Power-based topology control for mobile ad hoc networks

Tadesse, D.G.

Award date:
2012

TECHNISCHE UNIVERSITEIT EINDHOVEN

Power-based Topology Control for Mobile ad hoc Networks

by

Daniel G. Tadesse

Supervisor TU/e: prof.dr.ir. Erik Fledderus
Supervisor WMC: Jan Stoter

A thesis submitted in partial fulfillment for the degree of Masters of Science

in
Electrical Engineering

September 2009

TU/e Technische Universiteit Eindhoven University of Technology

WMC Twente Institute for Wireless and Mobile Communication

Abstract

Mobile ad hoc networks (MANETs) are wireless networks without any fixed infrastructure. When nodes use maximum transmission range, not only the power consumption increases but also the interference on each other increases. This has a negative effect on both energy conservation and network capacity. This becomes even worse when the network density further increases. One method to avoid this problem is to reduce the transmit power level with the constraint that the network should be connected. This is called power-based topology control. The problem of topology control has been extensively studied and protocols are proposed by authors. The methods can be catagorized into three groups. Methods which rely on the availability of location information (e.g. GPS) for the protocols to work are called location-based methods. Methods which rely on the availability of direction of nodes (e.g. using directional antenna and angle-of-arrival estimation techniques) are called direction-based methods. Methods which leave the assumption of additional hardware and rely on the quality of the links with their neighbors are called neighbor-based methods.

We have a network of mobile nodes which are equiped with a single omnidirectional antenna. The nodes are not equiped with any hardware which give location or direction of nodes. So the neighbor-based methods are possible candidates for our problem at hand. The XTC protocol is a neighbor-based topology control protocol which relies on distance estimation using received signal strength indicator (RSSI) and which preserves network connectivity under the assumption that the distance estimated by both of the communicating nodes is the same. Practically, this is not true, in which case the XTC protocol does not preserve connectivity. A number of protocols were proposed to avoid the disconnectivity problem of XTC. However, some of the protocols use link quality metrics which may not give efficient topology interms of energy or capacity. Still some authors use assumptions which are not practical. Moreover, the protocols proposed assume that the network topology is static. However, since we have mobile nodes, the network topology is dynamic. They also assume that the transmit power of a node can vary continuously. However, there are fixed discrete power levels.

In this thesis, we study a power-based topology control solution which can be practically implemented, is mobility adaptive and takes into account the available power levels for the wireless card. We study the various steps of power-based topology control solution from the practical point of view using XTC algorithm as our link selection algorithm. A mathematical analysis is done for the various performance parameters such as energy consumption, capacity and network delay. The analytical results show that decreasing the transmission range of a node, which also means decreasing the node degree, improves the network capacity and energy efficiency. However, this results in communication along multiple hops, and consequently, the network delay increases. The performance of our studied solution is compared to two (modified) protocols by simulation. The simulation results show that our studied solution performs better than the others interms of connectivity and throughput, but the XTC protocol performs better interms of node degree (and consequently energy consumption), but it has some proportion of disconnected nodes.

Acknowledgements

It is an honor for me to thank my university supervisor from TU/e - prof.dr.ir. E. Fledderus and my company supervisor from WMC - Jan Stoter for their guidance, encouragement and comments throughout my thesis work.

I am also deeply grateful to Jan Stemerdink for his help with the Java Simulator package he developed. He has made available his support any time I was in need.

My deepest gratitude goes to dr.ir. J.M. Vleeshouwers, without whom this thesis would not have been possible, for his follow up and encouragement.

I am also deeply grateful to all of my friends who were always there when I was in need.

Finally, I would like to thank all members of my family for their love and encouragement throughout my thesis work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Mobile ad hoc networks (MANETs) are wireless networks without any fixed infrastructure where nodes can be deployed randomly and the nodes are free to move [1]. Nodes communicate directly to each other possibly along multiple paths. Ad hoc wireless networks are more advantageous than infrastructure-based networks in applications like rescue and disaster relief operations. In case of disaster where no infrastructure is left, infrastructure-based networks can not be used because the infrastructure should be reinstalled before they are used. On the other hand, since ad hoc networks do not need any infrastructure to work, they can be used for such applications. Even distant rescuers would be able to communicate using other rescuers in between as a relay. Another application of wireless ad hoc networks is in military where building fixed infrastructure is not possible.

Because of the fact that wireless ad hoc networks lack centralized control, there are many challenges to be faced for a practical implementation of ad hoc network services. These challenges include [1]:

- *Energy conservation:* Nodes in MANETs are battery equipped most of the time. To increase the network lifetime, the limited energy resource should be used as efficiently as possible.
- *Dynamic network topology:* Network nodes are deployed arbitrarily in a certain region and are mobile. Hence, the network topology is dynamic. This arbitrary and time-varying nature of the network topology imposes challenges in network design.
- *Low quality communication:* The quality of communication on a wireless channel is influenced by environmental factors such as presence of obstacles between the transmitter and the receiver, and interference from other sources. As the nodes are typically mobile, the environmental factors are also time-varying. Thus, this time-varying link conditions put a challenge on MANET applications.
- *Limited network capacity:* Generally, the bandwidth availability for MANETs is small compared to other wireless networks, eg. cellular networks. Protocols for MANETs should use this small bandwidth as efficiently as possible.

Among the above outlined challenges, energy conservation and network capacity, can be tackled by using topology control.

1.1 Topology Control

Topology control is the art of coordinating nodes' decisions regarding their transmitting ranges in order to generate a network with desired properties (eg. connectivity) while reducing node energy consumptions and/or increasing network capacity [2]. In [2], a motivation for topology control regarding energy conservation and network capacity is given.

1.1.1 Topology Control and Energy Conservation

Suppose node u must send a packet to node v, which is at distance d as shown in Figure 1.1. Node v is within u's transmitting range at maximum power, so direct communication between u and v is possible. However, there exists also a node w in the region C circumscribed by the circle of diameter d that intersects both u and v. Since $\delta(u,w) = d_1 < d$ and $\delta(v,w) = d_2 < d$, sending the packet using w as a relay is also possible.We want to see which of the two alternatives is more convenient from the energy-consumption point of view. Assuming the radio signal propagates according to the free space model and that we are interested in minimizing the transmit power only, the power needed to send the message directly from u to v is proportional to d^2. In case the packet is relayed by node w, the total power consumption is proportional to $d_1^2 + d_2^2$. Consider the triangle $\widehat{uwv}$, and let γ be the angle opposite to side uv. By elementary geometry, we have $d^2 = d_1^2 + d_2^2 - 2d_1d_2cos\gamma$. Since w is in the maximum transmission range of both u and v, $cos\gamma \leq 0$, we have that $d^2 \geq d_1^2 + d_2^2$. It follows that, from the energy-consumption point of view, it is better to communicate using short, multihop paths between the sender and the receiver.

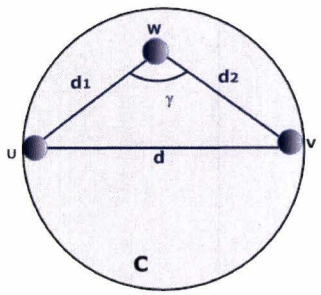

FIGURE 1.1: A network of three nodes to demonstrate energy consumption and topology control

1.1.2 Topology Control and Network Capacity

The amount of interference between concurrent transmissions is related to the transmission range of a node. Based on the interference model used in [3] as shown in Figure 4.5, the packet transmitted by a certain node u to node v is correctly received if $\delta(v,w) \geq (1+\eta)\delta(u,v)$ for any other node w that is transmitting simultaneously, where $\eta > 0$ is a constant which models situations where a guard zone is specified by the protocol to prevent a neighboring node from transmitting on the same channel at the

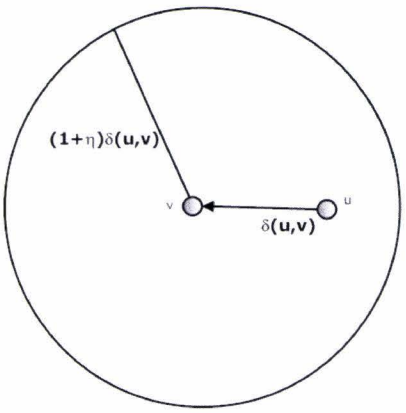

FIGURE 1.2: The protocol model for modeling interference.

same time. Thus, when a certain node is receiving a packet, all the nodes in its interference region must remain silent in order for the packet to be correctly received. The interference region is a circle of radius $(1+\eta)\delta(u,v)$ (the interference range) centered at the receiver. Since concurrent nonconflicting communications occur only outside each other interference region, the area of the interference region measures the amount of wireless medium consumed by a certain communication. Suppose node u must transmit a packet to node v, which is at distance d. We want to see if it is preferable to send the packet directly from u to v or to use two short transmissions using w as a relay from the network capacity point of view. Consider Figure 1.1. Let us consider the interference range(s) in the two scenarios. In case of direct transmission, the interference range of node v is $(1+\eta)d$, corresponding to an interference region of area $\pi d^2(1+\eta)^2$. In case of the two-hop transmission, we have to sum the area of the interference regions of each short, single-hop transmission. The interference region for any such transmission is: $\pi d_1^2(1+\eta)^2 + \pi d_2^2(1+\eta)^2 = \pi(1+\eta)^2(d_1^2+d_2^2)$. Since w is in the transmission range of both u and v implies that $cos\gamma \leq 0$, we have that $d^2 \geq d_1^2 + d_2^2$. We can conclude that, from the network capacity point of view, it is better to communicate using short, multihop paths between the sender and the destination.

1.2 Thesis Overview

This thesis is organized as follows. Chapter 2 reviews the various works on topology control. It revises the three categories of topology control protocols with emphasis on neighbor-based techniques. Chapter 3 discusses the various steps of topology control solution from the practical point of view, including neighbor discovery, neighbor information exchange, link selection and power adjustment. It also discusses how we can handle node mobility. Chapter 4 is dedicated to the mathematical analysis of the various performance parameters including energy consumption, network capacity and network delay. A reconfiguration interval for the topology control algorithm and the duration of the reconfiguration and the consumed energy consumed during the reconfiguration is also discussed in this chapter. In Chapter 5, we presented the simulation results for the various protocols and we evaluate the performances of the various protocols with respect to the various performance metrics including connectivity, node degree, throughput and

energy consumption. Finally, Chapter 6 concludes this thesis with summary of the work and recommendation for possible extension of this work.

Chapter 2

Literature Review

Topology control using transmit power adjustment has been extensively studied. Different protocols have been suggested by authors. We can classify them into centralized and distributed computing methods.

The centralized topology control methods, such as [4, 5] assume that a central entity knows the location of each node and is capable of determining the optimum transmission power of each node through the collected global information. Although this centralized method looks simple, it is not scalable. Moreover, such a central entity is against the nature of ad hoc networks in which it normally lacks infrastructure.

The distributed topology control methods such as [6–16] have the advantage of scalability and adaptation to mobility of nodes whereby each node makes a local decision of the suitable transmission power based on the gathered information from nearby neighbors. Therefore, centralized methods are not practical for mobile ad hoc networks and are not discussed here. Distributed methods can be classified into three groups [1]; namely, location-based methods, direction-based methods and neighbor- or cost-based methods.

