by any node in D_I resulting in at most n messages. ■

Theorem 4.6 *The CDS construction runs in $O(1)$ time complexity, and $O(n)$ message complexity.*

Proof: The proof directly follows from Lemmata 4.3 and 4.4. ■

4.5 Simulation Studies

For performance comparison, both the Wu and Li's algorithm [173] and our algorithm are executed on the same set of graphs. In each case, the size of the CDS reported by the algorithms are noted, and the average is used to compare the results. Figure 4.6 shows the variation of the dominating set size with the maximum node

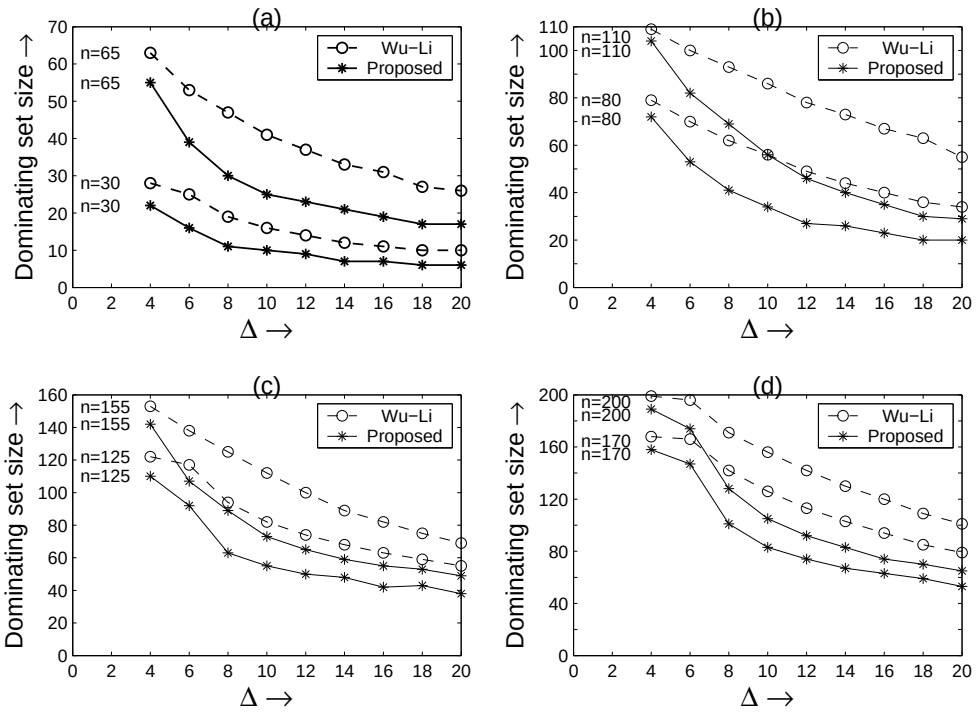


Figure 4.6: Dominating set size versus max node degree(Δ)

degree Δ, keeping network size (n) fixed. It is obvious from Figure 4.6 that our algorithm outperforms Wu and Li's algorithm [173]. In particular, our algorithm generates considerably smaller CDS when the network is sparse.

Also, the variation in the dominating set size versus the number of nodes (n) keeping the maximum node degree fixed is shown in Figure 4.7. We observed that our algorithm generates less number of dominating nodes for all Δ values. Also, the difference in the dominating set size continuously increases as n grows. In an environment with large number of hosts (e.g., conferences, meetings, etc.) in a small confined region, our method chooses a significantly smaller set of dominating nodes for routing.

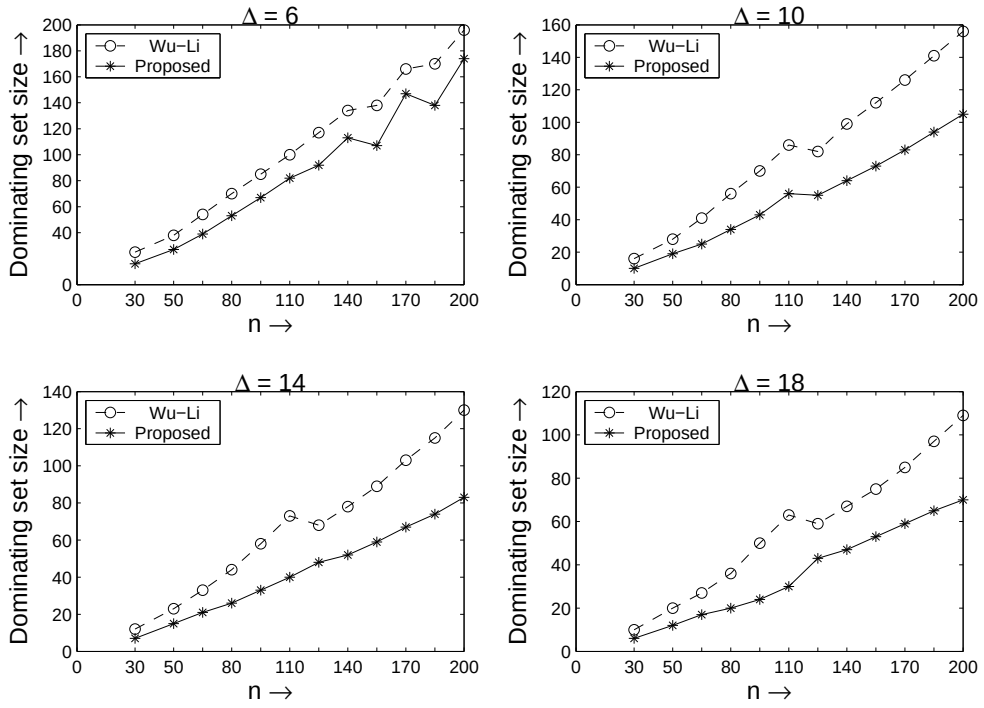


Figure 4.7: Dominating set size versus number of nodes(n)

Summary

We develop an elegant coordinated distributed algorithm based on the unique graph theoretic properties of dominating set to find the connected dominating set in ad hoc networks. It can be effectively used as a routing backbone for the network.

The proposed strategy of selecting a dominating node is not based on the node's own local computation only, but based on a coordinated decision of neighbours. This radically distinct approach with $O(1)$ time complexity and $O(m)$ messege complesity, produces connected dominating sets that are in general smaller than the sets produced by the algorithms introduced earlier. Based on the simulation study, we observe that the size of the dominating sets obtained by our proposed algorithm are always less than that of the algorithm in [173]. It generates considerably smaller CDS especially for sparse networks where the number of cliques are few.

Chapter 5

Energy Efficient Bluetooth Scatternet Formation

In a Bluetooth network, the fundamental backbone structure for a group of devices sharing a common frequency hopping sequence is called a *piconet*. Each piconet comprises a *master* and a few *slaves* of which only a predetermined constant number of slaves (κ) can simultaneously remain active, the rest remains idle termed as *parked* nodes. A Bluetooth network can be extended by interconnecting the piconets, using bridge nodes, forming a *scatternet*. The *scatternet formation problem*, is to partition the nodes into several piconets and to select a few nodes as *bridges* to realize inter-piconet communications. The primary objective is to make sure that the resulting scatternet remains connected satisfying all constraints posed by the Bluetooth specifications. Obviously the final goal of scatternet design is to offer faster intra-piconet and inter-piconet message exchanges.

The *Bridge* nodes are time-shared between multiple piconets, receiving data from one piconet and forwarding it to another. A *bridge* can be a master in one piconet and a slave in another (termed as M/S bridge) or slaves in both piconets (termed as S/S bridge) as shown in Figure 5.1. In an M/S bridge, two nodes, normally operating as slaves in their respective piconets, form a separate temporary piconet from time to time to relay messages between two piconets. Obviously, switching between roles and to synchronize with different piconets require more time and complicated procedures compared to that of S/S bridges. An S/S bridge is a conventional bridge that is time-shared between the two piconets easily, without affecting throughput. Therefore, in scatternet formation, M/S bridges are generally avoided unless they become essential for connectivity. Moreover, a bridge should not participate in more than two or three piconets, as this may reduce the throughput significantly. Since,

a bridge has to switch between the hopping sequences of the piconets it shares, it is desirable to have fewer bridges for faster communication.

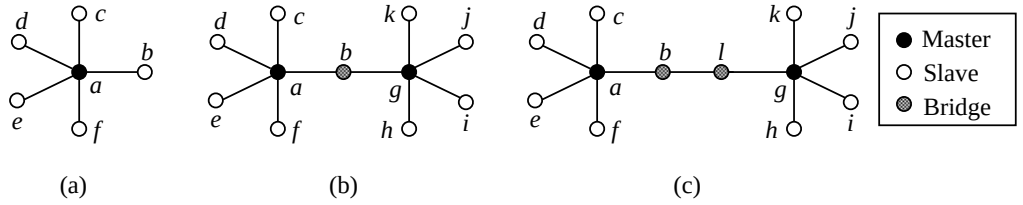


Figure 5.1: Bluetooth configurations: (a) a piconet, (b) a scatternet with the node b as S/S bridge and (c) a scatternet with the nodes b and l as M/S bridges

It is equally important that the scatternet should attempt to connect all devices by using minimum number of piconets, to reduce the probability of frequency overlapping between piconets. It has been shown in [110] that the burst failure rate increases with the number of piconets. It is also necessary to limit the diameter of the scatternet, so that nodes may communicate with less delay.

It is difficult to decide the exact metrics that can evaluate the performance of a scatternet solution. However, minimizing the number of piconets, bridges and parked nodes, and avoiding M/S bridges, are usually considered good indicators as mentioned in [95, 110]. A set of Bluetooth devices can be organized into a scatternet in numerous configurations, but it is evident that finding the optimal one that optimizes the multiple objectives mentioned here is of exponential complexity.

Chapter Organization

The rest of this chapter is organized as follows. Section 5.1 reviews the related works on Bluetooth scatternet formation, especially the distributed ones. In Section 5.2, we discuss the salient features of our study on scatternet formation protocol. Our energy efficient scatternet formation protocol is presented in Section 5.4. Five phases of this protocol and their complexity analysis and completeness proof are discussed in the five subsections of Section 5.4. Simulation results are discussed in Section 5.5. The chapter concludes with a summary.

5.1 Background

As the number of Bluetooth devices to be inter-connected increases day by day, researchers put extensive effort to develop a good methodology for that. In [116], some heuristic rules are derived based on experiments to build power-optimal Bluetooth scatternet. In [28], the power characteristics of the Bluetooth technology are investigated in broader sense, and accurate power consumption measurements for different Bluetooth operating modes are presented. Finally, it examines the trade-off between power consumption and performance for a commercial off-the-shelf Bluetooth device.

Some efficient scatternet formation protocols are presented in [94, 95, 96] for single-hop topologies. For multi-hop topologies, the scatternet formation protocol described in [180] requires that the protocol be initiated by a designated node *blueroot* and generates a tree-like scatternet. This solution is not truly distributed as some external agency must decide the *blueroot*. Also, the tree structure is vulnerable to faults. Some of the solutions that produce topologies different from a tree are those presented in [99, 16, 159, 165]. The protocols proposed in [99, 159] require that each Bluetooth device be accompanied with GPS (Global Positioning System) capability. The scatternet formation scheme BlueNet proposed in [165], produces a scatternet whose piconets have a bounded number of slaves, but the connectivity of the resulting scatternet is not guaranteed. In the scatternet formation scheme, *Configuring BlueStars*, collision arises when two or more would-be masters try to communicate with a common neighbour. Moreover, it does not attempt to solve the *parking* problem of scatternet. Algorithms presented in [1, 140] depend on a single device to design the scatternet topology and notify other devices. The distributed tree scatternet formation protocol proposed in [160] results relatively short scatternet formation latency, but it does not minimize the number of piconets.

The time complexities of all these algorithms are $O(\lambda)$ where λ is the diameter of the topology graph. So, in the worst case, the scatternet formation time grows linearly with the diameter of the topology. Moreover, none of the solutions attempts to minimize the number of *parked* nodes in the scatternet. Collision often arises when two or more would-be masters try to communicate with a common neighbour during scatternet formation. None of the algorithms has addressed this aspect as well.

5.2 Our Contributions

We present a distributed Energy-Efficient Scatternet Formation Protocol (EESFP) that produces a mesh of piconets, instead of a tree of scatternets, offering more robustness. We implement a novel synchronization mechanism during the initial phase of our protocol that ensures collision-free exchange of messages in the following phases. This technique assigns dedicated time slots to each link (i.e., specific node pairs) for communication. It allows each node to predict the exact time when it may need to participate in communication. Thus, it enables nodes to go into *sleep* mode during the idle time to save energy significantly. Secondly, we apply a distributed graph-theoretic degree reduction technique to reduce the degree of nodes to κ. This attempts to minimize the number of parked slaves in the generated scatternet making the solution more efficient. Simulation results show that in most of cases it can limit the node degree of a master to the desirable value κ thus forming the scatternet with minimum number of parked nodes.

5.3 Representation of Scatternet

Let us assume that $V = \{v_1, v_2, \cdots, v_n\}$ be a set of n identical Bluetooth devices forming a connected multi-hop topology in a Bluetooth ad hoc network.

Definition 5.1 *A scatternet, for a topology graph $G(V, E)$, can be represented as a connected graph $SG(V, R, E')$, where each device $v_i \in V$ is assigned a role $r_i \in R$, where $R = \{master, bridge, slave\}$, and $E' \subseteq E$ such that $(v_i, v_j) \in E'$ if $(v_i, v_j) \in E$ and if $r_i = master$ and $r_j \in \{slave, bridge\}$, or vice-versa (i.e., $r_j = master$ and $r_i \in \{slave, bridge\}$).*

Example 5.1 *Figure 5.2(a) shows a topology graph with* 15 *nodes. Assuming $\kappa = 4$, Figure 5.2(b) shows a scatternet graph SG satisfying all the constraints. Note that the degree of each node in SG is $\leq \kappa$.*

5.4 Energy-Efficient Scatternet Formation Protocol

Given a collection of Bluetooth devices, to design an efficient scatternet to offer quality communication backbone, the proposed scheme EESFP (energy efficient

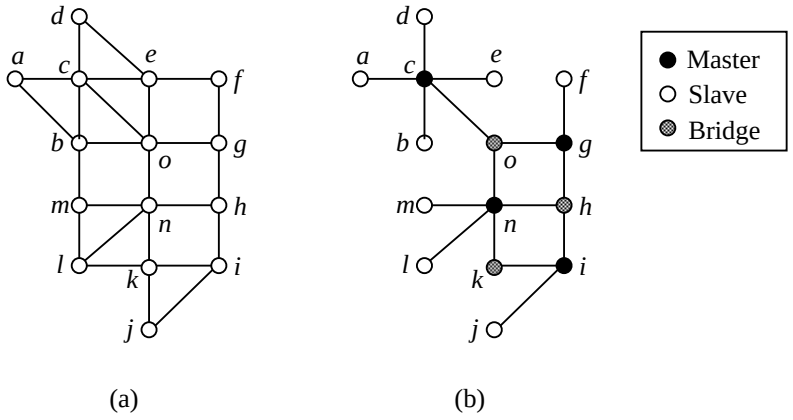


Figure 5.2: (a) A topology graph G and (b) corresponding scatternet graph SG

scatternet formation protocol) defines a sequence of five phases to be executed on each individual node in a distributed fashion. According to the basic Bluetooth specification, for any message exchange the transmitter and the receiver has to remain in the *busy waiting state* for synchronization. It consumes significant amount of energy as well as time. To avoid it, over and above the partial synchronization at the boundaries of the phases, this protocol imposes a slot-based synchronization, compatible with the original Bluetooth specification, during neighbourhood discovery phase. It makes the protocol highly energy-efficient. The consecutive phases are given below:

- Neighbourhood Discovery
- Adjacency List Exchange
- Degree Reduction
- Role Estimation
- Role Assignment

The following subsections describe the phases in detail.

5.4.1 Neighbourhood Discovery

Each Bluetooth device first discovers its neighbours for a predefined time T, referred here as the *discovery time*, as has been done in [140]. In EESFP, a synchronization technique is implemented within this phase. The communicating node pairs exchange their ID's and some additional synchronization information in the form of

sync bit vector. It helps to ensure collision-free message exchange in the following phases.

Each device v_i maintains a *sync bit vector* $S_i^B(k)$, and a *sync node vector* $S_i^N(k)$, where $1 \leq k \leq 2(\Delta-1)$, Δ is the maximum number of devices that can be discovered by any node within time T, as explained later in Lemma 5.1. Initially, for any device v_i, $S_i^B(k) = S_i^N(k) = 0$, $1 \leq k \leq 2(\Delta - 1)$. Whenever v_i discovers an adjacent device v_j, S_i^B and S_j^B are exchanged. Each device then finds the *minimum* index f, where $S_i^B(f) = S_j^B(f) = 0$. Both v_i and v_j update their *sync bit vectors* as $S_i^B(f) = S_j^B(f) = 1$, respectively, and set their own *sync node vector* $S_i^N(f) = j$, and $S_j^N(f) = i$, respectively.

Once the discovery time T is over, whenever nodes v_i and v_j require to exchange messages, they will form a temporary piconet in synchronized ways, during the f-th slot. Checking the entry $S_i^N(f)$ ($S_j^N(f)$), the node with higher ID will decide to be the master, ensuring that when any node v_i wants to communicate with an adjacent node v_j, if v_i is ready for transmission, it is guaranteed that v_j is ready for listening. No other neighbour of the two will interfere during this slot. For any node v_i, during a time slot x, if $S_i^B(x) = 1$, v_i should remain active in transmission or receiving. Whereas, if $S_i^B(x) = 0$, v_i may remain in the sleep mode. This leads to the development of an energy-efficient collision-free protocol for message transfer among the devices, enabling the nodes to conserve significant amount of energy.