The first group of algorithms, such as [6, 8, 11, 12, 16] are called location-based methods. In location-based methods, it is assumed that each node knows its accurate location (e.g. using GPS receivers). Location-based methods are not practical because, to be equipped with a positioning device not only increases the cost of hardware deployment but also brings about several other disadvantages. On one hand, currently, location-based methods work best outdoors, but new methods may appear indoors. On the other hand, the acquisition of location information will introduce computation delay, extra message overhead and energy consumption at each node.

The second group of algorithms, such as [7, 15] are called direction-based methods. In direction-based methods, it is assumed that each node knows the direction of its neighbors by using direction of arrival (or angle of arrival-AoA) estimation by equipping nodes with more than one directional antennas [17]. So, in the case of directional information also, extra hardware on the nodes is needed in order to provide the requested information.

The third group of algorithms, such as [9, 10, 13, 14], called neighbor-based methods, relies on the nodes' ability to determine the number and identity of neighbors within the maximum transmission range and to determine and compare the qualities of the links to all the neighbors and choose the best links.

Given the fact that our nodes which make up the MANET are equipped with single omnidirectional antenna and that they are not equipped with any extra hardware which can give location information of the mobile node, neighbor-based methods seem to be the most convenient methods for our problem at hand.

In the following, we present one direction-based protocol, the Cone-based Topology Control (CBTC) protocol presented in [7], one location-based protocol, the Local Minimum Spanning Tree (LMST) protocol presented in [8], and five neighbor-based protocols, the k-Neighbors (K-Neigh) protocol presented in [9], the X[1] Topology Control (XTC) protocol presented in [10], the Randomized Topology Control (RTC) and ε-RTC protocols presented in [13] and the Localized Topology Control Algorithm (LTCA) protocol presented in [14].

2.1 The CBTC Protocol

The CBTC protocol [7] is the first distributed topology control protocol. It is based on direction information. The CBTC protocol is composed of two phases. In the first phase, every node determines the minimum power needed to reach a neighbor in every direction (i.e. every cone with angle α). Initially, node u sends a broadcast beacon message, which contains the node ID at power p_0. Every node which receives the beacon messages responds with an acknowledgment (ACK) message with the same power used to send the beacon messages. When receiving the ACK message, node u stores the identity of the new neighbor and determines its relative direction using angle-of-arrival technique (using directional antenna). After all the ACKs for power level p_0 have been collected, each node u sends beacons with a growing power. If node u discovers a new neighbor node v, node u will put v into its neighbor list. Node u will continue to grow the transmission power until its neighbor set is big enough such that, for any cone with angle α, there is at least one neighbor, or until node u hits the maximum transmit power P_{max}.

In the second phase, energy-inefficient links are identified and removed from the topology by exchange of local transmission powers. If node u has two neighbor nodes $v, w \in N(u)$, such that the power needed to send from u to w directly is not less than the total power to send via v, we can remove w from $N(u)$. This way the final topology is obtained.

To sum up, the CBTC protocol has several nice features: it is fully distributed, is localized, preserves network connectivity and has bounded logical node degree (see Appendix A for a difference between logical and physical node degree). However, it requires directional information which is typically provided using expensive directional antennas.

2.2 The LMST Protocol

The LMST protocol [8] is a distributed topology control protocol based on location information. In this protocol, it is assumed that all the nodes have the same maximum transmit power and that the wireless medium is symmetric.

[1] By the date of submission the authors have not yet been able to agree on the meaning of the letter "X" in "XTC". However, the authors list the candidates comprising terms such as "exotic", "extreme", "exceptional", or "exemplary", but also "extravagant" or even "extraterrestrial".

Initially, each node sends its ID and location to all nodes in the visible neighborhood by sending a broadcast beacon message at maximum power. Once all the beacon messages of the visible neighbors have been received, each node constructs its local minimum spanning tree (MST) by applying the Prim's algorithm [18]. The link weight used to build the MST is its length (Euclidean distance). After Prim's algorithm execution, every node u in the network knows its MST connecting u to all its visible neighbors. Then every node u selects its final neighbor list. Node v is a neighbor of node u if and only if v is a one-hop neighbor of u in its minimum spanning tree. Finally, the transmit power is set to the level to reach the farthest neighbor node.

In [8], it is shown that the topology produced by LMST preserves connectivity and has maximal logical node degree equal to 6. It only requires exchanging n messages, where n is the number of network nodes. However, since it requires location information, it has additional cost of hardware and energy consumption at each node..

2.3 The k-Neigh Protocol

The k-Neigh protocol introduced in [9] is a distributed topology control protocol based on distance estimation. The algorithm is based on distance estimation using received signal strength indication (RSSI). Initially, a node broadcasts a beacon message at maximum power. When a node receive the beacon message, it registers the IDs of its neighbors along with their estimated distances from it. Then every node makes a distance-based ordering of the neighbors. Based on this ordering, every node selects the first k neighbors and broadcast this to its neighbors at maximum power. Then every node determines the set of symmetric neighbors and the asymmetric links are removed. The k-Neigh protocol is simple, is based on low-quality information (i.e.distance between nodes), has low message exchange (i.e. only $2n$ messages) and generates a topology with bounded physical node degree. However, k-Neigh does not preserve network connectivity in the worst case (i.e. for any node placement). More details on worst-case and average-case network performances can be found in [19].

2.4 The XTC Protocol

In [10], a neighbor-based topology control protocol called the XTC protocol is presented. XTC also uses distance estimation as the link metric. Before presenting the protocol, we need some notation. Let us consider a certain node u, and let $N(u)$ be its neighbor set (i.e. the set of nodes within u's transmitting range at maximum power). In the following, we denote the order relation on $N(u)$ by $\prec_u$; in particular, $w \prec_u v$ means that node w precedes node v in the ordering of node u. In terms of link quality, $w \prec_u v$ indicates that link (u, w) has relatively higher quality than link (u, v).

Initially, each node sends broadcast beacon messages at maximum power. When a node receives the beacon message, it measures the signal strength (using RSSI) and estimates the distance of the node and registers this distance along with the node ID. Then every node makes a distance-based ordering. Each node then broadcasts this ordered list at maximum power. The final step of XTC is the link selection procedure and this can be done locally. When considering a certain node, say node v, node u checks if there is a

third node w with $w \prec_u v$ such that $w \prec_v u$. If this condition is satisfied, v is removed from u's neighbor list; otherwise, it is included in u's neighbor list. The authors in [10] show that, in the ideal case that RSSI gives exact distance estimate, the XTC protocol preserves network connectivity and has bounded logical node degree.

2.5 The RTC and ε-RTC Protocols

XTC, which is the first location-independent protocol to produce a connected topology is based on the assumption that RSSI gives correct distance estimate. However, RSSI does not give correct distance estimate. When nodes estimate distances with errors, XTC does not preserve connectivity. An instance where XTC produces disconnected network when nodes estimate distances with errors is shown in [13] and [14] which is also shown here.

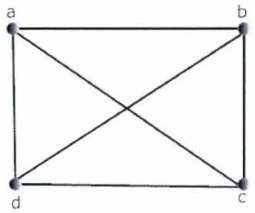

FIGURE 2.1: Topology at maximum power.

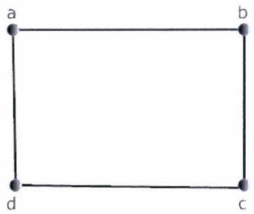

FIGURE 2.2: Topology produced by XTC in the ideal case.

FIGURE 2.3: Topology produced by XTC when nodes estimate distance incorrectly.

Let the distance-based neighborhood ordering $\prec$ for the ideal case (i.e.no error in distance estimation) be: $d \prec_a b \prec_a c$, $c \prec_b a \prec_b d$, $b \prec_c d \prec_c a$, and $a \prec_d c \prec_d b$. The topology after application of the XTC algorithm based on this ordering is shown in Figure 2.2. Let us now consider that nodes b and c estimate distances incorrectly and the distance-based ordering be: $d \prec_a b \prec_a c$, $c \prec_b d \prec_b a$, $b \prec_c a \prec_c d$, and $a \prec_d c \prec_d b$. When XTC algorithm is run based on this ordering, it gives the disconnected topology shown in Figure 2.3. Therefore errors in distance estimation makes XTC produce disconnected nodes.

Kevin and Sriran in [13] came up with two randomized protocols to avoid the disconnectivity property of XTC: RTC and ε-RTC. RTC uses randomized link labeling to produce a sparse topology. It consists of two phases. In the first phase, one of the two nodes u or v which has a higher ID picks a real number from the range $[0, 1]$ to serve as a link label (u, v). Each node makes a neighborhood ordering in increasing order based on the

link labels. In the second phase, XTC is executed with the neighborhood ordering as input.

ε-RTC uses distance estimates like XTC; however, to account for errors in distance estimation, ε-RTC first does a random perturbation of the distance estimates. It is assumed that every node knows the amount of perturbation ε. ε-RTC has two phases. In the first phase, every node u estimates the distance of its neighbors v. Due to errors, the distance estimated by u, $\delta(u, v)$ and the distance estimated by v, $\delta(v, u)$ can be different. The average of these distances is computed and a value is picked uniformly at random from the interval $[(1 - \varepsilon)\delta, (1 + \varepsilon)\delta]$. This value is assigned to both ends of the link (u, v) and this is used as the link label. Then every node makes a neighborhood ordering based on the link labels. In the second phase of ε-RTC, XTC is called with the neighborhood ordering as input.

The authors in [13] show that the topologies produced by RTC and ε-RTC are connected with bounded logical node degree. However, in the case of RTC, the resulting topology may not be efficient in terms of energy and network capacity because the link metric used is not related to any performance metric. In the case of ε-RTC, it is assumed that every node knows the amount of perturbation ε. However, it is not indicated how we can determine ε.

2.6 The LTCA Protocol

Another improvement towards the disconnectivity problem of XTC is the LTCA protocol presented in [14]. LTCA is a deterministic protocol which relies only on node ID and connectivity information to produce a sparse topology. Initially, every node sends broadcast beacons at maximum power. Up on receiving the beacon messages, every node broadcasts its ID and the IDs of the nodes at maximum power. Then every node u checks a pair of nodes (v, w) if they are in the range of each other at maximum power (this is possible because the IDs of both neighbors is known by node u). If they are not in the range of each other, u keeps both as its neighbors. If they are in the range of each other, if the ID of u is less than the IDs of both v and w, u removes none; otherwise, node u removes the node with the higher ID. This way the final topology is produced.

The authors in [14] show that the topology produced by LTCA is connected with bounded logical node degree. However, since the neighbors are selected based on node ID, and not based on the quality of the links, the resulting topology may have unbounded physical node degree and may not be efficient in terms of energy consumption or network capacity.

2.7 Summary

Various topology control protocols have been discussed with emphasis on neighbor-based techniques. We discussed one direction-based protocol, the CBTC protocol, one location-based protocol, the LMST protocol and five neighbor-based protocols, the K-Neigh protocol, the XTC protocol, the ε-RTC protocol, the RTC protocol and the LTCA protocol.

The neighbor based techniques are definitely of our interest, because our nodes are neither equipped with directional antennas (as in the case of CBTC) nor do we want to rely on availability of GPS (as in the case of LMST). The authors in the papers show the various properties of the resulting topologies produced after the application of the topology control protocol.

The properties of resulting network topology include node degree, connectivity and bidirectionality. Some authors also refer to node degree as *sparsity*. Lower node degree is a desirable property, which reduces interference and energy consumption. For instance, one of the nice features of the XTC protocol is its sparsity. The authors explicitly indicated that the number of links is in the order of the number of nodes. It is also shown that in all the protocols, the links in the topology produced are bidirectional. The properties of the topologies produced by the various protocols presented is summarized in Table 2.1.

It is not possible to compare the various protocols on the basis of performance parameters such as throughput, delay and power consumption because the authors in the papers did not measure performance with respect to such performance metrics.

The other issue which is not addressed by the protocols presented is mobility. Some recent papers, such as [20, 21] presented mobility aware algorithms; however, they require the availability of GPS receivers and they are not in the domain of our interest. However, the fact that all the protocols presented are local and distributed makes them suitable for adapting them to a mobility aware protocol.

To sum up, all the neighbor-based protocols are based on XTC algorithm (excepting K-Neigh). The disconnectivity and unbounded degree problems of the XTC protocol are solved in the ε-RTC protocol, by giving an error margin for distance estimates (assuming bounded error); the RTC protocol by using random edge labeling to produce a sparse topology; and the LTCA protocol, by using node IDs and connectivity information to produce a sparse topology. From the protocols presented, the LTCA protocol has many nice features including the fact that it is deterministic, is computationally simple, is local as well as distributed, guarantees connectivity, has bounded logical node degree and uses little assumptions. But it does not use some link quality metric to select links and may not produce a topology with the most efficient links in terms of energy or throughput.

Protocol	Connectivity	Node degree	Remark
CBTC	Guaranteed	Bounded	Requires directional antennas
LMST	Guaranteed	Bounded	Requires location information (GPS)
$k-$Neigh	Not guaranteed	Bounded (assuming correct distance estimate)	Uses distance estimation using RSSI
XTC	Connected (assuming correct distance estimate)	Not Guaranteed	Uses distance estimation using RSSI
ε-RTC	Connected	Bounded	Uses RSSI for distance estimation Gives margin for error in distance estimation, Assumes that the error margin is known
RTC	Connected	Bounded	Relies on randomization to build sparse topology Does not consider the quality of links
LTCA	Connected	Bounded	Relies on *id*s to build sparse topology Does not consider the quality of links

TABLE 2.1: Summary of the various topology control protocols

Chapter 3

Solution Direction

As pointed out in the previous chapter, neighbor-based topology control protocols are suitable for our problem at hand. The various desirable properties that a topology control protocol (for MANETs) should have include:

- It should rely on low-quality information which does not require additional hardware;
- It should have small number of message exchange;
- It should use realistic assumptions;
- It should be distributed as well as local;
- It should generate a topology with an upper bound on node degree, which is fundamental to maintain a relatively low level of interference in the network;
- It should preserve connectivity in mobility;
- It should generate a topology that contains only bidirectional links, which is essential for successful MAC layer operation.

The previous chapter introduces various neighbor-based protocols. Among the protocols presented, the LTCA protocol has many of the above properties. However, it does not use some link quality metric to select links and may not produce a topology with the most efficient links in terms of energy or throughput. All the protocols do not guarantee connectivity in mobility as they are based on the assumption that the network topology is static.

A power-based topology control solution has several steps including:

1. Neighbor discovery,
2. Link quality determination,
3. Neighbor information exchange,
4. Topology construction or link selection,

5. Transmit power adjustment.

This chapter explains the above steps from the practical point of view. We first discuss the methods of neighbor discovery and then the various link quality metrics are discussed. Then we discuss the link selection phase. We use the XTC algorithm as the link selection phase, with a simple remedy to avoid the disconnectivity property of XTC. Then we discuss about how we can practically vary the transmit power levels. Finally, we study how we handle mobility. To facilitate discussion, we first present definitions and assumptions.

Definitions:

The communication graph: The communication graph defines the network topology, that is, the set of wireless links that the nodes can use to communicate with each other. Let N be a set of wireless nodes located in a certain bounded region V, with $|N| = n$. A d-dimensional mobile ad hoc network is then represented by an undirected graph $M_d = (N, L)$, where $N = \{u_1, u_2, ..., u_n\}$ is the set of nodes in the network and L is the set of bidirectional links.

$1 - hop\ neighbor$: The directed wireless link (u, v) exists if and only if nodes u and v are at distance of at most $R(u)$ at time t, where $R(u)$ is the range assigned to u. In this case, v is said to be a $1 - hop\ neighbor$ of node u.

Bidirectional neighbors: A wireless link is said to be *bidirectional* at time t if $(u, v) \epsilon L$ and $(v, u) \epsilon L$. In this case, nodes u and v are said to be *bidirectional neighbors* and the link connecting the two neighbors is called *bidirectional link*. Many authors use the terms *symmetric* and *bidirectional* interchangeably; however, in this thesis, the term *symmetry* is used to refer the same received power by both nodes. More specifically, nodes u and v are *symmetric neighbors* if the received power by u from V is equal to the received power by v from u. The link between these two neighbors is referred to as *symmetric link*.

Assumptions: Since communication on links which are not bidirectional is not practical in wireless multihop networks [22], in this thesis, only bidirectional links are concerned. We also assume that, since we have a synchronization capability element in the MAC frame, it is logical to assume that all nodes which are in the reach of each other are time-synchronized.

3.1 Neighbor Discovery

A broadcast beacon message is used for neighbor discovery. The beacon frame is one of the ieee 802.11 MAC management frames. A typical beacon frame is approximately fifty bytes long, with about half of that being a common frame header and cyclic redundancy checking (CRC) field. As with other frames, the header includes source and destination MAC addresses as well as other information regarding the communications process. The destination address is always set to all ones, which is the broadcast Medium Access Control (MAC) address. This forces all other stations on the applicable channel to receive and process each beacon frame. The beacon's frame body resides between the header and the CRC field and constitutes the other half of the beacon frame. Each beacon frame carries, among others, the *Neighbor List* element, which is used by a node

to advertise its neighbor list and a *Timestamp Element* which a node uses to update its local clock (which enables synchronization among all nodes) in the frame body.

Initially every node sends beacon messages at maximum power along with the node ID and the network ID. Based on the network ID, a node decides whether or not it should associate with the broadcasting node. The node requests association using the credentials for that network. After successful authentication, every node stores the IDs of all the nodes from which it received the beacon. Beacon messages are also sent periodically to discover newly introduced neighbors.

3.2 Link Quality Determination

After nodes discover their neighbors, they should determine the qualities of the links to their neighbors. A link quality metric optimizes a certain performance metric. For example, power consumption is directly affected by the distance between two nodes. Distance is predicted by RSSI value. So distance between nodes (estimated from RSSI value) or the RSSI value itself can be considered as power-aware link quality metrics. This means that using distance between nodes or RSSI value as a link quality metric gives the best topology in terms of power consumption. The resulting topology is also expected to have a minimal transmission range under the constraint that the network is connected. Every node measures the RSSI value when it receives the beacon message.

On the other hand, the minimal transmission range topology might not give the best throughput. So we can also take throughput-aware link quality metrics. Such link quality metrics may not minimize the transmit power and hence may not give the most energy efficient links. However, using such link quality metrics has an advantage of maintaining the links which give best throughput and which would be removed if we used power-aware metrics. The metrics which can be used to predict the throughput of a link are:

- *Packet delivery ratio (PDR):* PDR from a node A to node B is the ratio of the received packets on node B divided by the transmitted packets on node A. The PDR is easily measured by generating measurement packets to neighboring nodes. These packets are generated at the higher layer; for example, the Neighbor Acknowledgment (NA) packets are generated by FLAME for the purpose of measuring the PDR [23].

- *Expected transmission time (ETT):* This is the expected time needed to transmit a packet over a link. This also takes the transmission rates and the packet loss of the links into account. The idea is that a high speed link with some packet loss might still be better than a low speed link without packet loss. It gives an indication of the time spent in transmitting the packet. The ETT (in microseconds) can be determined by the formula [23]:

  $ETT = \frac{packetsize}{PDR*rate}$,

  where PDR is the packet delivery ratio on a link, $rate$ is the rate of the link (in Mbits/sec), and $packetsize$ is the packet size (in bits).

  More details on using ETT as a link quality metric can be found in [23, 24].

Whether we use RSSI value or ETT to determine the link quality, we may get different link qualities measured by the two nodes for the same link, in which case the link is called an asymmetric link. The average of the two values of the link qualities is computed and this value is assigned as a common link quality for both nodes.

3.3 Neighbor Information Exchange and Link Selection

After neighbor discovery and link quality calculation, every node exchanges this information at maximum power (using the beacon frame). So at this step, every node u has the following minimal set of information about its neighbors:

- IDs of its neighbors v and the IDs of the neighbors of v,
- its link quality to every other node v and the link qualities of its neighbors v to every other neighbors of v,

Based on the above information, every node locally decides which neighbors it has to choose based on some rule. The link selection phase of the various protocols is discussed in Chapter 2. The link selection phase is as follows. In considering node v as its neighbor, node u takes another node w and sees if they are maximum transmission range of each other.

- If they are not in the transmission range of each other, u keeps both as its neighbors.
- If they are in the transmission range of each other and the quality of the link from u to w is greater than the quality of the link from u to v and the quality of the link from w to v is greater than the quality of the link from $u$to v, v is dropped from u's neighbor list.

This is similar to the link selection phase of XTC, the difference is the that we do not use neighbor ordering here and we use the same value (i.e. the average of the link quality values measured by the two nodes) for the link quality. This avoids the disconnectivity problem of the XTC protocol.

In the case of using ETT as a link quality metric, which is also a routing layer link quality metric, we want to keep the links (called essential links) selected in the form of a table so that it can be used by upper layer routing algorithm, which can be an advantage in reducing overhead. Through the construction of the topology, each node can construct a local table which is described as,

Next hop	Link quality

Each selected neighbor (in the new topology) has an entry in the table. The link quality represents the quality of the link connecting the current node and the next hop. It can be used by upper level routing algorithm to find the best link.

3.4 Transmit Power Adjustment

After a node selects its neighbors, the next step is the determination of the transmit power needed to send a message to any neighbor node. The mechanisms for transmit power information exchange in IEEE 802.11 wireless cards is described in [25]. Power information is exchanged using the TPC Report element which contains transmit power and link margin information sent in response to a TPC Request element. The TPC Report element is included in TPC Report frames, Beacon frames, and Probe Response frames. The Transmit Power field is set to the transmit power used to transmit the frame containing the TPC Report element.