Example 5.2 *Let us consider the nodes b and o of the topology graph shown in Figure 5.3. At any instant of the neighbourhood discovery phase, let us assume that nodes b and o have already discovered their neighbours {c, a, m} and {c, g, n} respectively. Now the nodes b and o are in the process of discovering each other.*

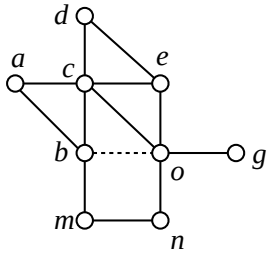


Figure 5.3: Topology graph before discovery of edge (b, o)

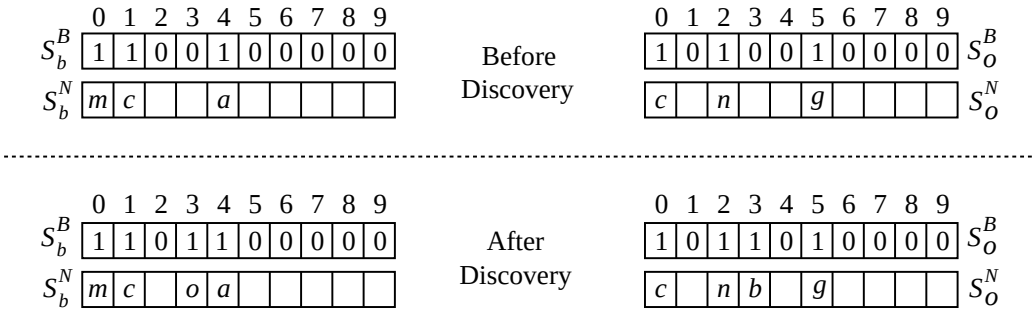


Figure 5.4: The *sync vectors* at nodes b and o during discovery

Figure 5.4 shows a possible snapshot of S^B and S^N vectors in the two nodes before and after the discovery.

Here follows the algorithm for the neighbourhood discovery phase that is to be executed in each node v_i.

Procedure NeighbourDiscovery
begin
Initialization for each node v_i:
$N(i) = \phi$.
$S_i^B[1 \cdots (2\Delta - 1)] = 0$ and $S_i^N[1 \cdots (2\Delta - 1)] = 0$.
T_{disc} = predetermined time interval.

Computations at each node v_i until $T_{disc} = 0$:
If $rand(0, 1) < 0.5$
  INQUIRY(). // BT protocol
Else
  INQUIRYSCAN(). // BT protocol
If v_i sends/receives packet to/from a new node v_j
  If v_i is in INQUIRY mode
    $Master = v_i$ and $Slave = v_j$.
  Else
    $Master = v_j$ and $Slave = v_i$.
  $N(i) = N(i) \cup \{v_j\}$.
  ESTABLISHLINK($Master$, $Slave$). // BT protocol
  Exchange IDs, and *sync bit vectors* S_i^B, S_j^B.
  For $f = 0$ to $(2\Delta - 1)$
    If $S_i^B(f) = S_j^B(f) = 0$
    $S_i^B(f) = 1$, $S_i^N(f) = j$ and break.
end

Lemma 5.1 *For a given discovery time T, if Δ be the maximum number of nodes that can be discovered by any node, $(2\Delta - 1)$ slots are sufficient to ensure synchronized communication between any pair of adjacent nodes in the process of scatternet formation.*

Proof: Let v_i and v_j be two Bluetooth nodes such that $dist(v_i, v_j) \leq R_{max}$, the maximum range of the devices and v_i has just discovered node v_j or vice versa. If it is assumed that v_i (also v_j) have already discovered $(\Delta - 1)$ other neighbouring devices, then, $(\Delta - 1)$ entries are already filled in S_i^B and S_j^B, respectively. In the worst case, all these $2(\Delta - 1)$ entries may be disjoint. Therefore, to find a common free slot, v_i and v_j need at least one more place in their respective S_B's. It proves that $(2\Delta - 1)$ slots are sufficient to assign slots to every individual neighbour of a node for synchronization. ■

In the following phases, it will be assumed that time is divided into cycles, each with $(2\Delta - 1)$ slots, where each pair of adjacent nodes is assigned a given slot of the cycle for communication. Whenever required, a pair of adjacent nodes v_i and v_j exchange messages msg_i and msg_j by forming a temporary piconet, as described in *BTInfoExchange* procedure, in the assigned time slot. Successful completion of *BTInfoExchange* procedure results in delivering msg_i and msg_j to v_j and v_i respectively. Here, the ESTABLISHLINK is the procedure to establish a communication session between a master and a slave following the Bluetooth specifications. Procedure Info-Exchange described below runs on every Bluetooth device v_i in every time slot until the scatternet formation is over.

Procedure BTInfoExchange(*slotid*, msg_i, msg_j)
begin
  $v_j = S_i^N(slotid)$.
  If $v_j = 0$ sleep.
  Elseif $v_i > v_j$ // v_i the initiator
    $Master = v_i$, $Slave = v_j$.
  Else // v_j the initiator
    $Master = v_j$, $Slave = v_i$.
  ESTABLISHLINK($Master$, $Slave$).
  Send msg_i to v_j
  Receive msg_j from v_j
end

5.4.2 Adjacency List Exchange

Once the *Neighbourhood discovery* phase is over, each node v_i acquires a list of its neighbours within its range. Next, every pair of adjacent devices v_i and v_j exchange their list of neighbours, by forming a temporary piconet in the synchronized time slot. Procedure *Info-Exchange* described below runs on every Bluetooth device v_i. It is evident that at the end of this phase, every node v_i knows its neighbours within 2-hop distance.

Procedure AdjacencyListExchange
begin
First round for each node v_i
  For $k = 1$ to $(2\Delta - 1)$
    BTInfoExchange(k, $N^1(v_i)$, msg_j)
end

5.4.3 Degree Reduction

If a node discovers d_i adjacent neighbours, and $d_i > \kappa$, it is referred to as a *congested* node. We apply *degree reduction* technique on the congested nodes to release the excess adjacent nodes. This helps to minimize the number of *park* nodes maintaining connectivity of the graph. The degree reduction procedure starts by finding 3-cycles and 4-cycles around the congested node. In a cycle, the device with minimum ID is considered to be the *leader*. A congested node v_i attempts to find at most $(d_i - \kappa)$ such cycles where, either v_i is the *leader* or its adjacent node v_j is the *leader*. For each such cycle, it attempts to delete a link to one of its neighbours maintaining its connectivity through other neighbours.

For a consensus on deleting links, the *leader* tries to eliminate just a single edge from every cycle. For those cycles, where v_i is the leader, it deletes an edge (v_i,v_j) of the cycle where v_j has the maximum degree among its neighbours in the cycle under consideration. Node v_i also informs v_j about such deletion by forming a temporary piconet. For those cycles, where v_j, a neighbour of v_i, is the leader, v_i requests v_j to delete the edge (v_i, v_j) and v_j permits it if and only if v_j is not loosing connectivity with v_i by that. The procedure is repeated for each cycle.

At the start of this phase, each congested node starts finding cycles, and rest of the nodes may remain in *sleep* mode. Let t_{dr} be the total time required to complete

this procedure. At the end of this period, rest of the nodes will also wake up and exchange information to update their neighbourhood lists before they start the next phase. Here follows the algorithm for degree reduction phase to be executed in each congested node.

Procedure DegreeReduction
begin
Computation at each congested node v_i *(i.e.,* $d_i \geq \kappa$*)*
  // Identify 3-cycles and 4-cycles
  $C = \phi$ // the set of cycles
  For two nodes v_{i1} and v_{i2} in $N^1(v_i)$
    If $v_i \in N^1(v_{i2})$
      $C = C \cup \{v_i, v_{i1}, v_{i2}\}$.
    Elseif $v_{i3} \in (N^1(v_{i1}) \cap N^1(v_{i2}))$ and $\exists k$, such that
        $ID(v_{i3}) > ID(v_k)$, $k = i, i1, i2$,
      $C = C \cup \{v_i, v_{i1}, v_{i2}, v_{i3}\}$.
  // Put deletion marker in each cycle
  $D[1 \cdots (2\Delta - 1)] = 0$
  For each cycle $C_j \in C$
    If v_i is the minimum of all nodes in C_j
      If $d_{i1} \geq d_{i2}$
        Find f such that $S_i^N(f) = v_{i_1}$.
        $D_i(f) = 1$. // mark for deletion
      Else
        Find f such that $S_i^N(f) = v_{i_2}$ if C_j is a 3-cycle
        or, find f such that $S_i^N(f) = v_{i_3}$ if C_j is a 4-cycle.
        $D_i(f) = 1$.
    Else Let v_{i_1} the *leader*
      Find f such that $S_i^N(f) = v_{i_1}$.
      $D_i(f) = 1$.

First round
  For $k = 1$ to $(2\Delta - 1)$
    If $D(k) = 1$
      msg_i = *to delete edge* (v_i, $S_i^N(k)$)
      BTInfoExchange(k, msg_i, msg_j)
    Else
      BTInfoExchange(k, NULL, msg_j)
      If $v_j \in msg_j$ wants to delete link (v_j, v_i)
        It is allowed if link (v_i, v_{j2}) exists.

Second round
  AdjacencyListExchange
end

Example 5.3 *Figure 5.5(a) shows a topology having two congested nodes d and g, assuming* $\kappa = 4$*. In degree reduction phase, as explained above, node d finds out a*

4-cycle (d-e-g-i-d), where node d is the leader (with minimum ID), breaks this cycle by deleting edge (d,e). Node g detects the 3-cycle (g-f-h-g), where its neighbour node f is the leader, which deletes the edge (f, g). Finally, in the resulting graph all nodes are of degree $\leq \kappa$.

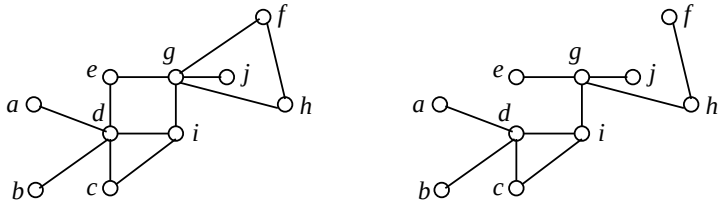


Figure 5.5: Degree Reduction in a topology with $\kappa = 4$: (a) the original topology and (b) the reduced topology

5.4.4 Role Estimation

Once the degree reduction phase is over, every node v_i attempts to estimate its role from the set $R = \{master, bridge, slave\}$. Firstly, based on the neighbourhood information, each node v_i estimates the largest scatternet that could be formed if it acts as a *bridge* and a pair of its neighbours, say v_{i1} and v_{i2}, act as *master*s. Also, the size of the largest cover corresponding to these roles is stored in v_i. It indicates the confidence level w_i with which the node v_i can assume the role of a *bridge*. Later these confidence levels are exchanged among the neighbours.

Based on the received role estimation information RI from each neighbour v_j, every node v_i checks if its own confidence level $w_i \geq w_j$. If it is greater, v_i selects itself as a *candidate bridge* node. Otherwise, one of its neighbours v_j with the highest w_j (i.e., $w_j > w_k, \forall v_k \in N^1(v_i)$) is selected as a *candidate bridge* node. Let B' be the set of nodes selected as *candidate bridge* nodes.

Theorem 5.1 *The set of nodes B' of candidate bridges forms a dominating set of the topology $G(V, E)$.*

Proof: The proof directly follows from Theorem 4.4. ■

Procedure RoleEstimation
begin
Computation at each node v_i
For $j = 1$ to $(d_i - 1)$
  For $k = j + 1$ to $(d_i - 1)$
    $w^{jk} = |N(i_j) \cup N(i_k)|$
Let $w_i = maximum\ w^{jk}$
Let (v_{i1}, v_{i2}) are the corresponding masters for w_i
$RI = (w_i, v_{i1}, v_{i2})$.

First round at each node v_i
  For $k = 1$ to $(2\Delta - 1)$
  Info-Exchange(k, RI, msg_j)

Second round at each node v_i
  Let $w_j = maximum$ among all the received w_l
  For $k = 1$ to $(2\Delta - 1)$
    If $S_i^N(k) = l$
      msg_i =recommend v_j as candidate bridge.
    Else
      $msg_i = NULL$
    Info-Exchange(k, msg_i, msg_j)
    else
      Info-Exchange(k, NULL, msg_j)

Third round for each node v_i
  If v_i is a recommended *candidate bridge*
    Recommend v_{i1} and v_{i2} in RI as *candidate masters*
end

Example 5.4 *Table 5.1 shows the computed value of* $RI = (w_i, M_{i1}, M_{i2})$ *at each node for the topology of Figure 5.2(a). From this table it is evident that when node a becomes a bridge and its two neighbours b and c act as masters, at most four other nodes (namely, d, e, m and o) will be covered by them. Similarly, node b finds that when it becomes a bridge and two of its neighbours c and m act as masters, six other nodes are covered by them. This is the largest subset that can be covered by node b with two adjacent masters.*

5.4.5 Role Assignment

The role assignment procedure starts with a set of *candidate bridges* B' selected in the previous phase. From Theorem 5.1, with B' as a dominating set, each node $v_i \in B'$ first computes the set of unreachable neighbours $UN_{B'}(v_i)$ as defined in Section 4.3. Each node $v_i \in B'$ generates a set cover problem similar

Node	w_i	M_{i1}	M_{i2}	Candidate bridge
a	4	b	c	
b	6	c	m	Yes
c	5	b	e	Yes
d	4	c	e	
e	4	c	f	
f	4	e	g	
g	5	h	o	Yes
h	5	g	n	Yes
i	4	h	k	
j	3	i	k	
k	5	i	n	Yes
l	5	k	n	
m	6	b	n	
n	7	k	o	Yes
o	8	c	n	Yes

Table 5.1: Role estimation for the topology of Figure 5.2(a)

to that in Section 4.4.3 with the universal set $UN_{B'}(v_i)$. The family of subsets $\mathcal{S} = \{S_1, S_2, \cdots, S_k\}$ is constructed as follows. For each $v_j \in N^1(v_i)$ and $v_j \notin B'$ we construct a set $S_j = N^1(v_j) \cap UN_{B'}(v_i)$ and place S_j in $\mathcal{S}$ if $S_j \neq \phi$. The set cover problem is solved at each node $v_i \in B'$ using greedy heuristic [68]. Let $\mathcal{C}_{\mathcal{SC}}$ be the solution of the set cover problem at node v_i and let $A_i = \{v_j | S_j \in \mathcal{C}_{\mathcal{SC}}\}$. If v_i has already assumed the role of a candidate bridge (in the previous phase) then the nodes in A_i will assume the roles of candidate masters and vice-versa. Let M' be the set of all the nodes selected as candidate masters and $\mathcal{S}_{\mathcal{MB}} = B' \cup M'$.

In the first step, each candidate master assumes the role of master and instructs all its neighbouring candidate bridges to assume the role of bridge. The master also selects remaining neighbouring nodes as slaves.

In the second step, if the roles of two neighbouring nodes v_i and v_j (where, $ID(v_i) < ID(v_j)$) are bridges (or masters) and there is no common master (or bridge), then v_i finds a node v_k in $N^1(v_i) \cap N^1(v_j)$ whose role is still unassigned. If such a node v_k exists then v_i informs v_k to become a master (or bridge) if v_i is a bridge (or master). Otherwise, v_i and v_j form a M/S bridge pair.

In the last step, a node whose role is still unassigned decides to act as a master. It must have at least one neighbour as a bridge node (otherwise, it would have already been assigned a role) and it gets connected to the neighbouring bridges.

Here follows the algorithm for role assignment phase.

Procedure RoleAssignment
begin
Computation at each candidate bridge v_i
If $UN_D(v_i) \neq \phi$
foreach $v_j \in N^1(v_i)$ and $v_j \notin D_I$
If $(S_j = N^1(v_j) \cap UN_D(v_i)) \neq \phi$
$\mathcal{S} = \mathcal{S} \cup S_j$
Let C be the solution of the set cover problem
$A_i = \{v_j | S_j \in C\}$
Let $M = \bigcup_{v_j \in B'}(A_j)$

Step 1.
If $v_i \in M$
$role(v_i) = master$
For all $v_j \in N(i)$ and $role(v_j) = NULL$
$role(v_j) = slave.$
If $role(v_i) = NULL$ and $v_i \notin M$
$role(v_i) = bridge$

Step 2.
For each *candidate bridge* b_i with masters (M_{i1}, M_{i2})
If $role(v_i) = role(v_j) = bridge$ and $v_j \in N^1(v_i)$
If $v_k \nexists (N^1(v_i) \cap N^1(v_j))$ such that $v_k \in M$
If $v_k \exists (N^1(v_i) \cap N^1(v_j))$ such that $role(v_k) = NULL$
$role(v_k) = master$
else
$role(v_i) = role(v_j) = M/Sbridge$

Step 3.
If $role(v_i) = NULL$
$role(v_i) = master.$
For all $v_j \in N^1(v_i) \cap B'$
v_i gets connected with the bridge v_j.
end

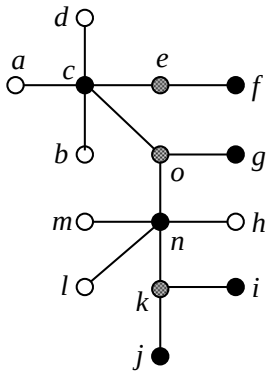


Figure 5.6: Scatternet formed by EESFP protocol

Theorem 5.2 *For a topology graph $G(V, E)$, the procedure RoleAssignment generates a scatternet.*

Proof: It follows directly from Theorem 4.5 that the set $\mathcal{S}_{\mathcal{MB}} = B' \cup M'$ is a connected dominating set of the underlying topology $G(V, E)$. So, $\mathcal{S}_{\mathcal{MB}}$ induces a connected subgraph in G. Let $G'(\mathcal{S}_{\mathcal{MB}}, E')$ be the induced connected subgraph. Let $SG(\mathcal{S}_{\mathcal{MB}}, E'')$ be the scatternet graph formed by the nodes in $\mathcal{S}_{\mathcal{MB}}$. We will first prove that $SG(\mathcal{S}_{\mathcal{MB}}, E'')$ is a scatternet of the subgraph $G'(\mathcal{S}_{\mathcal{MB}}, E')$.