We assume that each node knows its own threshold receiving power (or sensitivity), P_{th}. By measuring the received power, P_r of the beacon messages, which are sent at maximum power, and comparing it with P_{th}, a node can determine its link margin to every other neighbor.

The nodes exchanging power information (the transmit power level, which is P_{max} in this case because we send beacon messages at maximum power) and the link margins. So every node can know which neighbor is logically the farthest. The node with the smallest link margin is logically the farthest node. In the case that there are no obstacles between the nodes, that node is also physically the farthest. Every node also knows the path loss to every other node by the relation

$PL(\text{dB})=P_{max}(\text{dBm})\text{-}P_r(\text{dBm})$.

The transmit power level of node u is set to the power level to reach the farthest neighbor, so that the received power by the logically farthest neighbor is equal to its receiver sensitivity. This is equivalent to saying that the link margin to the logically farthest neighbor is set to 0. So the transmit power of node u is set to

$P(u)(\text{dBm})=P_{th}(\text{dBm})+PL(\text{dB})$.

Previous works on power based topology control assume that nodes can be assigned any transmit power level continuously between the minimum and the maximum power, which is impractical. So we have to take into account our practical degree of freedom to vary the transmit power. In some commercial IEEE 802.11 cards, transmit power level can be dynamically adjusted. For instance, the CISCO Aironet IEEE 802.11 $a/b/g$ card can use certain discrete transmit power levels ranging from 1 mW to 100 mW.[1] The nodes in the Figo network at WMC can change their transmit power from 1 mW to 63 mW in discrete levels.

Note that our goal is to adaptively adjust the transmit power of each node instead of using the maximum transmit power. So our algorithm should assign an available transmit power level by taking the next power level greater than or equal to the computed transmit power so that connectivity is maintained.

[1] http://www.cisco.com/en/US/prod/collateral/wireless/ps6442/ps4555/ps5818/product_data_sheet09186a00801ebc29.html

3.5 Accounting for Mobility

To account for mobility, an algorithm should not only be distributed but also local: Each node is allowed to exchange messages with its neighbors a few times and then must decide which links it wants to keep. Several neighbor-based protocols, such as K-Neigh, and XTC, are locally computed where each node only communicates with its neighbors twice.

In all the protocols, it is assumed that, for any given network, the topology control protocol is computed just once, thereby assigning powers to the nodes so that the MANET is continuously connected throughout the network lifetime. However, for mobile networks, since the nodes are mobile, the distance between nodes may change as nodes move. Most of the protocols presented did not incorporate what should be done to make the network continuously connected throughout the network lifetime. Some recent papers, such as [20, 21] presented mobility aware algorithms; however, they require the availability of GPS receivers and/or direction information to predict the future location of a node and they are not in the domain of our interest. The authors of [7] proposed, as their future research direction, that if the mobility is high, an on-demand approach to reconfigure the network topology may be used; and if mobility is low, proactive methods may be used.

The first approach falls in the category called reactive methods. A node is triggered, based on a certain condition, to run the topology control algorithm and establish a new set of links. The node can be triggered by the MAC layer or the routing layer.

The MAC layer can trigger re-execution of the topology control protocol in case it discovers new neighbor nodes. The MAC level can detect new neighbors by overhearing the network traffic and analyzing the message headers to discover new neighbors to ensure a quick response to changes in the network topology.

The routing layer can trigger the re-execution of the topology control protocol in case it detects many route breakages in the network, since this fact is probably indicative that the actual network topology has changed a lot since the last execution of topology control. On the other hand, the topology control protocol, which creates and maintains the list of the immediate neighbors of a node, can trigger re-execution of the routing protocol in case it detects that the neighbor list is considerably changed.

An advantage of this approach is that, since all the nodes do not need to run the algorithm, reconfiguration control traffic is kept low. Only the node which loses a link or a newly introduced node and/or a node which overhears the newly introduced node needs to re-execute the algorithm. So, all the other nodes do not need to go back to the maximum transmit range. This is advantageous in terms of power efficiency and reduced interference. An obvious disadvantage is that reactive methods are vulnerable to disconnectivity - that is the network connectivity at any time instant may not be guaranteed.

The second approach falls in the category called proactive methods. In order to maintain the connectivity of the network as a whole, rather than as a set of specific connections between pairs of nodes, we set a reasonable time interval and focus on maintaining network connectivity throughout each time interval. That is, to assign power levels to the nodes such that the network is connected throughout the prespecified interval of time. This can be incorporated, for example, by setting the transmit range equal to

the distance of the farthest node (at the time of computing the topology) plus the sum of the distances both nodes can move (taking the worst case that both nodes move in the opposite direction) to account for possible node movement. An advantage of this approach is that connectivity is maintained at any instant in time. The best topology is also constructed every time the algorithm is executed. A disadvantage is that we will have a high reconfiguration control traffic. The fact that all the nodes go to the maximum transmit range also implies higher power consumption and increased interference.

In both approaches, a node can increase or decrease its transmit power by detecting and following the farthest node at any instant until the entire topology control algorithm is executed. This can be done by continuously requesting power information (link margin) from all the neighbors and recalculating the transmit power based on that link margin. But this has a disadvantage of increased overhead.

3.6 Summary

In this chapter, we discussed the various steps of our topology control solution for MANETs from the practical point of view, including neighbor discovery, determination of link quality, neighbor information exchange and link selection and transmit power level adjustment. How to make the algorithm mobility adaptive is also studied.

For neighbor discovery, a broadcast beacon message at maximum power is used. The beacon is one of the ieee 802.11 MAC management frames. Beacon messages are also sent periodically to discover newly introduced neighbors.

The next phase is the determination of link quality. Distance between nodes and RSSI value are energy aware link quality metrics. Packet delivery ratio (PDR) and expected transmission time (ETT) are throughput-aware link quality metrics.

After neighbor discovery and link quality calculation, every node exchanges this information at maximum power (using the beacon frame). Based on the above information, every node locally decides which neighbors it has to choose based on some rule. This is called the link selection phase.

After a node selects its neighbors, the next step is the determination of the transmit power needed to send a message to any neighbor node.

Two approaches have been presented to make the algorithm adaptive to node mobility: reactive and proactive. In the reactive method, a node is triggered, based on a certain condition, to run the topology control algorithm and establish a new set of links. The node can be triggered by the MAC layer or the routing layer. Advantages of this approach is that, since all the nodes don't need to run the algorithm, reconfiguration control traffic is kept low. A disadvantage is that reactive methods are vulnerable to disconnectivity. In the proactive method, we set a reasonable time interval and the algorithm is re-executed at the beginning of each interval. An advantage of this approach is that connectivity is maintained at any instant in time. A disadvantage is that we will have a high reconfiguration control traffic.

In the coming chapters, we are going to investigate the effect of the various parameters such as the transmission range, the node degree and the message complexity on the

various performance metrics such as energy consumption, capacity and network delay by analysis and/or simulation.

Chapter 4

Analysis

The previous chapter explains the various steps of topology control for MANETs from the practical point of view. We also pointed out the various desirable properties that a topology control protocol should have including small message exchange, low degree, and connectedness. For a topology to have a low degree, the transmission range of a node should be kept small as long as the network stays connected.

In this chapter, we analyze the effect of the transmission range, node degree and the message complexity on the performance of mobile ad hoc network including energy consumption, capacity and delay. We also determine the reconfiguration interval of a topology control algorithm using statistical model. For analysis, we employ the IEEE 802.11 DCF MAC protocol. Hence, we start with brief overview of the IEEE 802.11 DCF.

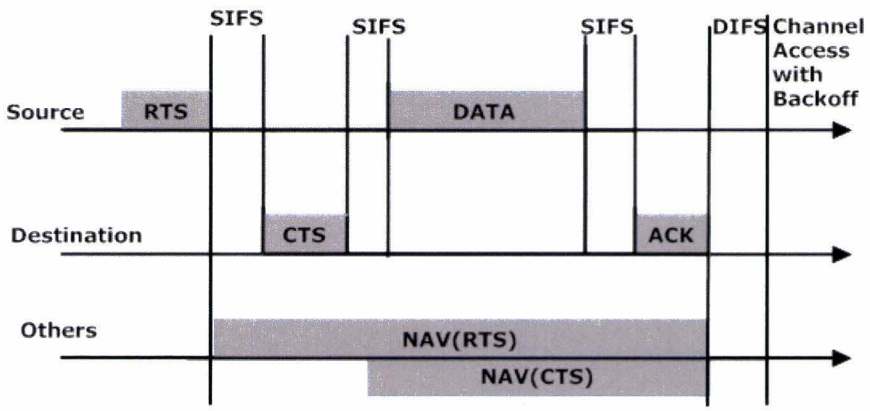

FIGURE 4.1: Four-way handshaking access mechanism.

4.1 The IEEE 802.11 MAC

Distributed Coordination Function (DCF) is the fundamental MAC technique of the IEEE 802.11 wireless LAN standard. DCF employs a CSMA/CA distributed algorithm and an optional virtual carrier sense using RTS and CTS control frames and a random

backoff time following a busy medium condition, or when a transmitting node infers a failed transmission. The four-way handshaking DCF is shown in Figure 4.1.

If a node wants to send data, it sends a Request-to-send (RTS) control packet. If the receiver receives the RTS, it answers with a Clear-to-send (CTS) control packet. During RTS/CTS handshaking, a sender and a receiver perform the virtual carrier-sense mechanism, which reserves the medium against interfering nodes. Specifically, an RTS and a CTS packet contain information about the time duration needed to finish delivering the data packet corresponding to the RTS/CTS packet. When a node receives an RTS or a CTS packet, the node updates the *Network Allocation Vector (NAV)* which indicates the expected duration of future traffic on the medium. When the sender receives the CTS packet, it sends the data. At last, the receiver responds with an Acknowledgment (ACK) packet.

The time interval between frames is called the *interframe spacing* (IFS). *Short interframe spacing* (SIFS) is the shortest of the IFSs. SIFS shall be used when nodes have seized the medium and need to keep it for the duration of the frame exchange sequence to be performed. Using the smallest gap between transmissions within the frame exchange sequence prevents other nodes, which are required to wait for the medium to be idle for a longer gap, from attempting to use the medium, thus giving priority to completion of the frame exchange sequence in progress. *DCF interframe spacing* (DIFS) denotes the time a node has to wait after the medium is determined to be idle.

A node desiring to initiate transfer of data invokes the CS mechanism to determine the busy/idle state of the medium. If the medium is busy, the node defers until the medium is determined to be idle. When the medium is determined to be idle, the node waits for a period of time equal to DIFS. After this DIFS medium idle time, the node then generates a random backoff period (using Equation 4.1) for an additional deferral time before transmitting, unless the backoff timer already contains a nonzero value, in which case the selection of a random number is not needed and not performed.

$$\text{Backoff Time} = \text{Random}() * \text{SlotTime}, \tag{4.1}$$

where:

- *Random()* = Pseudo-random integer drawn from a uniform distribution over the interval $[0, CW_i]$, where CW_i is an integer within the range of values of the PHY characteristics CW_{min} and $CW_{max}, CW_{min}, \leq CW_i \leq CW_{max}$.
- SlotTime = The value of the correspondingly named PHY characteristic.

The contention window (CW) parameter shall take an initial value of CW_{min}. Every node maintains a *short retry count* (SRC) as well as a *long retry count* (LRC), both of which shall take an initial value of zero. The CW shall take the next value in the series every time there is an unsuccessful attempt to transmit, until the CW reaches the value of CW_{max}. In the case of exponential back-off procedure, CW_i takes values $2^i CW_{min}$, for $0 \leq i < m$ and $2^m CW_{min}$, for $m \leq i \leq SRC$. A retry is defined as the entire sequence of frames sent, separated by SIFS intervals, in an attempt to deliver data. Once it reaches CW_{max}, the CW shall remain at the value of CW_{max} until the CW is reset.

A node performing the back-off procedure shall use the CS mechanism to determine whether there is activity during each backoff slot. If no medium activity is indicated for the duration of a particular backoff slot, then the backoff procedure shall decrement its backoff time by SlotTime. If the medium is determined to be busy at any time during a backoff slot, then the backoff procedure is suspended; that is, the backoff timer shall not decrement for that slot. If the medium is determined to be idle for the duration of a DIFS period, the backoff procedure is allowed to resume. Transmission shall commence when the Backoff Timer reaches zero.

4.2 Modeling a Node as a Queue

In this section, we introduce a queuing model for a node in a network. This model is used for analysis of energy consumption and network delay. We consider that N nodes are initially distributed uniformly at random over a deployment area of sides a by b. Each node generates traffic at a rate of λ packets per second on average with a characteristic of i.i.d. Poisson process [26]. Every node can be a source, a destination and/or a relay node. When a packet is received by a node, the node determines that it is the destination of the packet at reception time with probability of p_d. A node forwards a packet to every neighbor with an equal probability. This implies that a relaying node spreads a forwarding packet to every node within its transmission range.

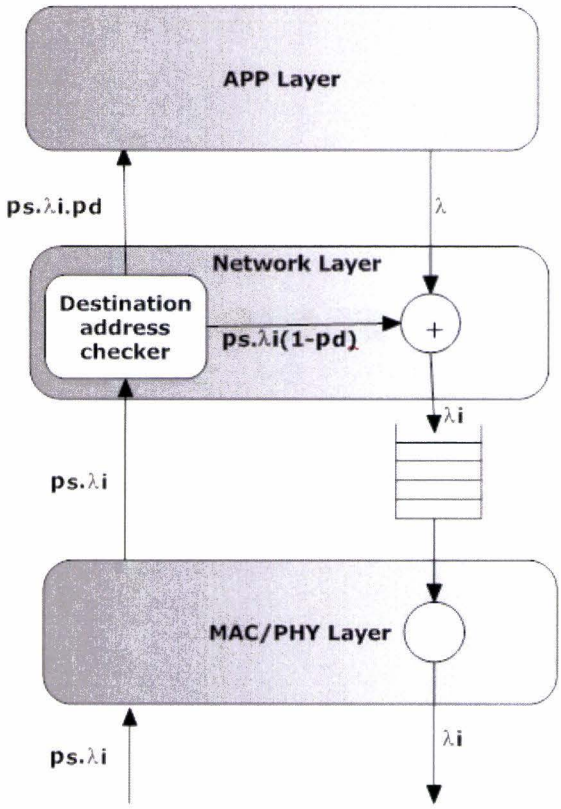

FIGURE 4.2: Queuing model for a node.

The message queuing within a node, which is used in [27] is shown in Figure 4.2. The node i has a radio range of R, and every node in the system is symmetric in traffic; each node generates the same amount of traffic and forwards an equal number of packets from the same number of neighbor nodes, denoted as λ_i. However, packet delivery over the wireless channel may fail (eg. due to collision). p_s denotes the probability of successful delivery. An application layer generates messages at the rate of λ, and the messages are passed to the network layer. Before the messages are transmitted from the node to the next hop, they are stored in a message buffer located between the network layer and the MAC/PHY layer. Node i also receives messages from neighbor nodes at the rate of $p_s\lambda_i$, and the messages are passed to the network layer for routing decisions. Packets destined to the node are dispatched to the application layer, and the remaining packets are enqueued to be forwarded to their next hop. From the system model, an application layer accepts packets at the rate of $p_s\lambda_i p_d$ on average, and packets which should be forwarded to other nodes are appended to the message buffer at the rate of $p_s\lambda_i(1-p_d)$ on average. Accordingly, the message buffer has two sources of packet arrivals (i.e., the application layer and packet forwarding). Therefore, since the node is modeled as an M/M/1 queuing system, the buffer maps to a queue, and the MAC/PHY layer to a server of the queue.

4.3 Analysis of Energy Consumption

The network interface has four possible energy consumption states: *transmit*, *receive*, *idle* and *sleep*. *transmit* and *receive* are for transmitting and receiving a message. In the *idle* mode, the interface can transmit or receive. This is the default mode for ad hoc environment. The sleep mode has extremely low power consumption. The interface can neither transmit nor receive until it is woken up.

Energy consumption of a node is affected by the following factors:

- Message complexity: In [28], it is shown that the energy consumed by the network interface when a node sends, receives or discards a packet at maximum power can be described using a linear equation:

$$\text{energy cost} = m * \text{size} + b, \tag{4.2}$$

  where b is a fixed component associated with device state changes and channel acquisition overhead and $m*$size is an incremental component which is proportional to the size of the packet. Experimental results are used to determine values for the linear coefficients m and b for various operations.

- Sparcity: One major factor contributing to the extra energy consumption in IEEE 802.11 cards is contention [29]. Contention increases when the number of neighbors of a node increases. Experimental results show that, for wireless mesh networks, the power consumption generally grows linearly with the number of connections [30]. A lower degree then consequently leads to lower power consumption.

- Transmission range: For communication systems that can individually adjust the transmit power, energy is saved by reducing the transmit power to the level that is just needed to reach the worst selected neighbor and not the whole neighborhood anymore.

In this section, we analyze the energy consumption of IEEE 802.11 wireless card using the 802.11 DCF scheme. Our analysis is based on [29]. Referring to the IEEE 802.11 DCF scheme, the total energy consumed by a certain node to transmit a packet has four parts: the energy consumed during back-off, E_b, the energy consumed during freezing or overhearing other nodes transmitting, E_f, the energy consumed during collision, E_c and the energy consumed during successful transmission, E_s. It can be assumed that the node stays in the receiving mode if it doesn't transmit.

Consider n contending nodes, where $(n-1)$ is the node degree. We assume that every node has a packet to be transmitted. Similar to [29], we assume that for each time a node transmits a packet, the collision probability with other nodes, denoted as p, is constant.

When node u tries to send a packet, the packet may collide with packets from other nodes, or transmission may fail due to transmission errors or due to mobility, in which case retransmission is scheduled. Similar to [29], we assume ideal channel conditions and that mobility is low compared to the propagation delay so that retransmission is only due to collisions. For each node, the probability that a node successfully transmits a packet after i failures is given by:

$$p_i = p^i(1-p), \tag{4.3}$$

where p is the probability that any one of the $(n-1)$ nodes transmit other than node u. This is also equal to the collision probability. p can be written as:

$$p = 1-(1-\tau)^{n-1}, \tag{4.4}$$

where τ is the probability that a station transmits in a randomly chosen slot time. τ is given by [31]:

$$\tau = \frac{2(1-2p)}{(CW_{min}+1)(CW_{min}+1)+pCW_{min}(1-(1-(2p)^m))}; \tag{4.5}$$

where CW_{min} is the minimum size of the contention window (CW) and m is the maximum stage of the exponential back-off procedure. p and τ are solved simultaneously by numerical methods.

The back-off time can be determined from the random access MAC model. From the IEEE 802.11 specification, the contention window size increases exponentially depending on the back-off stage, i, and the window size is given by:

$$CW_i = 2^i CW_{min}, \tag{4.6}$$

until it reaches the maximum limit CW_{max} for $i = m$. So the average back-off timer value B for four-way handshaking is[1]:

$$B = (\sum_{i=0}^{m} \frac{CW_i}{2} p_i + \sum_{i=m+1}^{SRC} \frac{CW_m}{2} p_i) * \text{SlotTime}. \tag{4.7}$$

[1] In [29], CW is taken from the interval $[0, CW_i - 1$, and the average back-off timer value B for four-way handshaking is given by: $B = (\sum_{i=0}^{m} \frac{CW_i-1}{2} p_i + \sum_{i=m+1}^{SRC} \frac{CW_m-1}{2} p_i) * \text{SlotTime}$. However,referring to [32], CW is chosen from the interval $[0, CW_i]$, and, hence the average back-off timer should be given by Eq. 4.7.

Let N_c be a random variable representing the number of collisions before a packet is successfully transmitted. The average value of N_c is given by the mean of the geometric distribution:

$$E[N_c] \approx \frac{p}{1-p}. \tag{4.8}$$

The node will back-off for $N_c + 1$ times before successful transmission. Hence, the time the node spends in back-off state, T_b is given by:

$$T_b = (N_c + 1) * B. \tag{4.9}$$

Let P_{rcons} refer to the instantaneous power consumption when the node is in the receiving mode. So, E_b is given by::

$$E_b = T_b * P_{rcons}. \tag{4.10}$$

When collision happens (assuming collision happens at RTS/CTS stage), the sending node should wait for a timeout interval of $T_{CTS} + 2T_{SIFS} + 2\gamma$, where γ is the propagation delay. If we denote the instantaneous power consumption of a node when it is transmitting by P_{tcons}, E_c can be written as:

$$E_c = N_c(P_{tcons}T_{RTS} + P_{rcons}(T_{DIFS} + 2T_{SIFS} + T_{CTS} + 2\gamma)). \tag{4.11}$$

To find E_f, we used a different approach from the approach used in [29] because the approach they used to determine the number of overheard transmissions by node u during its back-off is not clear. Since the back-off timer in a node is frozen whenever its interfering neighbors start to transmit, n_t represents the average number of nodes which are ready to transmit among the interfering neighbors of node i. n_t can be expressed as:

$$n_t = \rho n, \tag{4.12}$$

where n is the average number of interfering nodes and ρ is the utilization factor (i.e. the amount of time that the MAC works in transmitting a packet). ρ is given by:

$$\rho = \frac{\lambda_i}{\mu_i}, \tag{4.13}$$

where λ_i is the packet arrival rate, μ_i is the packet service rate. Referring to the queueing model introduced in Section 4.2, at steady state, the average packet arrival rate at node i is:

$$\lambda_i = p_s\lambda_i(1 - p_d) + \lambda. \tag{4.14}$$

Then the packet arrival rate λ_i is given by:

$$\lambda_i = \frac{\lambda}{1 - p_s + p_s p_d}, \tag{4.15}$$

where p_s and p_d are introduced in Section 4.2. p_s is given by (considering all the cases before the packet is finally discarded and assuming that packet delivery fails only due to collision):

$$p_s = 1 - p^{SRC}, \tag{4.16}$$

where p is given by Eq. 4.4.