It is to be noted that $E'' \subseteq E'$, as there is no edge between two adjacent S/S bridges in SG, but this edge is present in G'. Also, according to Bluetooth specification two M/S bridges are connected in a scatternet. We now just need to prove that for two S/S bridges v_i and v_j in $\mathcal{S}_{\mathcal{MB}}$ if $(v_i, v_j) \in E'$ then there exists a path between v_i and v_j in SG (i.e., although v_i and v_j are not adjacent in SG but they are still connected in SG). Step 2 of *RoleAssignment* procedure guarantees that no two neighbouring nodes are selected as S/S bridges unless they have common master. When there is a common master between v_i and v_j, they are connected via the common master. This proves that SG forms a scatternet for G'.

Step 3 of *RoleAssignment* procedure ensures that each node gets a role. However, $\mathcal{S}_{\mathcal{MB}}$ being a dominating set of G, the newly assigned nodes in Step 3 must have at least one neighbour in $\mathcal{S}_{\mathcal{MB}}$ which must be a bridge node. So, all the newly assigned nodes are connected to the scatternet $SG(\mathcal{S}_{\mathcal{MB}}, E'')$. Let M'' denotes the nodes selected in Step 3.

The nodes in the set $V - (\mathcal{S}_{\mathcal{MB}} \cup M'')$ are slaves and obviously are connected to their respective masters. So, they are also connected to all nodes in $\mathcal{S}_{\mathcal{MB}}$. This proves that each node $v_i \in V$ is connected to all other nodes in V in the scatternet.

■

5.5 Simulation Studies

Simulation studies have been done on the same environment as presented in Chapter 2. The algorithm in [16] and the EESFP protocol are executed on the same set of

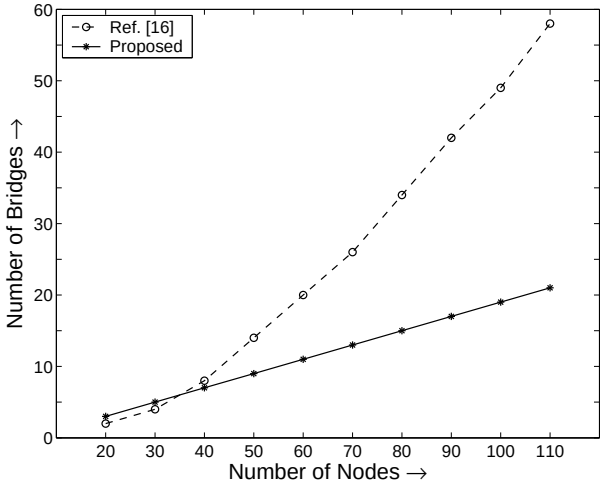


Figure 5.7: Variation of number of bridges with total number of nodes

graphs and the average number of *masters*, *bridges* and *slaves* are used to compare the performances of the two.

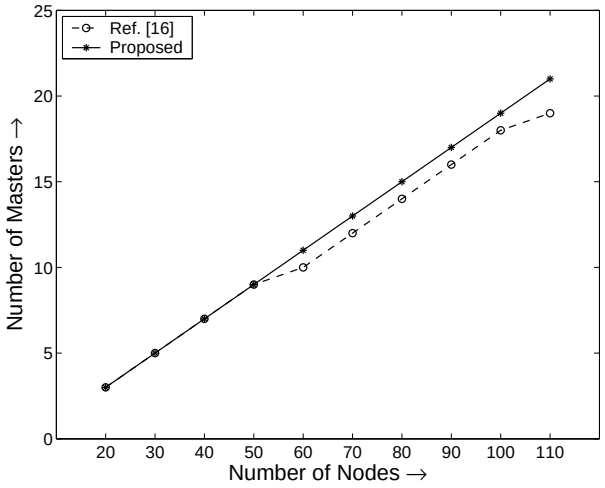


Figure 5.8: Variation of number of masters with total number of nodes

Figure 5.7 shows that for a given number of nodes n, the EESFP protocol forms scatternet with significantly fewer bridges n_b compared to that of [16]. Moreover, in [16], n_b includes both types of bridges (S/S and M/S) whereas EESFP protocol forms the scatternet using only S/S bridges making the scatternet more efficient for communication. Figure 5.8 shows that the number of masters is marginally greater at high values of n in the EESFP protocol. This rise in number of piconets,

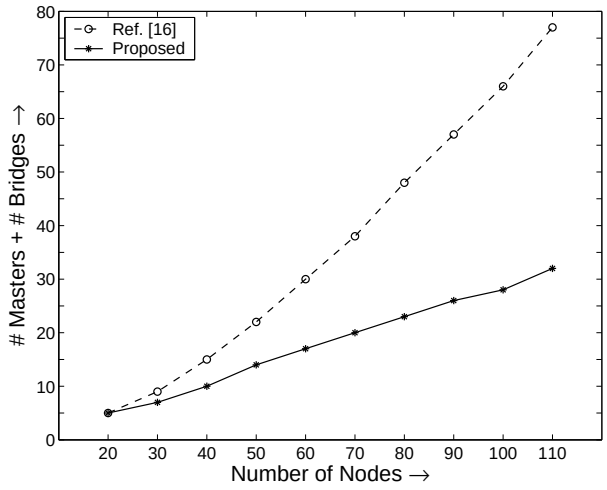


Figure 5.9: Variation of total number of masters and bridges with number of nodes

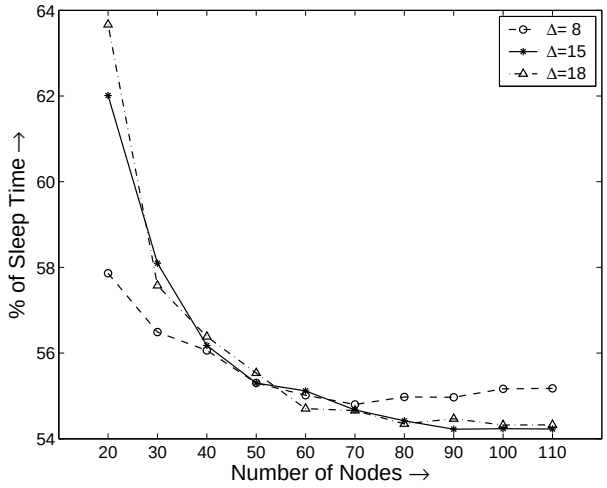


Figure 5.10: Variation of sleep time with number of nodes

in the EESFP protocol, is due to the fact that more number of nodes are active rather than being parked. But the total number of masters and bridges as shown in Figure 5.9 is significantly less than that of [16]. Finally, Figure 5.10 depicts the average sleep time for the nodes represented as a fraction of total computing time. It shows significant energy savings ($\approx 54 - 64\%$) on an average depending on the total number of nodes with Δ as a parameter.

Summary

All the scatternet formation algorithms proposed so far employed the basic Bluetooth specification by which for any message exchange the transmitter and the receiver has to spend a significant amount of time in *busy waiting state* for synchronization. The novelty of the proposed EESFP protocol is that it imposes a synchronization, compatible with the original Bluetooth specification, for all possible pairs of adjacent nodes during neighbourhood discovery phase eliminating the busy waiting state totally in the following phases. Thus, it enables the nodes to remain in the sleep mode during the idle cycles, conserving appreciable amount of energy. Moreover, simulation results show that in terms of number of bridges, types of bridges and total number of masters and bridges, the EESFP protocol outperforms the protocol proposed in [16]. Therefore, given any arbitrary distribution of Bluetooth devices, the proposed protocol enables us to configure a scatternet in a more energy-efficient way that is expected to offer an efficient backbone for quality communication.

Chapter 6

Power Aware Node Based Topology Control

It has been mentioned earlier that each node in an ad hoc network can potentially change the network topology by adjusting its transmission power in a coordinated way. The goal of topology control is to configure the network in an efficient way to achieve a network wide perspective like connectivity, coverage, or energy-efficient communication. Reducing transmission power at nodes not only saves its energy, but also helps in reducing interference effect that in turn improves network capacity and overall throughput.

The power control schemes can be either link-based or node-based. In link-based power control (LBPC), before each transmission, a node adjusts its power level to be just sufficient to communicate along the specific link. As has been explained in Section 1.2.4, link-based power control is, in general, not feasible in mobile ad hoc networks as it demands much communication overhead for link maintenance. In node-based power control (NBPC), each node v_i decides a fixed transmission power $\pi_i \leq P_{max}$, same for all its links that maintains its connectivity with the rest of the network, but at the same time helps to reduce the total power required for communication. Since transmission power of a node need not be adjusted before each communication, except in case of topology changes, its overhead is much lower compared to that of LBPC.

For ad hoc networks with static nodes, complex algorithms can be applied to optimize the node powers efficiently since computation is to be done only during initialization, or in case of failures. Whereas for networks with mobile nodes, the primary objective is fast computation of node powers whenever topology changes, even at the expense of the optimality of solution. With this fact in mind, two distributed algorithms have been presented in this paper for NBPC to build up

a network topology to support energy efficient communication, the first one is for lifetime-critical networks suitable with nodes having less mobility, and the second one is for mobile ad hoc networks where the topology changes frequently due to node mobility.

Chapter Organization

The rest of this chapter is organized as follows. In Section 6.1, we present some of the previous works on power aware topology control, more specifically on the node based power control. In Section 6.2, we discuss the distinct features of our study on NBPC. Our algorithms are presented in Section 6.3. We discuss two NBPC algorithms including their complexity analysis in two subsections of Section 6.3. Simulation results are discussed in Section 6.4.

6.1 Background

Extensive research has been done so far on topology control for wireless ad hoc networks [32, 125, 142, 144]. In case of homogeneous node-based power control (NBPC), where all nodes use same power, a primary area of research is to find the minimum value of the critical range that guarantees certain network properties like connectivity, coverage, energy consumption, or network capacity [114, 142, 143].

With nonhomogeneous NBPC, an important research issue is to compute a set of transmitting range assignments that generates an energy efficient communication graph for broadcast or multicast [171, 164].

All these topology optimization problems attempt to achieve an optimal network topology taking granted that all the information regarding node positions is available at a central node for computation. However, it is not at all a practical approach for mobile ad hoc nodes. Several algorithms, on NBPC, focused on establishing routes and maintaining these routes under frequent and unpredictable topology changes have been reported so far [12, 26, 48]. A 2-approximation distributed algorithm is presented in [100]. A topology control algorithm using heuristics based on a Delauney triangulation of graph is presented in [76]. A centralized spanning tree algorithm for achieving connected and bi-connected static networks, while minimizing the maximum transmission power, is presented in [132]. The distributed algo-

rithms, based on these heuristics, however, do not always guarantee connectivity. A distributed position-based topology control algorithm that preserves connectivity, is proposed in [137]. An improved algorithm is presented in [98].

The best approach of topology control for ad hoc networks is that which uses the minimum amount of information in nodes for the computation, namely, the identity of its neighbours, and possibly some ordering of them based on distance, or link quality, etc. [22, 169]. In fact, this is the minimum amount of information needed by the nodes to configure the topology.

6.2 Our Contributions

Our goal is to configure the network topology in a localized fashion for implementing energy efficient routing with minimum information of neighbour identities and the power demands of links at each node. We develop two greedy distributed NBPC algorithms to cope up with two different situations, one for lifetime-critical networks with nodes having less mobility, and the other for networks with mobility-prone nodes. The first algorithm, namely, the *Incremental Spanning Graph* (ISG) formation, is based on the idea of minimum spanning tree (MST) computation. It takes $O(n)$ rounds and results better optimization in terms of node power levels. The lower transmission power of nodes makes it suitable for lifespan-critical networks. But $O(n)$ time complexity limits its application to highly mobile ad hoc networks where topology changes frequently. So, we propose the second algorithm, namely, the *Localized Neighbour Prunning* (LNP) algorithm. It is based on the neighbour pruning scheme that completes transmission power assignment in two rounds only, i.e., with $O(1)$ time complexity. Though in terms of node power levels, the first algorithm performs better, the second one with less computational complexity can easily adapt to the topology changes, and hence is more suitable for mobile networks. Simulation studies have been done to compare the performances of the proposed schemes with earlier works.

6.3 Proposed Algorithms for NBPC

The NBPC problem is mapped to the problem of finding a *power assignment vector* $\Pi = \{\pi_i \mid 0 < \pi_i \leq P, \forall\, v_i \in V\}$ where π_i is the fixed transmission power assigned to

node v_i such that it maintains the connectivity among the nodes as well as conserves power in communicating among nodes for the underlying weighted topology graph $G(V, E, W)$. It is obvious that the power assignment Π will remove some edges as well as will change the weights $w_{ij} \in W$ of $G(V, E, W)$. These modifications are captured with a new graph, namely, the *power graph*, defined below.

Definition 6.1 *The power assignment vector* Π *for a weighted topology graph* $G(V, E, W)$ *transforms it to a power graph* $G_\Pi(V, E')$ *where* $E' \subseteq E$ *such that an edge* $(v_i, v_j) \in E$ *is also in* E' *if* $min\{\pi_i, \pi_j\} \geq w_{ij}$, *and each edge* (v_i, v_j) *is associated with transmission cost* $w'_{ij} = \pi_i$.

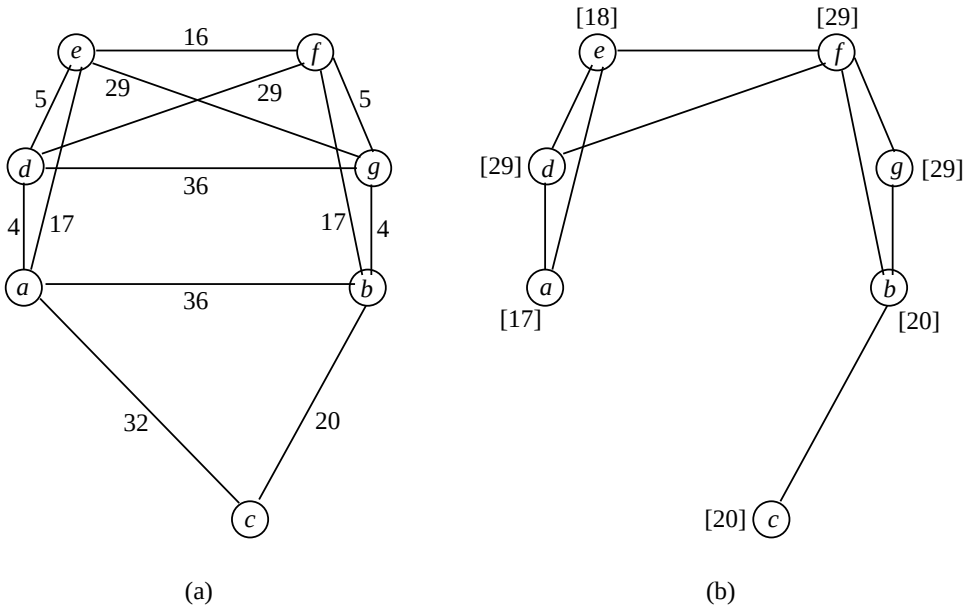


Figure 6.1: (a) A weighted topology graph $G(V, E, W)$ and (b) a power graph $G_\Pi(V, E')$

Example 6.1 *Let* $\Pi = \{17, 20, 20, 29, 18, 29, 29\}$ *be a power assignment vector for the nodes in the weighted topology graph of Figure 6.1(a). Figure 6.1(b) shows the corresponding power graph* $G_\Pi(V, E')$. *In Figure 6.1(b) each node* v_i *is labeled with* π_i. *However, the edge weights* w'_{ij} *are not shown in the figure for clarity. Note that, the edge* $(a, b) \notin E'$ *as* $\pi_a = 17$ *and* $\pi_b = 20$ *but* $w_{ab} = 36$. *Also, the edge* $(e, f) \in E'$ *as* $\pi_e = 18$ *and* $\pi_f = 29$ *and* $w_{ef} = 16$. *But after power assignment on nodes the changed weights on links may be different in two directions. Here,* $w'_{ef} = 18$ *whereas* $w'_{fe} = 29$ *depending on the power level of the corresponding source node.*

It is to be noted that in the power graph $G_{\Pi}(V, E')$, all the links are bidirectional, i.e. if $(v_i, v_j) \in E'$, $(v_j, v_i) \in E'$, but if $\pi_i \neq \pi_j$, then the weights in two directions are different.

In general, for ad hoc networks the nodes are assumed to be fairly mobile and the topology changes frequently. However, in some situations like sensor networks, after deployment the nodes become almost stationary. To cope up with these two different mobility scenarios two different distributed techniques, based on two methodologies, namely, the *minimum spanning tree* (MST) construction (subsection 6.3.1) and the *neighbour pruning* procedure (subsection 6.3.2), have been presented in the following subsections.

6.3.1 Incremental Spanning Graph Algorithm

We adopt the well-known distributed MST construction algorithm ([62, 63]) with a modification where the selection of edges is based on a dynamically computed weight parameter defined below. The resulting graph is defined as *incremental spanning graph* (ISG).