To find p_d, let N_h be the average number of hops which can be expressed as:

$$E[N_h] = \sum_{k=1}^{\infty} k(1-p_d)^{k-1}p_d = \frac{1}{p_d}. \tag{4.17}$$

For high density network, the path from the source to the destination is almost a straight line [33]. Hence, the average hop count can be written as:

$$E[N_h] \approx \frac{E[D_{ds}]}{r}, \tag{4.18}$$

where $E[D_{ds}]$ is the average distance between an arbitrary source-destination pair. Furthermore, in a rectangular area of sides a, b, $E[D_{ds}]$ is given by [34]:

$$E[D_{ds}] = \frac{1}{15}[\frac{a^3}{b^2}+\frac{b^3}{a^2}+\sqrt{a^2+b^2}(3-\frac{a^2}{b^2}-\frac{b^2}{a^2})]+\frac{1}{6}[\frac{b^2}{a}\arccos\frac{\sqrt{a^2+b^2}}{b}+\frac{a^2}{b}\arccos\frac{\sqrt{a^2+b^2}}{a}]. \tag{4.19}$$

From Eqs. 4.17, 4.18 and 4.19, we can determine p_d:

$$p_d \approx \frac{r}{E[D_{ds}]}. \tag{4.20}$$

The packet service rate is equal to the reciprocal of the packet service time, T_i:

$$\mu_i = \frac{1}{T_i}. \tag{4.21}$$

To find the packet service time, we use the IEEE 802.11 DCF. The average service time for a packet in a node can be expressed as:

$$T_i = \sum_{k=0}^{SRC-1} p^k(1-p)(kT_f + T_s) + p^{SRC}(SRC)T_f, \tag{4.22}$$

where T_s is the expected time spent in a packet delivery if successfully delivered and T_f is the expected time in delivery failure. The first term corresponds to the case in which the packet delivery succeeds in the k^{th} try, $(k \leq SRC)$, and the second term corresponds to the case in which the packet delivery finally fails and the packet is dropped.

Let T_t be the duration of successful transmission and T_c be the time consumed when a collision happens. T_t and T_c are given, respectively, by:

$$T_t = T_{DIFS} + T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK} + 4T_{SIFS} + 4\gamma, \tag{4.23}$$

$$T_c = T_{DIFS} + T_{RTS} + T_{CTS} + 2T_{SIFS} + 2\gamma. \tag{4.24}$$

Then T_s is then given by:

$$T_s = B + n_t T_{freeze} + T_t, \tag{4.25}$$

where B is the duration of the back-off timer and is given by Eq. 4.7, T_{freeze} is the amount of time for which the back-off timer is frozen due to transmissions from interfering nodes and T_t is the time required to send the data packet and is given by Eq. 4.23.

Let p_o be the probability that any of the $(n-1)$ nodes other than node u successfully transmits a packet. This happens when any one of the $(n-1)$ nodes are transmitting

and the other $(n-2)$ nodes are not transmitting, conditioned on p:

$$p_o = \frac{(n-1)\tau(1-\tau)^{n-2}}{p}. \tag{4.26}$$

Since an interfering node can succeed to send data with probability of p_o and can fail with probability $1-p_o$ with the timeout mechanism, and based on the assumption that packet delivery fails in the phase of RTS/CTS handshaking, T_{freeze} is expressed as:

$$T_{freeze} = p_o T_t + (1-p_o)T_c, \tag{4.27}$$

where T_t and T_c are given by Eq. 4.23 and Eq. 4.24, respectively.

Similarly,we can derive expression for T_f.

$$T_f = B + n_t T_{freeze} + T_c. \tag{4.28}$$

Thus E_f is given by:

$$E_f = n_t P_{rcons}[p_o T_t + (1-p_o)T_c], \tag{4.29}$$

where T_t and T_c are given by Eq. 4.23 and Eq. 4.24, respectively.

Finally, the energy spent by node u to successfully transmit a packet E_s is given by:

$$E_s = P_{tcons}(T_{RTS} + T_{DATA}) + P_{rcons}(T_t - T_{RTS} - T_{DATA}). \tag{4.30}$$

Thus, the total energy consumed by node u to transmit a packet is:

$$E = E_b + E_f + E_c + E_s. \tag{4.31}$$

The parameters we used for numerical analysis are given in Table 4.1. The power consumed during the *transmit* and *receive* modes were measured at WMC for the FIGO node [35]. The power consumed during the *transmit* mode was measured for the different transmit power levels (i.e. RF output power levels) set in the FIGO node. Figure 4.3 shows the results. The figure shows that the measured consumed power by the wireless card decreases when the transmit power level decreases. The power consumed during the receiving mode was also measured to be 1122 mW.

Figure 4.4 shows the energy consumed per successfully transmitted packet versus network density for varying transmit power level. The figure shows that, when the transmit power level increases, the energy consumed significantly increases. Increasing the transmit power level not only consumes high power, but also increases the transmission range (which also increases the node degree) and the MAC layer procedure takes nodes a longer time to successfully transmit the packet. So, decreasing the transmit power level is crucial for low energy consumption with the constraint that the network should be connected.

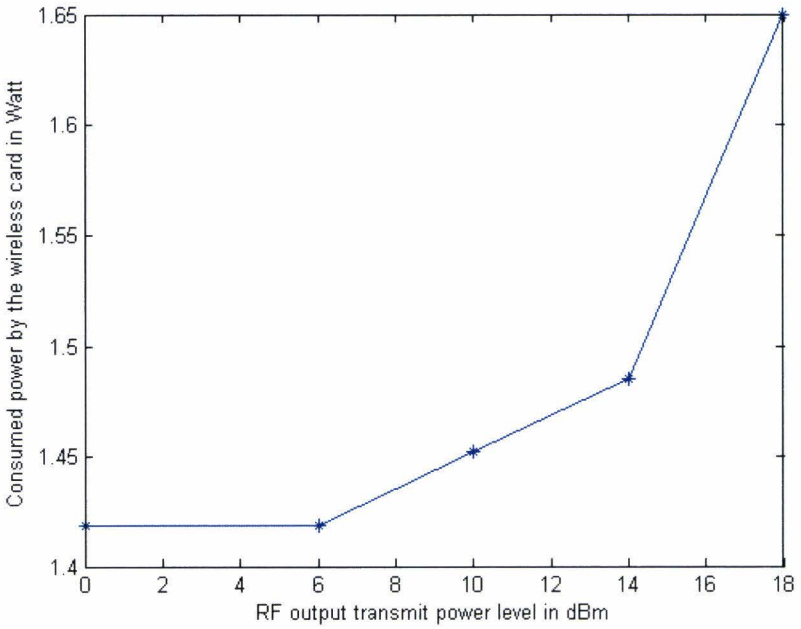

FIGURE 4.3: The measured power consumption by the wireless card vs. RF output power level.

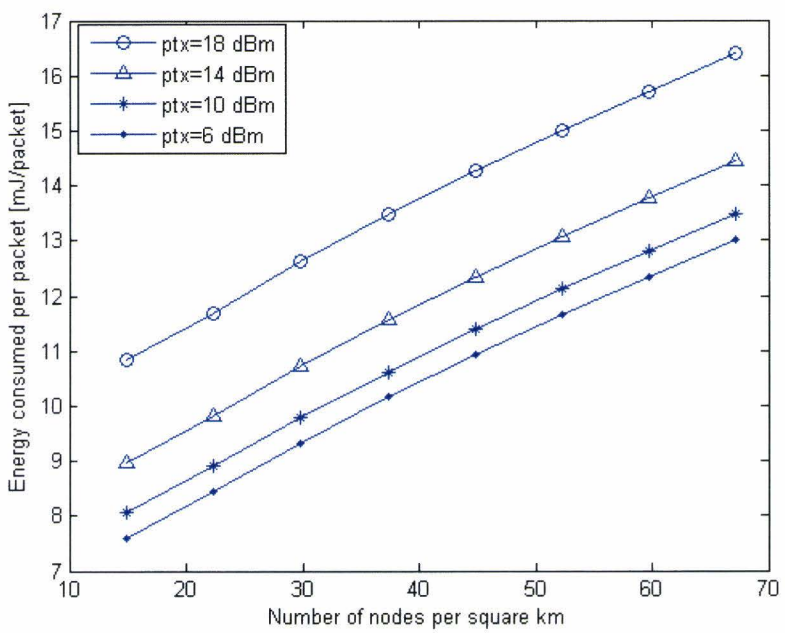

FIGURE 4.4: The energy consumed per successfully transmitted packet vs. network density for varying transmit power level p_{tx}.

m	5
SRC	7
CW_{min}	32
Channel bit rate	10^6 bps
RTS	44 bytes
CTS	38 bytes
ACK	38 bytes
Slot Time	$50\mu s$
T_{DIFS}	$50\mu s$
T_{SIFS}	$10\mu s$
γ	$2\mu s$

TABLE 4.1: Parameters used in numerical analysis.

4.4 Capacity Analysis

In [36], capacity is defined as the maximum possible information transfer rate over a channel. The network capacity depends on the achievable channel capacity at each individual wireless link and the level of spatial reuse - the total number of concurrent transmissions that can be accommodated in the network. The channel capacity at each individual wireless link depends on the SINR at the receiver. The spacial reuse can be increased by decreasing the transmit power. However, when we decrease the transmit power, the SINR decreases as a result of the smaller received signal. So, studying the effect of varying the transmit power and its effect on the SINR at the receiver is important for capacity analysis.

A formula for $E[C/I]$ is derived in [36]. They proposed the *Honey-grid* model to calculate the interference experienced by a node in ad hoc networks where nodes are uniformly distributed over a two-dimensional area larger than the coverage area of a node. Figure 4.5 shows the Honey-grid model. When a node, say node 0, is transmitting, there will be no interference from other nodes inside the coverage area of node 0 (due to MAC layer restrictions). In the worst case situation, the first set of interfering signals will come from signals transmitted from nodes just outside the coverage area of node 0 (at distance $R + \epsilon$ to node 0) for small ϵ, where R is the transmission range. In Figure 4.5, nodes are placed in co-centered hexagons. The first hexagon has a side of size Δ and contains six nodes. Δ depends on the network density. The i^{th} ring has a side of size $i\Delta$ and contains $6i$ nodes. If the total number of nodes in the network is N, the maximum reach, k, and N are related as:

$$N = 1 + \sum_{j=1}^{k} 6j = 1 + 3k(k+1). \tag{4.32}$$

With uniform distribution of nodes, each node has n other nodes inside its coverage area (except for nodes at the borders of the network). n is called the *node degree* and is given by:

$$n = \sum_{j=1}^{a} 6j = 3a(a+1), \tag{4.33}$$

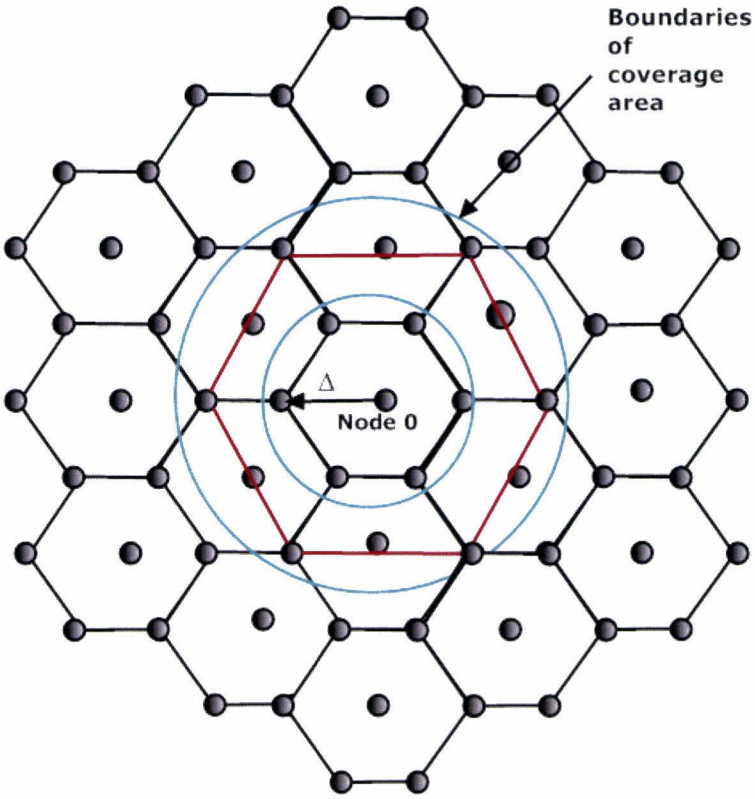

FIGURE 4.5: The Honey-grid model for modeling interference.

where a is the number of hexagonal rings inside the range of a node. Therefore, a gives indication of the transmission range of a node. Around node 0, the first set of interfering signals will come from signals that are transmitted from nodes just outside the coverage area of node 0. On the assumption that an entire ring is either included or excluded from the coverage area, the first ring of interference consists of 6 nodes positioned at distance $(a+1)R$ to node 0. The number of co-centered interference rings seen from node 0 is $k/(a+1)$, and the number of interfering nodes is given by:

$$N_i = \sum_{j=1}^{\frac{k}{a+1}} 6j = 3\frac{k}{a+1}(\frac{k}{a+1}+1). \tag{4.34}$$

To calculate the amount of interference experienced at node 0, we add the interference power received at node 0 from all interfering nodes. The j^{th} interference ring contains $6j$ nodes at approximated distance $j(a+1)\Delta$ to node 0. Using the path-loss power law model for radio propagation, the mean value of the received signal power is given by $P_{rx} = cd^{-\alpha}$, where P_{rx} is in Watts), d is the distance between the transmitter and the receiver, c is a constant that depends on transmitted power, the receiver and the transmitter antenna gains and the wavelength and α is the path loss exponent.

The mean power of interfering signals originating from ring j is $6jqc(j(a+1)\Delta)^{-\alpha}$, where q is the probability of transmission (transmission of own signals or relay signals) per node in a given slot time. q depends on the mean of the total traffic arriving at a node, Λ. The number of packets arriving per unit time is Λ/t_{ts}. Since we assumed

Poisson arrivals, the probability of k arrivals in a given time interval t is given by:

$$p_k(t) = \frac{(\frac{\Lambda}{t_{ts}}t)^k}{k!} e^{\frac{-\Lambda}{t_{ts}}t}. \tag{4.35}$$

So q can be written as:

$$q = 1 - p_0[t_{ts}] = 1 - e^{-\Lambda}. \tag{4.36}$$

The total traffic arriving at a node consists of its own traffic and the traffic that the node relays for other nodes. Consider any two nodes i and j. When the average hop count is $E[h]$, there are $(E[h]-1)$ relay nodes between any source and destination. Node i may be a relay node for node j with a probability of $(E[h]-1)/(N-1)$. The expected value of the traffic arriving at node i from node j is then $\lambda t_{ts}(E[h]-1)/(N-1)$. Any node in the network can be a relay for $(N-1)$ other nodes. Hence the expected value of the traffic arriving at any node is $\lambda t_{ts}(E[h]-1)$. The average total traffic per node, Λ is the sum of its own traffic, λt_{ts} and all relay traffic that reach that node:

$$\Lambda = \lambda t_{ts} + \lambda t_{ts}(E[h]-1) = \lambda t_{ts} E[h], \tag{4.37}$$

where $E(h)$ is the average hop count and λt_{ts} is the nodes own traffic. A formula for the average hop count is derived in [36] from the hop distribution in the Honey-grid model:

$$E[h] = 0.53 N_r^{0.5} + 2(1 - \frac{N_r}{N}), \tag{4.38}$$

where N_r is the number of relay nodes. The number of relay rings as seen by node 0 is k/a (assuming minimum hop routing). Hence, the number of relay nodes including the source node is given by:

$$N_r = 1 + \sum_{j=1}^{\frac{k}{a}} 6j = 1 + 3\frac{k}{a}(\frac{k}{a} + 1). \tag{4.39}$$

. The total amount of interference mean power is then:

$$I = 6qc((a+1)\Delta)^{-\alpha} \sum_{j=1}^{\frac{k}{a+1}} j^{-(\alpha-1)}, \tag{4.40}$$

In the Honey-grid model the lowest expected value for wanted signal power, C, is related to the situation that the wanted signal (signal from the source) is transmitted from the farthest neighbor of node 0 at distance $a\Delta$. The highest value of C is related to the situation that wanted signal is transmitted from the nearest neighbor of node 0, which is at distance Δ. The expected value for C is found then by taking into account all possible positions of the wanted signal transmitter:

$$E[C] = \frac{1}{n} \sum_{j=1}^{a} 6jc(j\Delta)^{-\alpha} = \frac{6c\Delta^{-\alpha}}{n} \sum_{j=1}^{a} j^{-(\alpha-1)}. \tag{4.41}$$

The expected value of C/I for a node in the center of an ad-hoc network is then given by:

$$E[C/I] = \frac{\frac{6c\Delta^{-\alpha}}{n}\sum_{j=1}^{a} j^{-(\alpha-1)}}{6qc((a+1)\Delta)^{-\alpha}\sum_{j=1}^{\frac{k}{a+1}} j^{-(\alpha-1)}} = \frac{\sum_{j=1}^{a} j^{-(\alpha-1)}}{nq(a+1)^{-\alpha}\sum_{j=1}^{\frac{k}{a+1}} j^{-(\alpha-1)}}. \tag{4.42}$$

An upper bound on the reliable data transmission speed between two nodes over the radio channel (with additive white Gaussian noise (AWGN) channel model) can be expressed by the Shannon channel capacity formula:

$$W = B\log_2(1 + E[C/I]), \tag{4.43}$$

where W (bits per second) is the upper bound on the time-averaged error free bit transmission speed over the radio channel, B is the channel bandwidth in Hz and $E[C/I]$ is the expected carrier to interference ratio. In other words, W is the maximum capacity of the wireless channel. When the expected value of C/I decreases, the capacity of the link between two nodes calculated with the Shannon formula decreases as well. An additional restriction on capacity is imposed by the MAC protocol. At any moment in time only one of the neighboring nodes may transmit. With node degree n, the channel capacity needs to be divided by $n+1$ to obtain the capacity, W, per node:

$$W = \frac{B}{n+1}\log_2(1 + E[C/I]), \tag{4.44}$$

where $E[C/I]$ is the average value of the signal to interference ratio at that node.

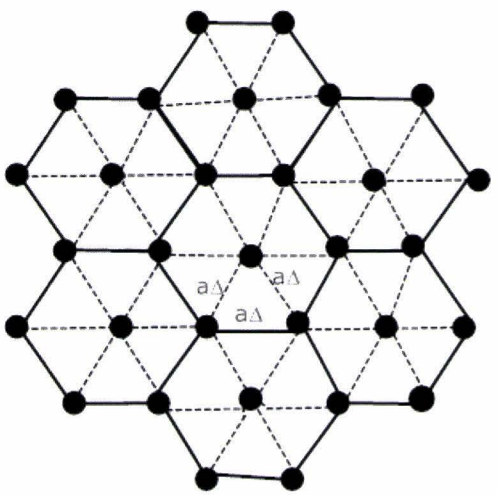

FIGURE 4.6: The consumed area for communication in the Honey-grid model.

The number of concurrent transmissions which are allowed under physical carrier sense in an area of A can be determined. The transmitters that can transmit concurrently will be positioned as shown in Figure 4.6. As each three transmitters shares a regular triangular with side length of $a\Delta$ and every transmitter is the vertex of six such triangles, each transmitter consumes an area of $A_o = \frac{\sqrt{3}}{2}(a\Delta)^2$. The network capacity can then

be expressed as:

$$W = \frac{A}{A_o}\frac{B}{n+1}\log_2(1 + E[C/I]), \tag{4.45}$$

where A is the total network area and $\frac{A}{A_o}$ is the total number of concurrent transmissions under physical carrier sense.

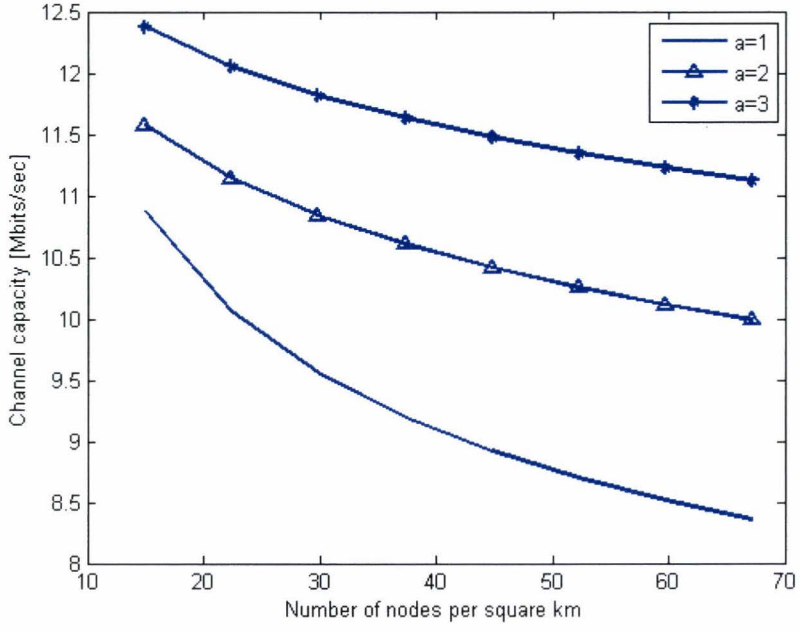

FIGURE 4.7: Channel capacity per node.

Figure 4.7 shows the channel capacity per node versus the density of nodes for different values of a. From the figure, we can see that as the number of nodes per square km increases, the channel capacity decreases. This is because, as the network density increases, the node degree n increases and the available channel capacity is shared among $(n + 1)$ nodes. Figure 4.7 also shows that, as the maximum reach of a node (a), which gives an indication of the transmission range of a node, increases for a given Δ, the channel capacity per node increases. This is because, if we have higher transmission range, which also means we have high transmit power, we have high expected value of the wanted signal, $E[c]$. So, $E[C/I]$ increases as a increases, which means as the transmission power increases. So the per channel capacity increases.