The algorithm starts with each vertex v_i as a distinct component T_i with power assignment $\pi_i = 0$. For each edge $(v_i, v_j) \in E$ of the weighted topology graph $G(V, E, W)$, a weight parameter $w_{ij}^{Tot} = (w_{ij} + w_{ji}) - (\pi_i + \pi_j)$ is defined. There is a designated node in every component called the leader. In each round, with the help of other nodes, each leader of a component finds the edge having minimum value of w_{ij}^{Tot} connecting its component to another component in the topology. The components are joined by the selected edge forming a larger component. Joining components in this way helps to minimize the incremental change in power level in each node in a greedy way, to achieve connectivity. Let two components T_i and T_j be joined by an edge (v_a, v_b) where $v_a \in T_i$ and $v_b \in T_j$. To support this edge the power levels of nodes v_a and v_b are updated to $\pi_a = w_{ab}$ and $\pi_b = w_{ba}$, respectively. Also, node v_a updates weight parameter w_{ak}^{Tot} for all its neighbours v_k outside the new component.

This process of edge selection and component merging continues until a single component is formed containing all n vertices of V. The detailed procedures of *component merging* and *leader selection*, along with their time and message complexities can be found in [62, 63].

A formal representation of the algorithm is given below.

Procedure ISG
begin
Initialization for each node v_i:
  $T_i = \{v_i\}$ and $\pi_i = 0$
  For each edge (v_i, v_k)
    Set $w_{ik}^{Tot} = (w_{ak} + w_{ka})$

In each round for each component T_i:
  Find edge (v_a, v_b) where $v_a \in T_i$, $v_b \in T_j$, $i \neq j$ s.t. w_{ab}^{Tot} is minimum
  Merge two components, $T_i = T_i \cup T_j$ and *elect the leader*
  If $w_{ab} > \pi_a$ then $\pi_a = w_{ab}$
  If $w_{ba} > \pi_b$ then $\pi_b = w_{ba}$
  For each edge (v_a, v_k) where $v_k \in T_q$ and $q \neq i$
    get updated value of π_k from v_k
    $w_{ak}^{Tot} = maximum\ \{(w_{ak} + w_{ka}) - (\pi_a + \pi_k), 0\}$
    If $w_{ak}^{Tot} = 0$
      Merge components: $T_i = T_i \cup T_q$
end

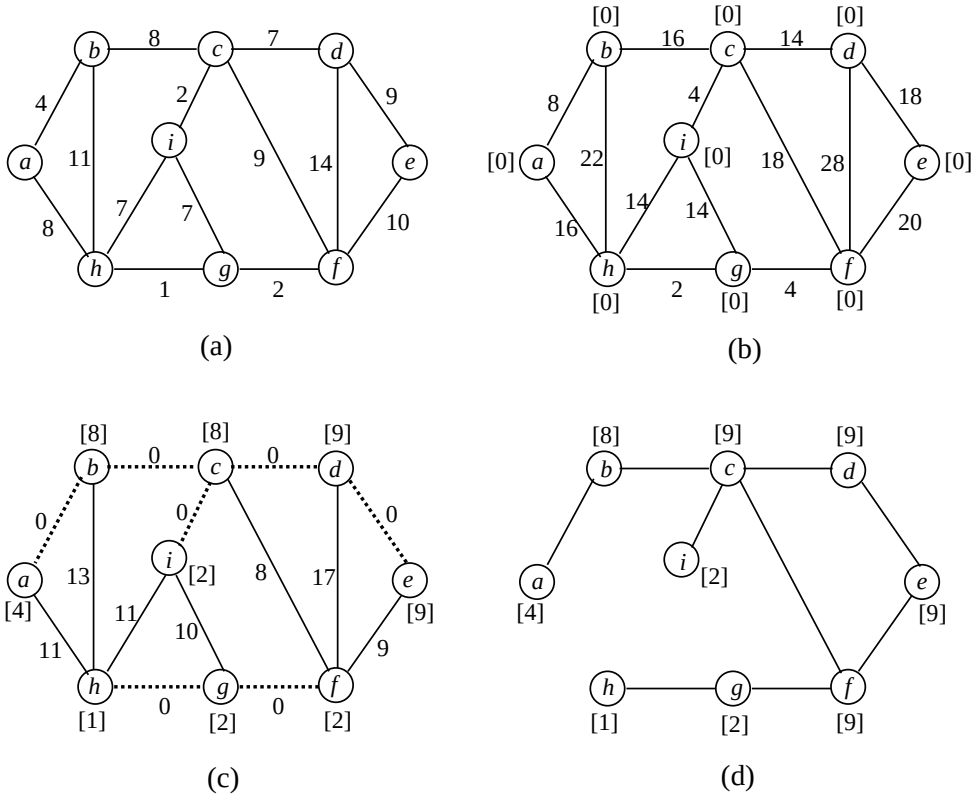


Figure 6.2: (a) Weighted topology graph, (b) initial stage of ISG (c) ISG after round 1, and (d) final power assignment

Example 6.2 *Figure 6.2(a) shows a weighted topology graph $G(V, E, W)$. Figure 6.2(b) shows the same graph after initialization, where label $[\pi_i]$ denotes the power level of node v_i and the label w_{ij}^{Tot} denotes the weight of edge (v_i, v_j). Figure 6.2(c) shows the modified values of Π and w_{ij}^{Tot} after round 1 of ISG formation. In Figure 6.2(c) the dotted lines and the solid lines represent the intra-component and inter-component edges, respectively. The final ISG is shown in Figure 6.2(d).*

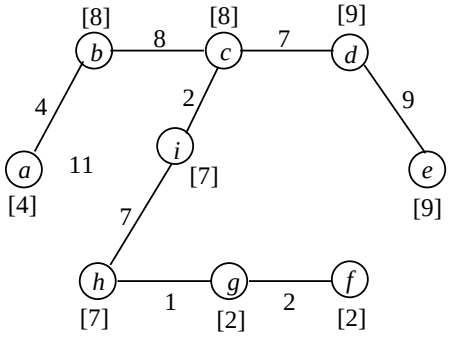


Figure 6.3: MST and its power assignment derived by [85]

Example 6.3 *Corresponding to $G(V, E, W)$ shown in Figure 6.2(a), the ISG generated by the Procedure ISG and the MST derived by [85] are shown in Figure 6.2(d) and Figure 6.3 respectively. Note that for MST, Power Sum $S_\Pi = 56$ which has been improved in ISG as $S_\Pi = 53$.*

Remark 6.1 *The time complexity and the message complexity of the proposed algorithm are the same as those of the distributed MST algorithms. Using the technique suggested in [62], it requires $O(n)$ rounds of time and $O(n \log n)$ message communication.*

6.3.2 Localized Neighbour Pruning Algorithm

Each node v_s first executes the basic edge relaxation operation (as in Dijkstra's shortest path algorithm) [47] on its 2-hop partial graph $PG^2(v_s)$, for the underlying weighted topology graph $G(V, E, W)$. It removes those edges from the graph which are not part of a minimum cost path for any pair of nodes. Here, each node v_s maintains two sets of vertices S and Q. Set S contains all nodes for which minimum cost paths from v_s are already known and set Q contains all other nodes of $PG^2(v_s)$.

It initializes $c[v_s] = 0$ and $c[v_i] = \infty$, $\forall v_i \in PG^2(v_s), i \neq s$, where $c[v_i]$ is the cost of the path from v_s to v_i. Set S is initially empty, and in each step the node with minimum value of $c[v_i]$ is moved from Q to S. $c[v_j], v_j \in Q, \forall v_j \in N^1(v_i)$, is updated to $(c[v_j] + w_{ij})$ if it is less than the existing value of $c[v_j]$. When the set Q becomes empty, each $c[v_i]$ denotes the least cost of the path from v_s to v_i within $PG^2(v_s)$.

Once this preprocessing is over, node v_s removes unnecessary edges in the following two passes and assigns its node power. In the first pass, all edges (v_s, v_i) for which $w_{si} > c[v_i]$ are removed. In the next pass, an edge (v_s, v_i) from the modified $PG^2(v_s)$ is deleted by node v_s, if and only if there exist two edges (v_s, v_k) and (v_k, v_i) each of whose weight w_{sk} and w_{ki} are less than w_{si}.

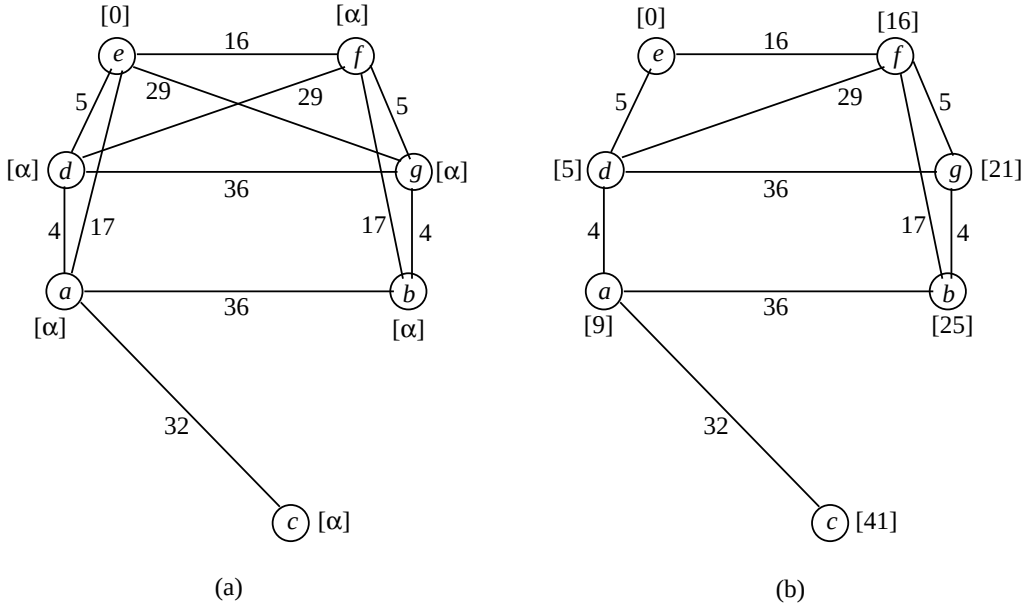


Figure 6.4: (a) $PG^2(e)$ for weighted topology graph of Figure 6.1(a), and (b) $PG^2(E)$ after assigning $c[v_i]$ to nodes

Example 6.4 *For the weighted topology graph $G(V, E, W)$ shown in Figure 6.1(a), the 2-hop partial graph of node e, $PG^2(e)$ is shown in Figure 6.4(a). The initial and final values of $c[v_i]$ are shown as labels $(c[v_i])$ to each node v_i in Figure 6.4(a) and 6.4(b), respectively. Figure 6.5 explains the neighbour pruning in pass 1 and 2 respectively of LNP procedure. In Figure 6.5(b) and (c), the label $[\pi_i]$ to each node v_i represents node power level after pass 1 and pass 2 of pruning respectively.*

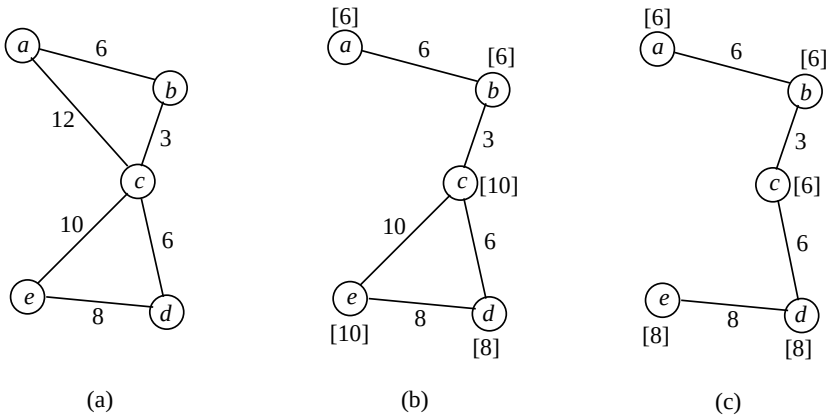


Figure 6.5: Weighted topology graph and power assignment after (a) neighbour pruning, (b) the 1st pass of LNP, and (c) the 2nd pass of LNP

Procedure LNP(v_s)
begin
for each vertex v_i in graph $PG^2(V, E)$ // Initialization
  $c[v_i] = \infty$;
$c[v_s] = 0$; $S = \phi$; $Q = V$
while $Q \neq \phi$ // Assigning minimum cost to vertices
  Find node v_i such that $c[v_i]$ is minimum $\forall v_i \in Q$
  $S = S \cup \{v_i\}$
  for each edge (v_i, v_j)
    if $c[v_j] \geq c[v_i] + w_{ij}$
      $c[v_j] = c[v_i] + w_{ij}$
for each $v_i \in N^1(v_s)$ if $c[v_i] < w_{si}$ // Pass 1 Neighbour pruning
  Delete the edge (v_s, v_i).
for each edge (v_s, v_i) in modified $PG^2(v_s)$ // Pass 2 Neighbour pruning
  if $\exists(v_s, v_k)$ in modified $PG^2(v_s)$ s.t., $w_{sk} < w_{si}$ and $w_{ki} < w_{si}$
    Delete edge (v_i, v_j)
// Transmission power assignment
$P(v_s) = max\{w_{si}\}$, $\forall v_i$ s.t., edge (v_i, v_j) exists in modified $PG^2(v_s)$
Inform neighbours about $P(v_s)$ and modify 1-hop neighbour set $N^1(v_s)$
end

Remark 6.2 *The LNP procedure at each node v_i takes $O((|E'|+|V'|)\log|V'|)$ time for internal computation where, $|V'|$ and $|E'|$ are the number of vertices and edges in $PG^2(v_i)$. This procedure requires one round to know $PG^2(v_s)$ and one round to broadcast its power assignment and the modified 1-hop neighbour set. Message complexity is $2|E|$, $|E|$ is the total number of edges.*

6.4 Simulation Studies

To compare the performances of different power assignment techniques following NBPC, several metrics have been studied so far in the literature.

6.4.1 Performance Metrics

In our simulation study, the following metrics have been considered for performance evaluation.

Definition 6.2 *Given a weighted topology graph $G(V, E, W)$, the shortest-path-power for any two nodes v_i and v_j, $SPP(v_i, v_j)$ is defined as the sum of weights on the edges along the shortest path from v_i to v_j.*

Definition 6.3 *Given a weighted topology graph $G(V, E, W)$, the all-pair-shortest-path-power $APSPP$ is defined as: $APSPP = \sum SPP(v_i, v_j)$, $\forall v_i, v_j \in V$.*

If it is assumed that for message transmission, all source destination pairs occur with equal probability, and routes in general follow the shortest paths, $APSPP$ can be considered as a good measure of the total transmission power required for communication in a round if link-based power control is implemented. However, as it is mentioned earlier, link based power control is not feasible as it needs high overhead in terms of additional communications as well as increased complexity in ad hoc nodes.

For comparison, we introduce some new parameters for networks with a power assignment vector Π.

Definition 6.4 *For a power assignment Π of n nodes of a topology graph $G(V, E)$, the Power Sum, i.e., the sum of power levels of all nodes is represented by $S_\Pi = \Sigma_i^n \pi_i$.*

The power sum S_Π is a measure of total power requirement in case of flooding that is an essential mode of communication for topology discovery or route discovery in ad hoc networks.

Definition 6.5 *Given a power graph $G_\Pi(V, E')$, the shortest-path-power for any two nodes v_i and v_j, $SPP_\Pi(v_i, v_j)$ is defined as the sum of transmission costs on the edges along the shortest path from v_i to v_j.*

Definition 6.6 *Given a power graph $G_{\Pi}(V, E')$, the all-pair-shortest-path-power $APSPP_{\Pi}$ is defined as: $APSPP_{\Pi} = \sum SPP_{\Pi}(v_i, v_j), \forall v_i, v_j \in V$.*

For performance evaluation, a set of performance metrics, namely the power sum S_{Π}, the all-pair shortest path power $APSPP$, which are already defined in section 2.2, and the average hop count H have been considered, and compared with the parameters for link-based power control technique that imposes a lower bound on $APSPP$, and also with the parameters corresponding to the Common-Power algorithm (COMPOW) presented in [114] which gives an upper bound on it.

The COMPOW algorithm assigns a common power level to all nodes, which is the minimum one required to keep the graph G connected. A routing table RT_{P_i} for each admissible power level P_i is constructed by sending and receiving *hello* messages at the power level P_i. Thus, the number of entries in RT_{P_i} (those nodes reachable within a finite number of hops) gives the number of reachable nodes with power P_i. The optimum power P_i selected for the node v_i is the smallest power level whose routing table has the same number of entries as that of the routing table at maximum power P. However, the distributed spanning tree construction procedure can also be used to find the minimum common power.

6.4.2 Results

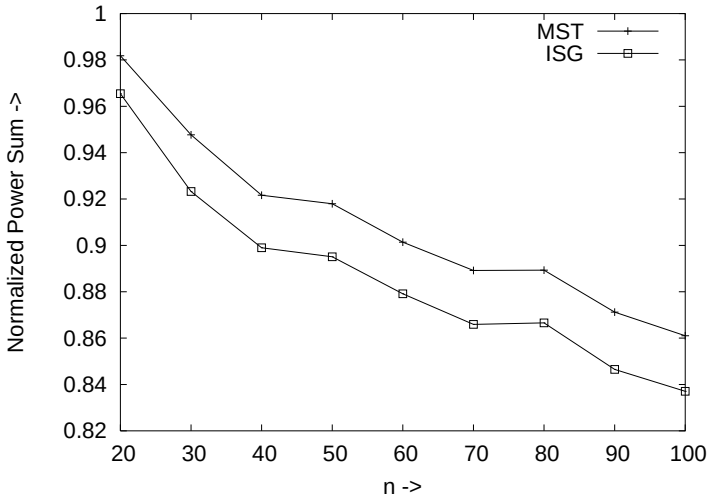


Figure 6.6: Comparison of normalized Power Sum S_{Π} for MST and ISG

The performances of proposed algorithms have been studied by simulation on

random graphs. The results obtained by our algorithms are compared with the minimum spanning tree based 2-approximation algorithm presented in [85] with respect to the performance metric S_{Π}. It is evident from Figure 6.6 that with the same time and message complexity, the proposed ISG technique performs better compared to the algorithm in [85].

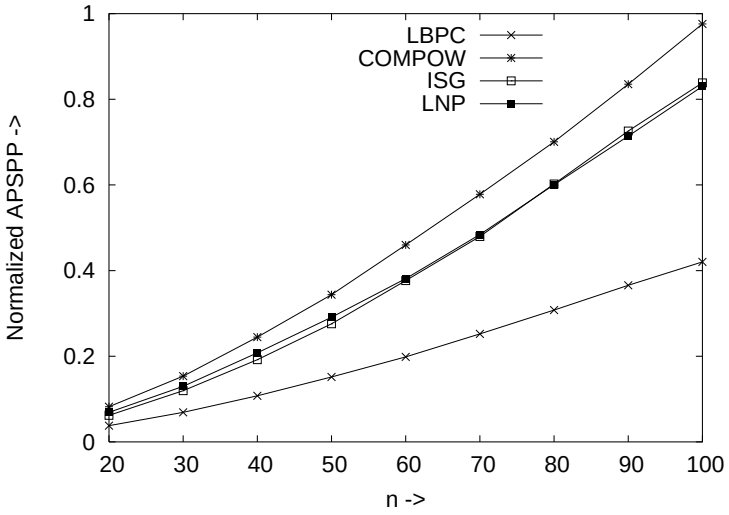


Figure 6.7: Variation of normalized $APSPP$ with n

Figure 6.7 shows the variation of normalized $APSPP$ with the number of nodes n. It is evident that link based power control (LBPC) always results minimum $APSPP$ as it enables a node to communicate with the power just sufficient for the link under consideration. On the other hand, COMPOW algorithm results higher $APSPP$ as all nodes transmit with a common power needed for connectivity. However, the $APSPP$ values produced by the proposed algorithms ISG and LNP are very close and remains in between those given by LBPC and COMPOW.

Figure 6.8 shows the variation of power sum with the total number of nodes n. Since in LBPC the transmission power of a node varies according to the link of communication, power sum S_{Π} does not bear any significance. It is evident that ISG results minimum power sum and COMPOW the maximum, whereas LNP produces S_{Π} in between the two.

Figure 6.9 shows the variation of average hop count with the number of nodes n. It is obvious that COMPOW results minimum number of hops compared to ISG since for the latter the node powers are much less. The average hop counts in LBPC

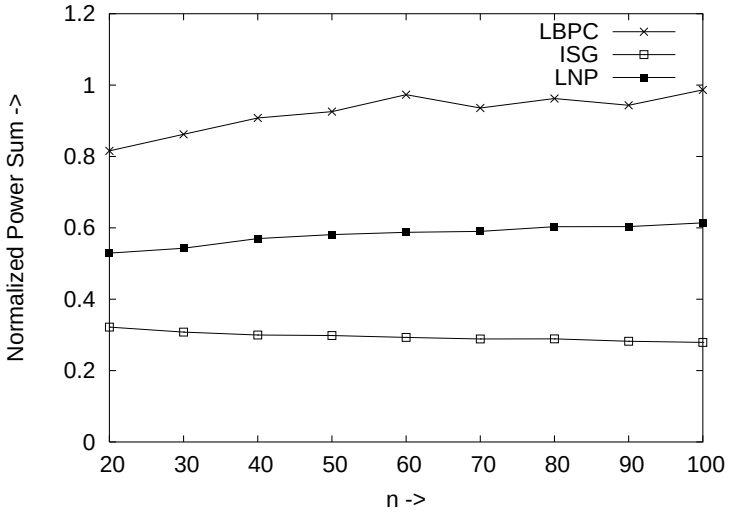


Figure 6.8: Power sum S_{Π} vs n

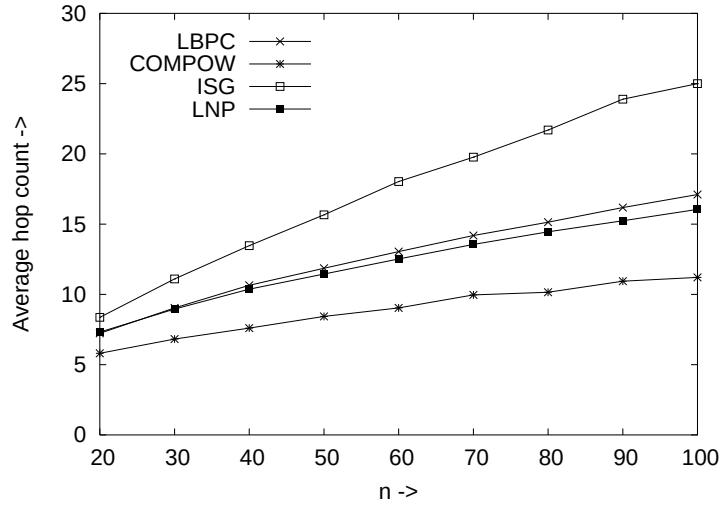


Figure 6.9: Hop count H vs n

and LNP are almost the same; LNP shows better performance for higher n, whereas ISG results highest hop counts and COMPOW the least value for any given n.

6.4.3 Mobility Consideration

For large ad hoc networks with highly mobile nodes, it is evident that the network topology will change frequently. For such networks, ISG technique will not be suitable since, whenever the mobility of nodes causes a change in the topology, the power graph is to be recomputed. The computation needs $O(n)$ rounds which poses

a large overhead on the nodes in terms of message communication and hence the power. Also, the network will need so many rounds to gain stability after each time the topology changes. On the other hand, in such scenarios, LNP technique can perform well since it needs only two rounds for recomputing the power levels after each time the topology changes.

In summary, ISG technique uses higher complexity in terms of rounds and messages, but improves the power sum better. Whereas LNP procedure with less complexity sacrifices a bit in terms of power sum keeping $APSPP$ comparable to that of ISG. Also, it is to be mentioned that the average hop count is much better in LNP compared to ISG and LBPC. So, the ISG algorithm is suitable for NBPC in networks where the lifetime is a more crucial issue than the mobility. On the contrary, for networks where topology changes frequently due to the mobility of nodes, the LNP technique performs better for NBPC by recomputing the node power levels in two rounds only.

Summary

We present two distributed node-based power control schemes, namely the Incremental Spanning Graph (ISG) technique and Localized Neighbour Pruning (LNP) procedure, appropriate for two different situations, one for lifetime-critical networks with nodes having less mobility, and the other for networks with mobility prone nodes. Simulation studies have been done for performance comparison. The ISG algorithm takes $O(n)$ rounds (n is the total number of nodes) to assign transmission power to individual nodes and results better optimization in terms of sum of transmission powers of individual nodes. The LNP procedure is based on neighbour pruning scheme that completes transmission power assignment in just two rounds of message passing. In spite of lower time complexity, it results comparable communication cost for end-to-end packet delivery. It also improves the hop count appreciably. Therefore, depending on the specific application the appropriate power control technique is to be applied to improve the throughput and the lifetime of the network.

Chapter 7

Data Gathering in Multihop Wireless Sensor Networks

A typical Wireless Sensor Network (WSN) consists of a large number of inexpensive sensor nodes, densely deployed over an area to gather information about the surroundings in a self-configured and unattended fashion for a long period of time. The basic operation of WSN is the periodic sensing, gathering and transmission of data by individual nodes to a final *sink* or base station (BS).

In WSN since the nodes are battery driven and there is no facility for recharging in general, nodes must route data in most energy-efficient way. The energy of a node is drained out with time for data sensing, computing and communicating with other nodes and the base station. However, data communication consumes the major amount of energy in WSN.

Following the same energy consumption model described in Section 2.3, we see that the amount of energy spent in transmitting a packet has a fixed cost in electronics and a variable cost that depends on the distance between the communicating nodes, whereas, the reception of a data packet has a fixed energy cost in electronics. In WSN, data from all nodes are collected and transmitted to the BS in periodic intervals called rounds. A *data gathering schedule* specifies how the data packets from all sensor nodes are routed and collected at the base station in each round. According to the most restricted definition, the *lifetime* of a WSN is defined to be the time duration (i.e., the number of rounds) for which the base station can gather data from the sensors.

A simple approach to accomplish this task is for each node to transmit its data directly to the BS. Since the BS is usually located far away, the cost to transmit directly to the BS from any node is high, and therefore, the total energy cost per round will be high. So, it is evident that instead of a single-hop path, it is better

to route data in multihop paths via intermediate nodes. Here, data fusion [101] or aggregation helps to reduce the amount of data transmitted between sensor nodes and the BS. Data fusion combines one or more data packets from different sensors to produce a single packet. The key idea is to fuse data collected from different nodes to eliminate redundancy and to provide a rich aggregated view of the environment keeping the volume of total traffic limited. The exact data aggregation logic depends on the application. Usually, the gathered data move from node to node, get fused, and eventually received at the BS. To maximize network lifetime, configuring efficient routing paths that essentially determines a data gathering schedule emerges to be a critical design issue for WSN.

The problem of finding a data gathering schedule that maximizes the lifetime is in general NP-hard. Moreover, the challenge here is to have a reasonably good data gathering schedule that can also adapt with changes in the topology due to failure of sensor nodes. The location of the base station as well as the data fusion or aggregation logic play crucial roles in finding a good data gathering schedule.

Chapter organization

The rest of this chapter is organized as follows. Section 7.1 outlines the previous works on data gathering in WSN. In Section 7.2, we discuss the salient features of our study on the data gathering problem in WSN. In Section 7.3, we present the system model extension required for WSN over and above the general network model presented in Section 2.2. The data gathering algorithm for multihop WSN including its complexity analysis is presented in Section 7.4. Simulation results are described in section Section 7.5. The chapter concludes with a summary.

7.1 Background

The problem of finding an efficient data gathering algorithm that maximizes the lifetime, referred to as the Maximum Lifetime Data Aggregation (MLDA) problem [79, 82] is in general NP-hard [132].

Extensive research has been done so far on maximizing the lifetime of wireless sensor networks [156, 168]. Several protocols attempt to improve lifetime by reducing energy consumption in MAC for WSN [154, 176, 178]. In [146] the lifetime of WSN

is enhanced by efficient use of MAC. It reduced MAC overhead in WSN by using STDMA and an encoded representation of the addresses in data packets. It is shown by simulation that the MAC overhead is reduced by a factor of three compared to the existing approaches reducing the energy consumption per transmission and reception.

The other approach to improve the lifetime is by partitioning the nodes into two sets, one as operational (*awake*) and the other as backup (*asleep*) [30, 174, 33, 73, 74]. However, such partitioning requires knowledge of their positions using GPS or similar devices.

In [82], the problem is mapped to integer linear programming and later solved in polynomial time by linear relaxation to produce a linear approximation to the optimal solution.

Some study [103, 13] have also been carried out using hierarchical routing based on clustering to improve lifetime. The HEED protocol [177] proposes a distributed clustering approach that enhances the lifetime by distributing energy consumption among the cluster head nodes, terminating within a constant number of steps and considering a combination of energy and communication cost for selecting cluster heads. In LEACH [70], a distributed data gathering procedure is proposed using clustering based protocol. PEGASIS [105] improved the performance by forming a chain with the sensors where each sensor communicates with the base station in turn to deliver the aggregated data. But all these works assumed a single-hop WSN, where each sensor can communicate directly with each other and also with the base station. They also used the global knowledge of topology.

The study on data fusion or aggregation essentially shifted the focus of routing from address-centric approaches to data centric approaches [79, 87, 104, 105, 108] to reduce the size of data. However, the technique can be applied with any of the data gathering schedules proposed so far.

7.2 Our Contributions

We study the problem of data gathering schedule for a realistic deployment of the sensors where each sensor with a limited range is capable of direct communication with only a subset of nodes (sensors or base station) in a single hop, and with the

rest via multiple hops. Starting from a random deployment of sensor nodes over a two dimensional region, we develop a localized distributed algorithm to construct a spanning tree rooted at BS that determines a data gathering schedule with an objective to maximize the network lifetime. The algorithm assumes uniform traffic that is in each round just a single packet is generated in each node. We use the data fusion or aggregation technique presented in [105]. A node may receive data packets from a number of nodes, each packet is of same size, say k bits. It fuses all incoming data as well as its own sensed data, and forwards the aggregated information in a single data packet of k bits to the next node, finally to be forwarded to the BS. Also, the nodes are capable of controlling the transmission power to be just sufficient to reach the desired node. The algorithm is to be executed just once during the initialization and operates until any node dies out when a reconfiguration procedure will span the remaining nodes. The proposed algorithm requires just the neighbourhood information at each individual node. An extensive simulation study shows the proposed algorithm significantly outperforms the scheduling based on the Minimum Spanning Tree (MST), or the Shortest Path (SP) routing techniques. Also, when this algorithm is applied for single-hop WSN, simulation studies show that the proposed algorithm performs marginally better than the well-known algorithms in the literature as more and more nodes die out. Moreover, our algorithm does not restrict the location of the base station with respect to the deployed region and applicable to situations where the base station is located either inside or outside the deployed region.

7.3 System Model Extension

We follow the same network and energy consumption model as described in Chapter 2 with a marginal extension to it. To include the base station in our basic network model discussed in Section 2.2, it is assumed that the system initializes with a set of $(n-1)$ sensor nodes $\{v_2, v_3, \cdots, v_n\}$, randomly distributed over a wide region along with a fixed base station $BS(= v_1)$ as a special node within the deployment region. The base station (or, node v_1 or BS) has unlimited energy and can transmit directly to $(n-1)$ sensor nodes. All the sensor nodes can be modeled as any normal ad hoc node as mentioned in Chapter 2.

7.4 Data Gathering Algorithm

Given a random deployment of sensor nodes having the capability of transmission power control, the proposed algorithm extracts a *rooted spanning tree* with the BS as the root to minimize the energy consumption in message transmission and reception at each node for the underlying weighted topology graph $G(V, E, W)$, where $w_{i,j}$ is the transmission energy required for the link (v_i, v_j) in WSN.

Here, a *rooted tree* is denoted by $T(v_t, V_T, E_T)$, where, v_t is the root of the tree, V_T is the set of nodes and E_T is the set of directed edges of the tree. A *weighted rooted tree* (WRT) denoted by $T(v_t, V_T, E_T, W_T)$ is a rooted tree where $w_{ij} \in W_T$ is the weight of the directed edge $(v_i, v_j) \in E_T$ that actually represents the transmission power required for the link (v_i, v_j).

Definition 7.1 *For a weighted rooted tree $T(v_t, V_T, E_T, W_T)$ the node cost for each node $v_i \in V_T - \{v_t\}$ is $C_i = d_{in} \times Rx + w_{i,out(v_i)}$, where d_{in} denotes the in-degree of v_i, $out(v_i) \in V_T$ is the node to which v_i forwards its data. The maximum node cost is defined as $C_{max} = max\ \{C_i | \forall v_i \in V - \{v_t\}\}$.*

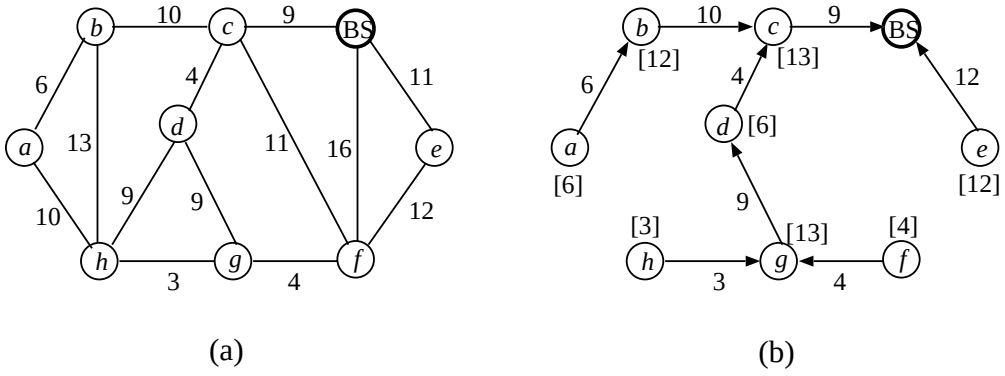


Figure 7.1: (a) a weighted topology graph, and (b) a WRT rooted at BS

Example 7.1 *Figure 7.1(b) shows a weighted rooted tree T with the node BS as the root corresponding to $G(V, E, W)$ shown in Figure 7.1(a). Assuming $Rx = 2$, the node costs are shown as the node labels. For this WRT, $C_{max} = 13$.*

Therefore, the lifetime improvement problem is mapped to the problem of finding a weighted rooted spanning tree $T(v_1, V, E', W')$, (i.e., rooted at BS denoted as

v_1), such that the maximum node cost C_{max} is minimized, where, $E' \subseteq E$, and $w'_{i,j} = w_{i,j}, \forall (v_i, v_j) \in E'$. Then, the nodes can follow a static scheduling determined by $T(v_1, V, E', W')$ unless any node gets drained out when it is needed to recompute the tree again.

The algorithm starts by considering the node $BS(= v_1)$ as the only member of WRT, i.e., $T^0 = (v_1, \{v_1\}, \phi, \phi)$. In kth iteration the tree is $T^k(v_1, V^k, E^k, W^k)$. In $(k+1)$th iteration, a node $v_i \notin V^k$, but is adjacent to some node $v_j \in V^k$, is included in T^{k+1} as an adjacent node to v_j such that it results in a minimum increase in C_{max} of T^k. The node costs are updated accordingly. This process terminates when T covers all n nodes. To execute this greedy algorithm, each node $v_i \in V^k$ maintains the following variables:

C_i : the node cost

$N^1(v_i)$: the set of nodes adjacent to v_i

lcn_i : the neighbouring node of a node v_i where $lcn_i \notin V^k$ and $w_{i,lcn_i} \leq w_{i,j}, \forall v_j \in (N_i - V^k)$. When all $v_j \in N_i$ are within the V^k, $lcn_i = \phi$.

Procedure WRTConstruction
begin
In Step k for each node $v_i \in V^{(k-1)}$
  If lcn_i joined the tree $T^{(k-1)}$ in step $(k-1)$
    Recompute lcn_i
  If v_i joined the tree $T^{(k-1)}$ in step $(k-1)$ or,
      lcn_i is recomputed in step k
    If $w_{i,lcn_i} > C_i + Rx$
      $C_i^L = w_{i,lcn_i}$ and $C_i^H = C_i + Rx$
    Else
      $C_i^L = C_i + Rx$ and $C_i^H = w_{i,lcn_i}$
    Send (lcn_i, C_i^H, C_i^L) message to BS
  If a message is received from BS
    Update $C_i = C_i + Rx$ and inform lcn_i to get connected

In Step k for each node $v_j \notin V^{(k-1)}$
  If received a message from a neighbour v_i
    Set $C_j = w_{i,j}$ and recompute lcn_j

In Step k for BS
  If received (lcn_i, C_i^H, C_i^L)
    Select v_i such that $(C_i^H, C_i^L) \leq (C_j^H, C_j^L)$, $\forall v_j \in V^k$
    Inform v_i to include lcn_i into $T^{(k+1)}$
end

Example 7.2 *Figure 7.2 shows the steps of construction of WRT from* $G(V, E, W)$ *of Figure 7.1(a). Starting with BS, after step* 2*, WRT contains nodes BS, c and*

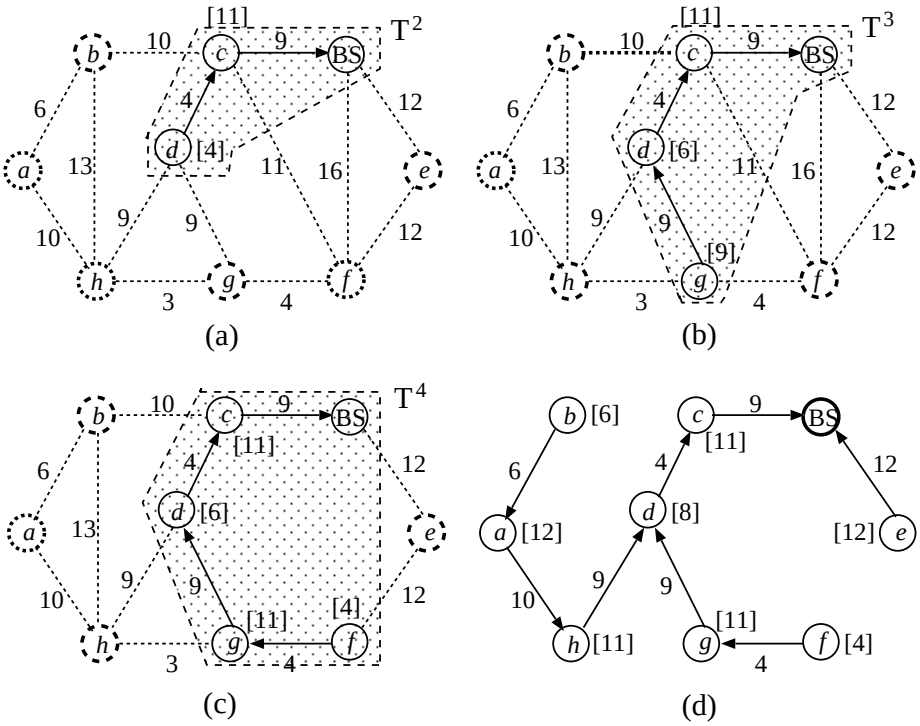


Figure 7.2: The WRT construction (a) after step 2, (b) after step 3, (c) after step 4, and (d) the final WRT

d respectively as shown in Figure 7.2(a). In next step, one node that is adjacent to a node already in T^2 (i.e., adjacent to BS, c or d), is to be included. Based on the computed node costs, node g is included in WRT as shown in Figure 7.2(b). Similarly, WRT in step 4 is shown in Figure 7.2(c). The final WRT is shown in Figure 7.2(d).

Theorem 7.1 *The WRTConstruction procedure terminates in $O(n)$ steps. Both the time and message complexities of the WRTConstruction procedure are $O(n^2)$.*

Proof: The WRT construction starts with the BS as the only node in WRT. In each step of the *WRTConstruction* procedure, a new node is added to the WRT. So, it will terminate after $n-1$ steps.

After $(k-1)^{th}$ step, there will be k nodes in the WRT. But after each addition of the node, a message is sent from the parent of the newly added node to the base station. The maximum length of path that this message has to traverse is $k-1$ to reach the BS. The *WRTConstruction* procedure has to wait for $k-1$ rounds before it can execute the next step (i.e., the k^{th} step). So, the total number of rounds

require to complete the WRT construction is $\Sigma_{i=1}^{n-1}(k-1) \equiv O(n^2)$.

As in each round, exactly one message is transmitted, the message complexity is also $O(n^2)$. ■

The high time and message complexities are due to the fact that we assumed a multihop sensor network. If the direct communication from all the sensors to the BS is possible (i.e., 1-hop sensor network), it can be observed directly from the previous theorem that the resulting time and message complexities are $O(n)$ only.

7.5 Simulation Studies

The performance of the proposed algorithm has been investigated by simulation on random distribution of sensor nodes on a two dimensional plane. For comparison with earlier works, the parameters ε_{elec}, ε_{amp} and k use the same values as in [104]:

$\varepsilon_{elec} = 50\ nJ/bit$ $\quad \varepsilon_{amp} = 100\ pJ/bit/m^2$

$k = 2000$ $\quad$ Initial energy $= 0.25J$

7.5.1 Comparison with MST and SP

For multi-hop sensor networks, the performance of the proposed algorithm is compared with the data gathering techniques via minimum spanning tree (MST) and Shortest Path (SP) routes. Given a topology graph the MST algorithm generates a spanning tree that minimizes the total link cost. Now, it is assumed that data gathering takes place from leaf nodes towards the root (i.e., the base station). Whereas, by SP algorithm, shortest paths in terms of link costs are computed from each sensor node to the base station. Each sensor forwards its data via this shortest path towards the base station.

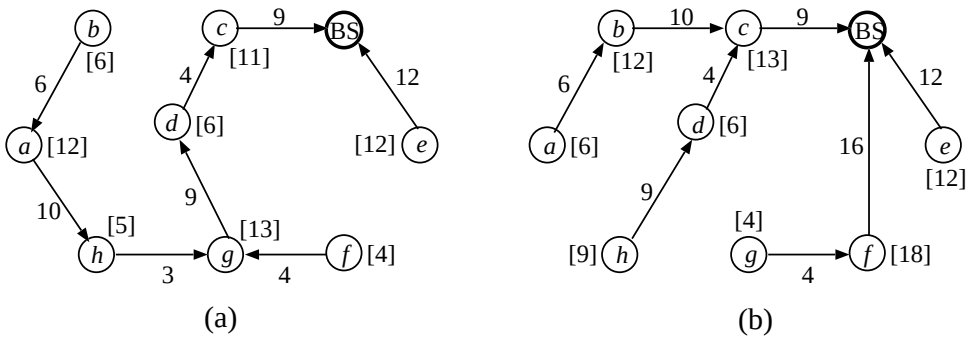


Figure 7.3: Data gathering schedule using (a) MST and (b) SP

Example 7.3 *Given $G(V, E, W)$ shown in Figure 7.1(a), Figure 7.3(a) and Figure 7.3(b) show the data gathering schedules (rooted trees) corresponding to MST and SP respectively. Assuming $Rx = 2$, the node costs are shown as the node labels. It is to be noted that whereas the proposed algorithm results $C_{max} = 12$ (as shown in Figure 7.2(d)), the corresponsing values obtained by MST and SP are* 13 *and* 18 *respectively.*

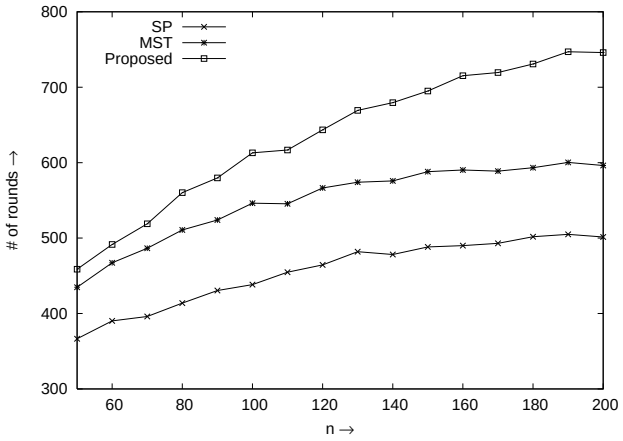


Figure 7.4: Rounds vs n for $range = 50$ units

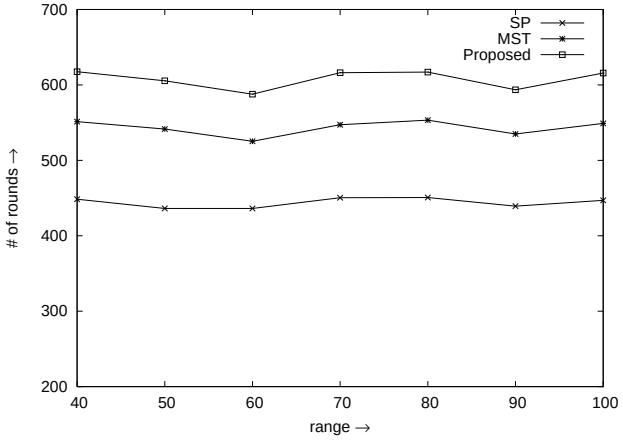


Figure 7.5: Rounds vs range for $n = 100$

The three algorithms are executed on same graph and the corresponding lifetimes are compared. Figure 7.4 shows the variation of lifetime with number of nodes n, $50 \leq n \leq 200$ assuming range $R = 50$ units. For each value of n, 200 random graphs are studied.

Figure 7.5 shows the variation of lifetime with the range R of sensor nodes. With $n = 100$ the range of each node is varied from 40 to 100. For each range, 200 random graphs are studied. In all the cases, the proposed algorithm shows significant improvement in performance [49].

7.5.2 Comparison with PEGASIS

The performance of our algorithm is also compared with PEGASIS [105], where all sensor nodes can access the base station directly. A spanning tree is constructed with the sensor nodes following the algorithm WRT, and each node in turn acts as root and forwards data to the remote base station. Random graphs are generated with 100 nodes distributed over a 50×50 region, and the BS is placed at $(25, 150)$. The range of each node is assumed to be $R = 110$ so that transmission from each node can directly reach BS. Figure 7.6 shows the performance comparison when 1%, 20%, 50% and 100% of nodes die out respectively. It shows that in all the cases the lifetimes are comparable, and also it is marginally better as more and more nodes die out.

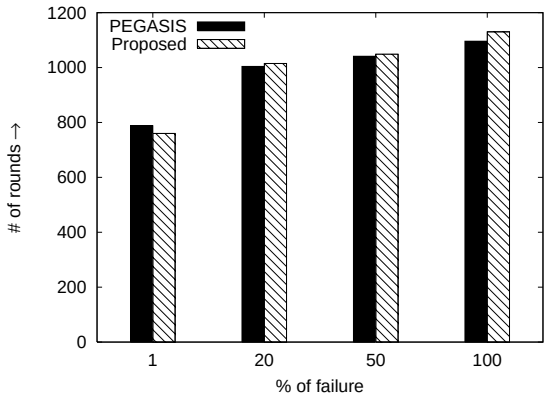


Figure 7.6: Comparison with PEGASIS

Summary

Since the sensor nodes, in general, are severely energy constrained, it is less likely that each node will be able to communicate with the remote base station directly. Therefore, single-hop sensor networks are not feasible in real practice. We developed a distributed algorithm to compute a data gathering schedule for multi-hop WSN.

In worst case, both the time and message complexities are $O(n^2)$. We have shown by simulation that the algorithm improves the lifetime of networks compared to data gathering via Minimum Spanning Tree (MST) or by Shortest Path (SP) routes, and PEGASIS. The proposed distributed algorithm is suitable for large multi-hop sensor networks with nodes having limited transmission range since it requires just the 1-hop neighbourhood information at each node for computation. Also, the algorithm is executed at the time of initialization only, and the network gathers data according to the schedule until a node gets exhausted when recomputation of the routing tree is needed.

Chapter 8

Conclusions

Mobile ad hoc networks must be designed to resolve a number of challenges, including severe energy limitations, robustness, scalability and inherent distributed nature of operation. This thesis considers the problem of initialization and self-configuration of a given distribution of ad hoc nodes over a two dimensional region.

8.1 Contribution of the Thesis

From this work, it is clear that whatever be the network, either a general ad hoc network, or a Bluetooth network, or a wireless sensor network, given any distribution of large number of nodes, it is a very challenging task to design efficient distributed algorithms based on local information only to resolve the problems related to the media access control, and routing and topology control, termed here as the initialization procedures, to build up an efficient network that can function appropriately to achieve its global objective. This work focuses on these issues with special emphasis on optimal utilization of the limited resources, namely the battery power, bandwidth, computing power and storage capacity.

The graph theoretic design concepts and the distributed algorithms developed in this thesis surely will help to advance the state of the art of ad hoc network design. This study shows that the fundamental challenges are to be tackled by appropriate design choices, with possibly cross-layer optimizations. From the classical concepts of unit disk graphs, or random geometric graphs, the establishment of the linear upper bound on the TDMA frame length is definitely a result of theoretical importance. We also have shown that in our works on TDMA frame length minimization and CDS construction, the maximum node degree Δ plays an important role, whereas appropriate power control at individual nodes has a direct impact on Δ. So it is advisable to determine the transmission powers at individual nodes prior

to the execution of the algorithms for TDMA slot assignment in MAC layer, or for CDS construction in the network layer for better message and time complexity. Therefore, with in-built physical and data link layer, using the proposed techniques, the deployed ad hoc nodes may decide on topology control for energy efficient configuration of the communication links. For efficient media access, nodes may apply TDMA with reduced frame length and multiple collision-free slot assignments to keep idle radios in the off state reducing the duty cycle appreciably. In the network layer, nodes can follow backbone based routing to reduce topology maintenance message overhead to minimize energy usage. Scalability concerns are addressed at all layers, though in a limited sense, by emphasizing distributed algorithms with local interactions only.

However, it is true that the proposed protocols are only partial solutions like most of the earlier works in this area. Many open issues remain to be addressed to explore the potential of tomorrow's practical ad hoc networks useful in real life.

8.2 Open Issues

Before ad hoc networks can actually be used in real life, the fundamental issue that should be resolved is how to guarantee cooperation among nodes in a competitive scenario. For instance, say we are interested to provide ubiquitous Internet access to rural areas without any infrastructure. Here, the ad hoc nodes may be managed by different authorities, so it is obvious that a voluntary participation of the nodes in the execution of a network-wide objective cannot be taken for granted. In presence of some *selfish* nodes, it may so happen that packets are not forwarded as they are expected that may cause the network performance to deteriorate with time or to fail in the worst case. In order to alleviate this problem researchers have proposed several techniques. The most natural one is to use incentives in terms of either *reputation* or *monetary transfer*, but it is hard to devise a fair payment strategy [143]. It is reported that a *good topology* for the purpose of cooperative routing demands a relatively dense topology, i.e., higher node degree which is conflicting with the goal of reducing energy consumption and interference [54]. This finding has opened up new research directions in this area.

In this thesis, we have considered the self-configuration problem for ad hoc net-

works with primary focus on reduction of energy consumption at individual nodes. Another important aspect of self-organization is minimization of interference that is essential to improve the signal quality, and hence the network throughput. Very little work has been reported so far to study if the interference-optimal topology is the same as the energy-optimal topology or not. Also, an acceptable model of interference is yet to come up. Some recent results [27, 109] have shown that at least in a worst case, reducing node energy consumption and minimizing interference are conflicting criteria. These issues are to be investigated further introducing a well-defined model for interference in case of multihop communication with a more practical radio channel model.

As we have mentioned in Chapter 2 that the typical radio channel model used for the design of ad hoc networks is based on the log-distance path model which assumes a circular radio coverage with a radius equal to the transmission range. This classical model helped the analysis of properties of the network in a great way. However, this model is sufficiently accurate only in open environments such as free space. In real-world scenarios, more thorough studies are to be done to model the radio channel appropriately.

It is always true that the success of any technology depends on the rate at which various barriers or challenges in using the same are overcome. On the other hand, the barriers still existing in popularising the applications of wireless ad hoc networks provide enough scope for further investigation to the researchers interested in this field.

Bibliography

[1] A. Aggarwal, M. Kapoor, L. Ramachandran and A. Sarkar, "Clustering algorithms for wireless ad hoc networks", *Proc. of the 4th Int. Workshop on Discrete Algorithms and Methods for Mobile Comp. and Comm.*, pp. 54-63, August 2000.

[2] D. P. Agrawal and Q.-A. Zeng, "Introduction to wireless and mobile systems", *Brooks/Cole Publishing, Pacific Grove*, 2003.

[3] D. P. Agrawal and B. Xie, "Encyclopedia on ad hoc and ubiquitous computing", *World Scientific*, 2009.

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

[5] N. Alon, A. Bar-Noy, N. Linial and D. Peleg, "On the complexity of radio communication", *P*roc. of 21st Annual ACM Symp. on Theory of Computation, pp. 274-285, May 1989.

[6] E. Arikan, "Some complexity results about packet radio networks", *IEEE Tran. on Information Theory*, vol. 30, no. 4, pp. 190-198, July 1984.

[7] K. Arvind, "Probabilistic clock synchronization in distributed systems", *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 5, pp. 474-487, May 1994.

[8] F. Atay, I. Stojmenovic, H. Yanikomeroglu, "Generating random graphs for the simulation of wireless ad hoc, actuator, sensor, and internet networks", *Pervasive and Mobile Computing (Elsevier)*, vol. 4, iss. 5, pp. 597-615, October 2008.

[9] The Bluetooth Special Interest Group. http://www.bluetooth.com.

[10] Specification of the Bluetooth System, version 1.1, February 2001.

[11] R. O. Baldwin, N. J. Davis IV, and S. F. Midkiff, "A real-time medium access control protocol for ad hoc wireless local area networks", *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 3, no. 2, pp. 20-27, Apr. 1999.

[12] N. Bambos, "Toward power-sensitive network architectures in wireless communications: concepts, issues, and design aspects", *IEEE Personal Communications Magazine*, vol. 5, no. 3, pp. 50-59, June 1998.

[13] T. Banerjee, B. Xie, J. H. Jun, D. P. Agrawal, "LIMOC: enhancing the lifetime of a sensor network with mobile clusterheads", *IEEE Vehicular Tech. Conf.*, pp. 133-137, Fall 2007.

[14] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A.Woodward, "A distance routing effect algorithm for mobility (DREAM)", *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pp. 76-84, October 1998.

[15] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network", *Proc. of the ACM Int. Symp. on Mobile Ad Hoc Networking and Computing*, pp. 80-91, June 2002.

[16] C. Petrioli, S. Basagni and I. Chlamtac, "Configuring BlueStars: multihop scatternet formation for Bluetooth networks", *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 779-790, June 2003.

[17] V. Bharghavan, A. J. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LAN's", *in ACM SIGCOMM*, pp. 212-225, Aug. 1994.

[18] V. Bharghavan and B. Das, "Routing in ad hoc networks using minimum connected dominating sets", *International Conference on Communications*, pp. 376-380, June 1997.

[19] R. S. Bhuvaneswaran, J. L. Bordim, J. Cui, N. Ishii, and K. Nakano, "An energy-efficient initialization protocol for wireless sensor networks", *IEICE Trans. on Fundamentals*, vol. E85-A, no. 2, pp. 447-454, February 2002.

[20] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function", *IEEE J. on Selected Areas in Comm.*, vol. 18, no. 3, pp. 535-547, March 2000.

[21] P. Bjorklund, P. Varbrand and D. Yuan, "Resource optimization of spatial tdma in ad hoc radio networks: a column generation approach", *Proc. of the IEEE INFOCOM 2003*, pp. 818-824, Apr. 2003.

[22] D. Blough, M. Leoncini, G. Resta and P. Santi, "The k-neighbors Protocol for

for symmetric topology control in ad hoc networks", *Proc. ACM MobiHoc*, pp. 141-152, 2003.

[23] H. L. Bodlaender, T. Kloks, R. B. Tan and J. van Leeuwen, "Approximations for coloring of graphs", *University of Utrecht, Department of Computer Science, Technical Report* 2000-25, 2000 (25 pages).

[24] J. L. Bordim, J. Cui, T. Hayashi, K. Nakano, and S. Olariu, "Energy-efficient initialization protocols for ad-hoc radio networks", *IEICE Trans. Fundamentals*, vol. E83-A. no. 9, pp. 1796-1803, September 2000

[25] J. L. Bordim, J. Cui, N. Ishii, and K. Nakano, "Doubly-logarithmic energy-efficient initializaiton protocol for single-hop radio network", *IEICE Trans. on Fundamentals*, vol. E85-A, no. 5, pp. 967-976, May 2002.

[26] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad-hoc network routing protocols", *Proc. of the ACM/IEEE MOBICOM*, pp. 85-97, October 1998.

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

[28] J-C Cano, J-M Cano, C. T. Calafate, E. Gonzalez and P. Manzoni, "Trade-off between power consumption and performance in Bluetooth", *Ad Hoc and Sensor Wireless Networks*, vol.8, no. 1-3, pp. 141-159, 2009.

[29] M. Cardei, D-Z. Du, "Improving wireless sensor network lifetime through power aware organization", *ACM Wireless Net.*, vol. 11, no. 3, pp. 333-340, May 2005.

[30] A. Cerpa and D. Estrin, "ASCENT: Adaptive self-configuring sensor networks topologies", *Proc. of IEEE INFOCOM*, vol. 3, pp. 23-27 June 2002.

[31] C-F. Chiasserini, I. Chlamtac, P. Monti, A. Nucci, "Energy efficient design of wireless ad hoc networks", *NETWORKING 2002, LNCS 2345*, pp. 376-386, 2002.

[32] C-F. Chiasserini, I. Chlamtac, P. Monti, A. Nucci, "An energy-efficient method for nodes assignment in cluster-based ad hoc networks", *Wireless Networks*, vol. 10, no. 3, pp. 223-231, May 2004.

[33] 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, September 2002.

[34] G. Chen and I. Stojmenovic, "A clustering and routing in wireless ad hoc networks", *Technical Report TR-99-05, Computer Science, SITE, University of Ottawa*, June 1999.

[35] H. Cheng, J. Cao and X. Fan, "GMZRP: Geography-aided multicast zone routing protocol in mobile ad hoc networks", *ACM/Springer Mobile Networks and Applications Journal* vol. 14, no. 2, pp. 165-177, 2009.

[36] H. Cheng, J. Cao, X. Wang, S. K. Das and S. Yang, "Stability-aware multi-metric clustering in mobile ad hoc networks with group mobility", *Wireless Communications and Mobile Computing*, vol. 9, no. 6, pp. 759-771, 2009.

[37] I. Chlamtac, A. D. Myers, V. R. Syrotiuk, G. V. Zaruba, "An adaptive medium access control (mac) protocol for reliable broadcast in wireless networks", *Proceedings of the IEEE International Conference on Communications*, vol. 3, 2000, pp. 1692-1696, 2000.

[38] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges", *Ad Hoc Networks*, vol. 1, no. 1, pp. 13-64, January 2003.

[39] I. Chlamtac, A. Farago, "Making transmission schedules immune to topology changes in multi-hop packet radio networks", *IEEE/ACM Trans. on Networking*, vol. 2, pp. 23-29, February 1994.

[40] A.-M. Chou and V. O. K. Li, "Slot allocation strategies for TDMA protocols in multihop packet radio networks", *Proc. of the IEEE INFOCOM*, pp. 710-716, May 1992.

[41] K. R. Chowdhury, N. Nandiraju, P. Chanda, D. P. Agrawal and Q-A Zeng, "'Channel allocation and medium access control for wireless sensor networks", *Ad Hoc Networks*, vol. 7, iss. 2, pp. 307-321, March 2009.

[42] V. Chvatal, "A greedy heuristic for the set-covering problem", *Mathematics of Operation Research*, vol. 4, no. 3, pp. 233-235, 1979.

[43] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs", *Discrete*

Mathematics, vol. 86, pp. 165-177, 1990.

[44] T. Clausen, G. Hansen, L. Christensen, and G. Behrmann, "The optimized link state routing protocol, evaluation through experiments and simulation", *in IEEE Symposium on Wireless Personal Mobile Communications*, September 2001.

[45] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)". IETF Request for Comments 3626, 2003.

[46] C. Cordeiro and D. P. Agrawal, "Ad hoc & sensor networks, theory and applications", *World Scientific*, Spring 2006.

[47] T. H. Cormen, C. E. Leiserson, R. L. Rivest, "Introduction to algorithms", *MIT Press and McGraw-Hill*, 1998

[48] M. S. Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks", *ACM/Baltzer Wireless Networks*, vol. 1, no. 1, pp. 61-81, February 1995.

[49] D. Couto, S. J. Douglas, "Performance of multihop wireless networks: shortest path is not enough", *Proc. First Workshop on Hot Topics in Networking*, pp. 83-88, October 2002.

[50] F. Cristian, "Probabilistic clock synchronization", *Distributed Computing*, vol. 3, no. 3, pp. 146-158, Springer Verlag, 1989

[51] R. L. Cruz and A. V. Santhanam, "Optimal routing link scheduling and power control in multi-hop wireless networks",*P*roc. of the IEEE INFOCOM 2003, vol. 33, no. 9, pp. 934-944, 2003.

[52] B. Das, R. Sivakumar, and V. Bhargavan, "Routing in ad-hoc networks using a spine", *International Conference on Computers and Communications Networks*, pp. 34-39, September 1997.

[53] R. J. Douglas, "NP-completeness and degree restricted spanning trees", *Discrete Mathematics*, vol. 105, no. 1-3, pp. 41-47, 1992.

[54] S. Eidenbenz, G. Resta and P. Santi, "COMMIT: A sender-centric truthful and energy-efficient routing protocol for ad hoc networks with selfish nodes", *IEEE International Parallel and Distributed Processing Symposium*, vol. 13, pp. 239, 2005.

[55] J. Elson, L. Girod and D. Estrin, "Fine-grained network time synchronization using reference broadcasts", *Proc. of the 5th Symp. on Operating Systems Design and Implementation (OSDI 2002)*, vol. 36, pp. 147-163, 2002.

[56] A. Ephremides and T. V. Truong, "Scheduling broadcasts in multihop radio networks", *IEEE Trans. Communications*, vol. 38, no. 4, pp. 456-460, 1990.

[57] S. Even, O. Goldreich, S. Moran and P. Tong, "On the NP completeness of certain network testing problems", *Networks*, vol. 14, no. 1, pp. 1-24, 1984.

[58] M. Fellows, D. Lokshtanov, N. Misra, M. Mnich, F. Rosamond, S. Saurabh, "The complexity ecology of parameters: an illustration using bounded max leaf number", *Theory of Computing Systems*, vol. 45, no. 4, pp. 822-848, 2009.

[59] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks", *Proc. of ACM SIGCOMM*, pp. 262-273, Aug. 1995.

[60] E. M. Gafni and D. P. Bertsekas, "Distributed algorithms for generating loop-free routes in networks with frequently changing topology", *IEEE Trans. on Comm.*, vol. COM-29, pp. 11-18, January 1981.

[61] G. Galbiati, F. Maffioli, A. Morzenti, "A short note on the approximability of the maximum leaves spanning tree problem", *Information Processing Letters* vol. 52, no. 1, pp. 45-49, 1994.

[62] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees", *ACM Trans. on Programming Languages and Systems*, vol. 5, no. 1, pp. 66-77, Jan 1983.

[63] J. Garay, S. Kutten, and D. Peleg: "A sub-linear time distributed algorithm for minimum-weight spanning trees", *SIAM Journal of Computing*, vol. 27(1), pp. 302-316, 1998.

[64] M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network", *ACM/Baltzer Journal of Wireless Networks*, vol. 1, no. 3, pp. 255-265, February 1995.

[65] J. Gronkvist, "Assignment methods for spatial reuse TDMA", *Proc. of the IEEE Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobilHOC)*, pp 119-124, Aug. 2000.

[66] J. Gronkvist and A. Hansson, "Comparison between graph-based and interference-based STDMA scheduling", *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing*, pp. 255-258, October 2001.

[67] J. Gronkvist, "Novel assignment strategies for spatial reuse tdma in wireless ad hoc networks", *Wireless Networks*, vol. 12, no. 2, pp. 255-265, Apr. 2006.

[68] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets", *Proceedings of the Fourth Annual European Symposium on Algorithms*, Springer-Verlag London, UK, pp. 179-193, September 1996.

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

[70] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks", *Proc. Hawaii Int. Conf. Sys. Sc.*, pp. 1-10, Jan 2000.

[71] T Herman and S Tixeuil, "A distributed TDMA slot assignment algorithm for wireless sensor networks", *Proc. of the 1st Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks*, ALGOSENSORS 2004, Springer LNCS 3121, pp. 45-58, July 2004.

[72] J. Hightower and G. Borriello, "Location systems for ubiquitous computing", *IEEE Computer*, vol. 34, no. 8, pp. 57-66, Aug. 2001.

[73] J. Hong, J. Cao, W. Li, S. Lu and D. Chen, "Sleeping schedule-aware minimum latency broadcast in wireless ad hoc networks", *44th Annual IEEE International Conference on Communications*, pp. 1-5, June 2009.

[74] J. Hong, J. Cao, W. Li, S. Lu and D. Chen, "Minimum-transmission broadcast in uncoordinated duty-cycled wireless ad hoc networks", *IEEE Transactions on Vehicular Technologies*, vol. 59, no. 1, January 2010.

[75] T. C. Hou and V. O. K. Li, "Transmission range control in multihop packet radio networks", *IEEE Trans. on Commun.*, vol. COM-34, pp. 38-44, January 1986.

[76] L. Hu, "Topology control for multihop packet radio networks", *IEEE Trans. on Communications*, vol. 41, no. 10, pp. 1474-1481, October 1993.

[77] IEEE, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications", IEEE Standard 802.11, 1999.

[78] IEEE, "Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)", IEEE Standard 802.15.1, 2002.

[79] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks", *Proc. ACM/IEEE MobiCOM Conf.*, 56-67, Aug. 2000.

[80] D. Johnson, D. Maltz, and J. Broch, DSR: The dynamic source routing protocol for multihop wireless ad hoc networks, ch. 5, pp. 139-172. Addison-Wesley, 2001.

[81] E.-S. Jung and N. H. Vaidya, "A power control MAC protocol for ad hoc networks", *in ACM International Conference on Mobile Computing and Networking (MobiCom)*, pp. 36-47, September 2002.

[82] K. Kalpakis, K. Dasgupta, P. Namjoshi, "Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks", *Comp. Net.*, vol. 42, no. 6, pp. 697-716, Aug 2003.

[83] P. Karn, "MACA - a new channel access protocol for packet radio", *in ARRL/CRRL Amateur Radio Computer Networking Conference*, pp. 134-140, September 1990.

[84] V. Kawadia and P.R. Kumar, "Power control and clustering in ad hoc networks", *Proc. IEEE Infocom*, pp. 459-469, March 2003.

[85] L.M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc, "Power consumption in packet radio networks", *Theoretical Computer Science*, vol. 243, pp. 289-305, 2000.

[86] L. Kleinrock and J. Silvester, "Spatial reuse in multihop packet radio networks", *Proc. IEEE*, vol. 75, no. 1, pp. 156-167, January 1987.

[87] B. Krishnamachari, D. Estrin, S. Wicker, "The impact of data aggregation in wireless sensor networks", *Proc. Int. Work. Dist. Event-Based Sys.*, pp. 575-578, July 2002.

[88] B. Krishnamachari, "Networking wireless sensors", *Cambridge University*

Press, 2006.

[89] S. O. Krumke, M. V. Marathe and S. S. Ravi, "Models and approximation algorithms for channel assignment in radio networks", *Wireless Networks*, vol. 7, no. 6, pp. 575-584, 2001.

[90] S. S. Kulkarni, U. Arumugam, "Collision-free communication in sensor networks", *Proc. of the 6th Int. Symp. on Self-Stabilizing Systems*, pp. 17-31, June 2003.

[91] S. S. Kulkarni and U. Arumugam, "Transformations for Write-All-With-Collision Model", *P*roc. of the 7th Int. Conf. on Principles of Distributed Systems (OPODIS), Springer LNCS 3144, pp. 184-197, December 2003.

[92] S. Kumar, V. S. Raghavan, and J. Deng, "Medium access control protocols for ad hoc wireless networks: A survey", *Ad Hoc Networks*, vol. 4, no. 3, pp. 326-358, May 2006.

[93] L. Lamport, "Time, clocks, and the ordering of events in a distributed system", *Communications of the ACM*, vol. 21, iss. 7, pp. 558-565, 1978.

[94] C. Law and K-Y. Siu, "A Bluetooth scatternet formation algorithm", *Proc. of the IEEE Sym. on Ad Hoc Wireless Networks*, San Antonio, Texas, USA, November 2001.

[95] C. Law, A. K. Mehta, and K-Y. Siu, "Performance of a new Bluetooth scatternet formation protocol", *Proc. of the ACM Symposium on Mobile Ad Hoc Networking and Computing*, Long Beach, California, USA, October 2001.

[96] C. Law, A. K. Mehta and K.-Y. Siu, "A new bluetooth scatternet formation protocol", *ACM/Kluwer J. Mobile Networks and Applications (MONET)*, special issue on mobile ad hoc networks, A. Campbell, M. Conti and S. Giordano, eds., vol. 8, no. 5, October 2003.

[97] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad-hoc routing", *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pp. 120-130, Aug. 2000.

[98] L. Li and J. Y. Halpern, "Minimum energy mobile wireless networks revisited", *Proc. of IEEE Int. Conf. on Communications*, pp. 278-283, June 2001.

[99] X. Li and I. Stojmenovic, "Partial Delaunay triangulation and degree-limited

localized Bluetooth scatternet formation", *Proc. AD-HOC NetwOrks and Wireless (ADHOC-NOW)*, pp. 17-32, September 2002.

[100] N. Li, et al., "Design and analysis of a MST-based distributed topology control algorithm for wireless ad-hoc networks", *IEEE Trans. on Wireless Communications*, vol. 4, no. 3, pp. 1195-1206, May 2005.

[101] A. Li, H. Kang, J. Cao, "Coordinated workload scheduling in hierarchical sensor networks for data fusion applications", *Journal of Computer Science and Technology* (Springer), vol. 23, no. 3, pp. 355-364, May 2008.

[102] B. Liang and Z.J. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management", *Proc. InfoCom*, , pp. 1293-1302, March 2000.

[103] C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265-1275, September 1997.

[104] S. Lindsey, C. S. Raghavendra, K. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics", *IEEE TPDS*, vol. 19, no. 3, pp. 924-935, Sep 2002.

[105] S. Lindsey, C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems", *Proc. of the IEEE Aerospace Conf.*, pp. 1-6, Mar 2002.

[106] M. Liu, J. Cao, Y. Zheng, L. Xie, "A energy-efficient protocol for data gathering and aggregation in wireless sensor networks", *Journal of Supercomputing*, vol. 43, no. 2: pp.107-125, 2008.

[107] N. A. Lynch, "Distributed algorithms", *Morgan Kaufmann Publishers, Inc.*, San Francisco, California.

[108] S. Madden, R. Szewczyk, M. J. Franklin, D. Culler, "Supporting aggregate queries over ad-hoc wireless sensor networks", *Proc. IEEE Work. Mobile Comp. Sys. Appl.*, pp. 49-58, August 2002.

[109] F. H. Meyer, C. Schindelhauer, K. Volbert, M. Grunewald, "Energy, congestion and dilation in radio networks", *Proc. of the 14th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pp. 230-237, 2002.

[110] Gy. Miklos, A. Racz, Z. Turanyi, A. Valko, and P. Johansson, "Performance

aspects of Bluetooth scatternet formation", *Proc. of the 1st Annual Workshop on Mobile Ad Hoc Networking and Computing*, pp. 147-148, August 2000.

[111] S. Murthy and J.J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks", *ACM/Baltzer Mobile Networks and Applications*, vol. 1, no. 2, pp. 183-197, October 1996.

[112] K. Nakano and S. Olariu, "Energy-efficient initialization protocols for radio networks with no collision detection", *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, pp. 851-863, 2000.

[113] K. Nakano, S. Olariu, and A. Zomaya, "Energy-efficient routing in the broadcast communication model", *IEEE Trans. on Parallel and Distributed Systems*, vol. 12, no. 6, pp. 544-557, June 2001.

[114] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas and P. R. Kumar, "Power control in ad-hoc networks: theory, architechture, algorithm and implementation of the COMPOW Protocol", *Proc. of the European Wireless Conference*, pp. 156-162, Italy, February 2002.

[115] A. Nasipuri, J. Zhuang, and S. R. Das, "A multichannel CSMA MAC protocol for multihop wireless networks", *in IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1402-1406, September 1999.

[116] L. Negri, D. Zanetti, R. Montemanni and S. Giordano, "Power-optimized topology formation and configuration in Bluetooth sensor networks: an experimental approach", *Ad Hoc and Sensor Wireless Networks*, vol.6, no. 1-2, 2008, pp. 145-175

[117] R. Nelson and L. Kleinrock: "Spatial-TDMA: A collision-free multihop channel access protocol", *IEEE Trans. Communications*, vol. 33, no. 9, pp. 934-944, September 1985.

[118] S. Olariu and A. Y. Zomaya, "An overview of mobile communications and computing", *A. E. Abdallah (Ed), Springer Verlag, State of the Art Series*, Heidelberg, August 2004.

[119] S. Olariu and Q. Xu, "A simple and robust virtual infrastructure for massively deployed wireless sensor networks", *Computer Communications*, vol. 28, no. 13, pp. 1505-1516, 2005.

[120] S. Olariu and I. Stojmenovic, "Design guidelines for maximizing lifetime and

avoiding energy holes in sensor networks with uniform distribution and uniform reporting", *Proc. IEEE INFOCOM*, pp. 1-12, April 2006.

[121] S. Olariu, "Wireless sensor networks and applications", *Cambridge University Press*, 2006.

[122] F. J. Ovalle-Martinez, I. Stojmenovic, F. Garcia-Nocetti and J. Solano-Gonzalez, "Finding minimum transmission radii and constructing minimal spanning trees in ad hoc and sensor networks", *Journal of Parallel and Distributed Computing*, vol. 65, no. 2, pp. 132-141, February 2005.

[123] J. Park and S. Sahni, "Maximum lifetime broadcasting in wireless networks", *IEEE Transactions on Computers*, vol. 54, no. 9, 2005, pp. 1081-1090

[124] J. Park and S. Sahni, "An online heuristic for maximum lifetime routing in wireless sensor networks", *IEEE Transactions on Computers*, vol. 55, no. 8, 2006, pp. 1048-1056.

[125] J. Park and S. Sahni, "Power assignment for symmetric communication in wireless sensor networks", *Int. Journal on Distributed Sensor Networks*, vol. 5, no. 2, 2009, pp. 185-200.

[126] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks", *Proc. INFOCOM* '97, pp. 1405-1413.

[127] C. E. Perkins, "Ad hoc networking". *New York: Addison-Wesley*, 2001.

[128] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing", *IETF Request for Comments 3561*, 2003.

[129] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers", *in ACM SIGCOMM*, pp. 234-244, Aug. 1994.

[130] C. Petrioli, S. Basagni, I. Chlamtac, "BlueMesh: Degree-constrained multihop scatternet formation for Bluetooth networks", *ACM/Springer Journal on Special Topics in Mobile Networks and Applications*, vol. 9, no. 1, pp. 33-47, February 2004.

[131] L. C. Pond and V. O. K. Li, "A distributed time-slot assignment protocol for mobile multi-hop broadcast packet radio networks", *Proc. MILCOM* '89, pp. 70-74, October 1989.

[132] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment", *Proc. of IEEE INFOCOM 2000*, pp. 404-413, March 2000.

[133] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multi-hop radio networks", *IEEE/ACM Transactions on Networking*, vol. 1, no. 2, pp. 166-177, 1993.

[134] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks", *Wireless Networks* vol. 5, no. 2, pp. 81-94, 1999.

[135] S. Ramaswami and K. K. Parhi, "Distributed scheduling of broadcasts in a radio network". *P*roc. of the IEEE INFOCOM, vol.2, pp. 497-504, April 1989.

[136] T. S. Rappaport, "Wireless Communications: Principles and Practice". *Prentice Hall, NJ*, 1999.

[137] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks", *IEEE J. Selected Areas in Communications*, vol. 17, no. 8, pp. 1333-1344, Aug. 1999.

[138] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks", *IEEE Personal Communications Magazine*, vol. 6, no. 2, pp. 46-55, Apr. 1999.

[139] S. Sahni and X. Xu, "Algorithms for wireless sensor networks", *Intl. J. on Distr. Sensor Networks*, Invited Paper, pp. 35-56, 2004. (also in vol. 1, no. 1, 2005, pp. 35-56)

[140] T. Salonidis, P. Bhagwat, L. Tassiulas and R. LaMaire, "Distributed topology construction of bluetooth personal area networks", *Proc. IEEE Infocom 2001*, pp. 1577-1586, April 2001.

[141] E. Sampathkumar, H. B. Walikar, "The connected domination number of a graph", *J. Math. Phys. Sci*, vol. 13, no. 6, pp. 607-613, 1979.

[142] P. Santi and D. M. Blough, "The critical transmitting range for connectivity in sparse wireless ad hoc networks", *IEEE Trans. on Mobile Computing*, vol. 2, no. 1, pp. 25-39, March 2003.

[143] P. Santi, "Topology control in wireless ad hoc and sensor networks", *John Wiley & Sons, Ltd.*, 2005.

[144] P. Santi, "Topology control in wireless ad hoc and sensor networks", *ACM Comp. Surveys*, vol. 37, no. 2, pp. 164-194, June 2005.

[145] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad hoc networks", *Proc. of the 3rd Int. Symposium on Information Processing in Sensor Networks*, pp. 11- 19, April 2004.

[146] C. Schurgers, G. Kulkarni and M. B. Srivastava, "Distributed on-demand address assignment in wireless sensor networks", *IEEE Transactions on Parallel and Distributed Systems*, vol 13, issue 10, pp. 1056-1065, October 2002.

[147] A. Sen and J. M. Capone, "Scheduling in packet radio networks - a new approach", *P*roc. of the IEEE Global Telecommunications Conf., (GLOBECOM 1999), vol. 1b, no. 1, pp. 650-654, December 1999.

[148] S.-M. Senouci and G. Pujolle, "Energy efficient routing in wireless ad hoc networks", *in IEEE International Conference on Communications (ICC)*, vol. 27, no. 1, pp. 4057-4061, June 2004.

[149] G. Sharma, R. R. Mazumdar and N. B. Shroff, "On the complexity of scheduling in wireless networks", *Proceedings of the 12th annual international conference on Mobile computing and networking*, pp. 227-238, September 2006.

[150] S. Singh and C. S. Raghavendra, "PAMAS-power aware multi-access protocol with signaling for ad hoc networks", *ACM Computer Communications*, vol. 28, no. 3, pp. 5-26, July 1998.

[151] R. Sivakumar, B. Das and V. Bharghavan, "An improved spine-based infrastructure for routing in ad hoc networks", *IEEE Symposium on Computers and Communications*, Athens, Greece. pp. 34-39, June 1998.

[152] J. So and N. H. Vaidya, "A multi-channel MAC protocol for ad hoc wireless networks", *Univ. Illinois Technical Report*, January 2003.

[153] K. Sohrabi and G. J. Pottie, "Performance of a novel self-organization protocol for wireless ad hoc sensor networks", *Proc. of the IEEE 50th Vehicular Technology Conference*, pp. 1222-1226, September 1999.

[154] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network", *IEEE Personal Comm. Mag.*, vol. 7, no. 5, pp. 16-27, October 2000.

[155] R. Solis-Oba, "2-approximation algorithm for finding a spanning tree with maximum number of leaves", *Proc. 6th European Symposium on Algorithms (ESA'98)*, LNCS 1461, Springer-Verlag, pp. 441-452, 1998.

[156] C. Song, J. Cao, M. Liu, Y. Zheng, H. Gong and G. Chen, "Maximizing network lifetime based on transmission range adjustment in wireless sensor networks", *Computer Communications*. vol. 32 , issue 11, pp. 1316-1325, July 2009.

[157] I. Stojmenovic and X. Lin, "Power aware localized routing in wireless networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 11, 1122-1133, November 2001.

[158] I. Stojmenovic, M. Seddigh and J. Zunic, "Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, January 2002, 14-25.

[159] I. Stojmenovic, "Dominating set based Bluetooth scatternet formation with localized maintenance", *Proc. IEEE Int. Parallel and Distributed Processing Symposium and Workshops*, April 2002.

[160] G. Tan, A. Miu, J. Guttag and H. Balakrishnan, "An efficinet scatternet formation algorithm for dynamic environment", *Proc. IASTED Comm. and Computer Networks (CCN)*, November 2002.

[161] Z. Tang and J. J. Garcia-Luna-Aceves, "Hop reservation multiple access (HRMA) for ad-hoc networks", *in IEEE Conference on Computer Communications (INFOCOM)*, pp. 194-201, Mar. 1999.

[162] C.-K. Toh, "Associativity-Based Routing For Ad HocMobile Networks", *Wireless Personal Communications Journal, Special Issue on Mobile Networking and Computing Systems*, vol. 4, no. 2, pp. 103-139, Mar. 1997.

[163] P-J. Wan, K. M. Alzoubi and O. Frieder, "Distributed construction of connected dominated set in wireless ad hoc network", *IEEE INFOCOM*, pp. 1597-1604, June 2002.

[164] P. J. Wan, G. Calinescu, X. Li and O. Frieder, "Minimum-energy broadcasting in static ad hoc wireless networks", *ACM/Kluwer Wireless Networks*, vol. 8, no. 6, pp. 607-617, 2002.

[165] Z. Wang, R. J. Thomas and Z. Hass, "BlueNet - A new scatternet formation

scheme", *Proc. 35th Hawaii Int. Conf. System Science*, vol. 2, pp. 61, January 2002.

[166] L. Wang and S. Olariu, "A unifying look at clustering in mobile ad hoc networks", *Wireless Communications and Mobile Computing*, vol. 4, no. 6, pp. 623-637, 2004.

[167] L. Wang and S. Olariu, "Towards a general-purpose virtual infrastructure for mobile ad-hoc networks", *Ad Hoc and Sensor Networks, Nova Science Publishers*, 2006.

[168] D. Wang, Y-I Cheng, Y-W Wang And D. P. Agrawal, "Nonuniform deployment prolongs lifetime of wireless sensor networks", *Ad Hoc and Sensor Wireless Networks*, vol. 5, no. 1-2, pp. 137-160, 2008.

[169] R. Wattenhofer and A. Zollinger, "XTC: A practical topology control algorithm for ad hoc networks", *Proc. 4th Int. Workshop on Algorithms for Wireless, Mobile, Ad hoc and Sensor Networks*, 2004.

[170] R. M. Whitaker, L. E. Hodge, I. Chlamtac, "Bluetooth scatternet formation: a survey", *Ad Hoc Networks*, vol. 3, no. 4, pp. 403-450, July 2005.

[171] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Algorithms for energy-efficient multicasting in static ad hoc wireless networks", *Journal on Special Topics in Mobile Networking and Applications (MONET)*, vol. 6, no. 7, pp. 251-263, June 2001.

[172] J. Wu, F. Dai, M. Gao and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks", *IEEE/KICS Journal of Communication Networks*, vol. 4, no. 1, pp. 59-70, Mar. 2002.

[173] J. Wu and H.L. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks", *Proceedings of the 3rd ACM international workshop on Discrete algorithms and methods for mobile computing and communications*, pp. 7-14, May 1999.

[174] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing", *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 70-84, July 2001.

[175] X. Xu and S. Sahni, "Approximation algorithms for sensor deployment", *IEEE*

Transactions on Computers, vol. 56, 2007, pp. 1681-1695.

[176] W. Ye, J. Heidenmann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks", *Proc. of the IEEE INFOCOM*, pp. 1567-1576, June 2002.

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

[178] G. Yu, B. Krishnamachari, C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks", *Journal of Wireless Communications and Mobile Computing, special issue on Advances in Resource-Constrained Device Networking*, vol. 7, iss. 7, pp. 863-875, September 2007.

[179] W. Yu, J. Cao, X. Zhou, X. Wang, K. C.C Chan, A. T.S. Chan, H.V. Leong, "A high-throughput MAC protocol for wireless ad hoc networks", *IEEE Transactions on Wireless Communications*. vol. 7, no. 1, pp. 135-145, January 2008.

[180] G. V. Zaruba, S. Basagni, and I. Chlamtac, "Bluetrees - Scatternet formation to enable bluetooth-based personal area networks", *Proc. of IEEE Int. Conf. on Communications*, pp. 273-277, June 2001.

[181] Q. Zhang and D. P. Agrawal, "Dynamic probabilistic broadcasting in MANETs", *Journal of Parallel and Distributed Computing*, vol. 65, iss. 2, pp. 220-233, February 2005.

[182] C. Zhu and M. S. Corson, "A five-phase reservation protocol (FPRP) for mobile ad hoc networks", *Wireless networks*, vol. 7, no. 4, pp. 371-384, July 2001.

[183] H. Zhu, Y. Tang, I. Chlamtac, "Unified collision-free coordinated distributed scheduling (CF-CDS) in IEEE 802.16 mesh networks", *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3889-3903, October 2008.

[184] M. Z. Zamalloa, K. Seada, B. Krishnamachari, and A. Helmy, "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks", *ACM Transactions on Sensor Networks*, vol. 4, iss. 3, August 2008.

Publications from the Thesis

1. S. Bhattacharjee and N. Das: *An Energy-Efficient Distributed Algorithm for Bluetooth Scatternet Formation*, Proc. of the 12th Int. Conf. on Advanced Computing and Communications, pp. 365-373, India, Dec. 2004.

2. S. Bhattacharjee, J. Dattagupta and P. K. Srimani: *An Efficient Coordinated Scheme of Constructing Connected Dominating Set in Ad Hoc and Sensor Networks*, Proceedings of the 12th Int. Conf. on Advances in Computing and Communications, pp. 402-411, Ahmedabad, India, Dec. 2004.

3. S. Bhattacharjee and N. Das: *Distributed Time Slot Assignment in Wireless Ad Hoc Networks for STDMA*, Proceedings of the 2nd Int. Conf. on Distributed Computing and Internet Technology, LNCS 3816, pp. 93-104, Bhubaneswar, India, Dec. 2005.

4. S. Bhattacharjee and N. Das: *Distributed Node-Based Transmission Power Control for Wireless Ad Hoc Networks*, Proceedings of the 3rd Int. Conf. on Distributed Computing and Internet Technology, LNCS 4317, pp. 49-62, Bhubaneswar, India, Dec. 2006.

5. S. Bhattacharjee and N. Das: *Distributed Data Gathering Scheduling in Multi-hop Wireless Sensor Networks for Improved Lifetime*, Proceedings of Int. Conf. on Computing: Theory and Applications, pp. 46-50, India, Mar 2007.