4.5 Delay Analysis

The delay of a packet in a network is the time it takes the packet to reach the destination after it leaves the source. We do not take delay at the source into account, since our interest is in the network delay. In this section we analyze the packet delay in mobile ad hod networks. This delay is the sum of the delays on each link traversed by the packet. Each link delay in turn consists of four components [26]:

- The *processing delay* between the time the packet is correctly received at node and the time the packet is assigned to an outgoing link queue for transmission.

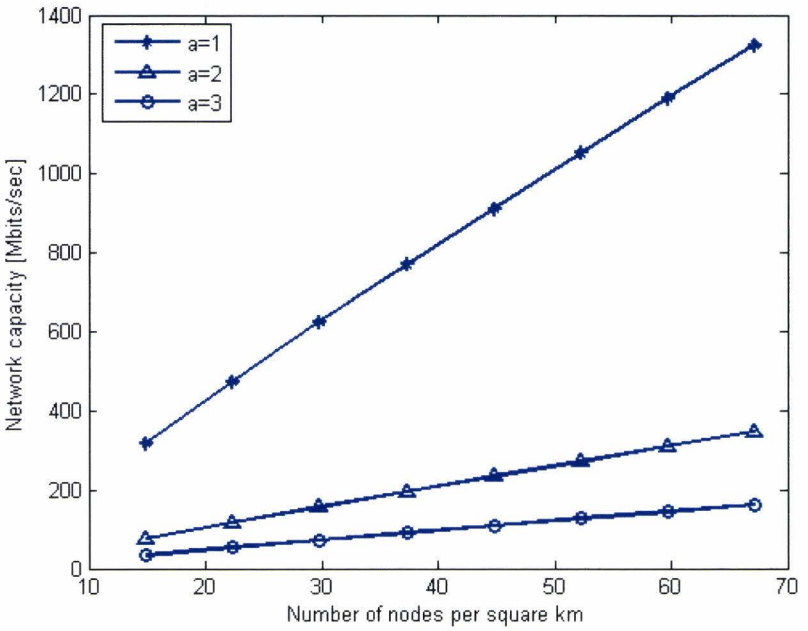

FIGURE 4.8: Network capacity.

- The *queuing delay*- the time the packet is assigned to a queue for transmission and the time it starts being transmitted.
- The *transmission delay* between the times that the first and last bits of the packet are transmitted.
- The *propagation delay* from the time the last bit is transmitted at the transmitted node of the link until the time it is received at the received node.

The authors in [27] use the queuing model as shown in Figure 4.2 to study the packet delay in MANETs, namely, M/M/1 queue. We also use their model for delay analysis, but we do not take mobility into account since in our case we consider pedestrian nodes and the propagation time is negligible compared to the time a destination node takes to go out of the transmission range of the source node. The processing delay is not considered.

Average delay in each node in M/M/1 queue is given by:

$$E[t] = \frac{1}{\mu_i - \lambda_i}, \tag{4.46}$$

where λ_i is the packet arrival rate and μ_i is the packet service rate. To find the packet arrival rate and the packet service rate, we follow the approach used in [27].

The average delay in each node can be found by putting the values of λ_i and μ_i in Eq. 4.46. This delay consists of the delay in the queue plus the service time. To find the average network delay, $E[D]$, we multiply the node delay by the number of hops, N_h:

$$E[D] = E[N_h]\frac{1}{\mu_i - \lambda_i}. \tag{4.47}$$

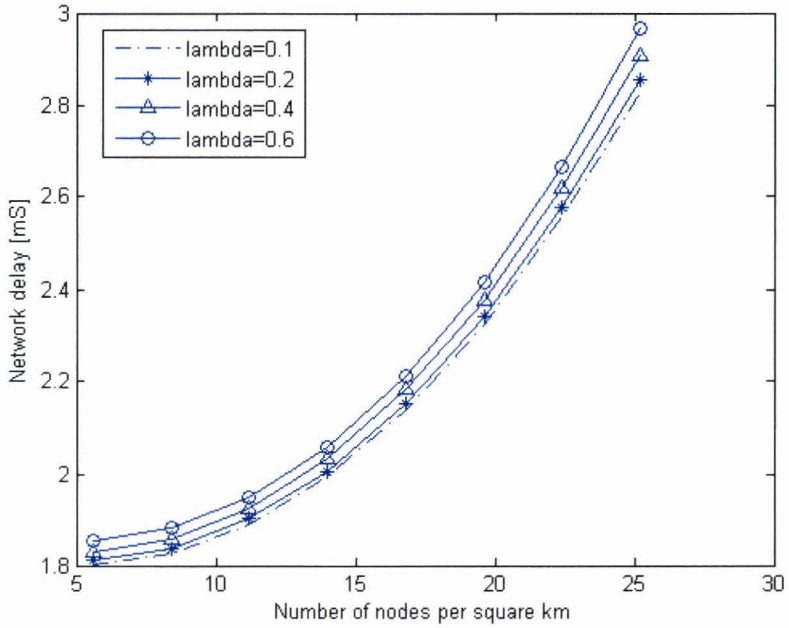

FIGURE 4.9: Numerical results for network delay vs. network density for varying lambda.

Figure 4.9 shows the numerical result for the average network delay versus network density for different packet generation rates at maximum transmission range. From the figure, we can see that as the number of nodes per square km increases, the network delay increases. This is because, as the number of nodes per square km increases, the node degree increases. As the node degree increases, the time the node spends in the MAC layer procedure also increases. This consequently leads to increase in average network delay. Numerically, we found that as the network density becomes a very high value, the average network delay becomes very high. We can also see that, when the packet generation rate λ increases, the network delay increases. However, the increase in the network delay is not significant.

Figure 4.10 shows the average network delay vs network density for varying transmission ranges for same packet generation rate ($\lambda = 0.5$). From the figure, we can see that decreasing transmission range increase network delay. Specifically, we can see that halving the transmission range almost doubles the network delay. This is because, as the transmission range is decreased, the number of links traversed by a packet is also increased and this increases the network delay.

4.6 Reconfiguration Interval

Two approaches for mobility adaption have been presented in Section 3.5; namely, reactive and proactive. In this section, we analyze the proactive approach. In the presence of node mobility, there is a clear trade-off between the message overhead generated by the repeated execution of a TC protocol and the quality of the constructed topology: the more frequently the protocol is re-executed (i.e. the higher the message overhead), the

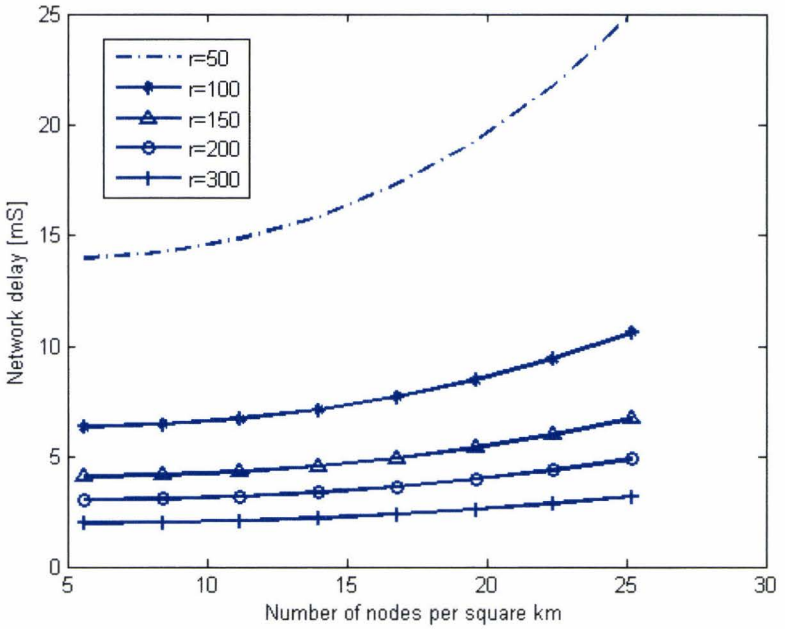

FIGURE 4.10: Numerical results for network delay vs. network density for varying transmission range.

higher the quality of the constructed topology (e.g. a topology that preserves connectivity). In [8], the interval between two broadcast reconfiguration messages is determined using a probabilistic model and we use the same approach to calculate the reconfiguration interval.

Let us assume that all nodes are randomly distributed within a disk of area A_0 and the total number of nodes is N and for a short time interval of length t, each node moves independently toward a random direction in $[0, 2\pi]$, with a constant speed v. Under these assumptions, we can calculate the probability that an existing neighbor moves into or out of the transmission range of node u, within a time interval of t.

Suppose node u is located in position A, with its neighbor v in position B as shown in Figure 4.11. The maximum transmission range of node u is r, and the distance between nodes u and v is $x(> r)$. Let $d = v * t$ be the distance that node v moves in time interval, t. We assume that d is less than the transmission range, r, of node u. The probability that node v moves into u's transmission range can be is given by:

$$p_{join} = \int_{r}^{r+d} \frac{2\pi x}{S_0} \frac{A(S_1)}{\pi d^2} dx, \tag{4.48}$$

where

$$A(S1) = \arccos(\frac{x^2 + r^2 - d^2}{2xr}) + \arccos(\frac{x^2 + d^2 - r^2}{2xd}) - xd\sin(\arccos(\frac{x^2 + d^2 - r^2}{2xd})), \tag{4.49}$$

is the area of the shaded region S_1 in Figure 4.11; d is the distance that node v moves; and r is u's transmission range.

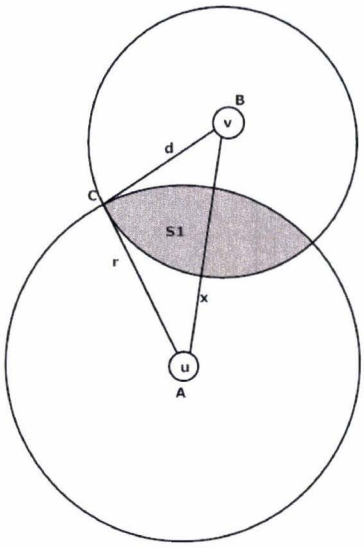

FIGURE 4.11: The probability that node v joins the transmission range of node u.

The probability that an existing neighbor v moves out of the maximum transmission range of node u within time t is given by (Figure 4.12):

$$p_{leave} = \int_{r-d}^{r} \frac{2\pi x}{S_0} \frac{A(S_2)}{\pi d^2} dx, \tag{4.50}$$

where

$$A(S_2) = (\pi - \arccos(\frac{x^2 + d^2 - r^2}{2xd}))d^2 - (\arccos(\frac{x^2 + r^2 - d^2}{2xr})r^2 - xd\sin(\arccos(\frac{x^2 + d^2 - r^2}{2xd}))). \tag{4.51}$$

is the area of the shaded region S_2 in Figure 4.12.

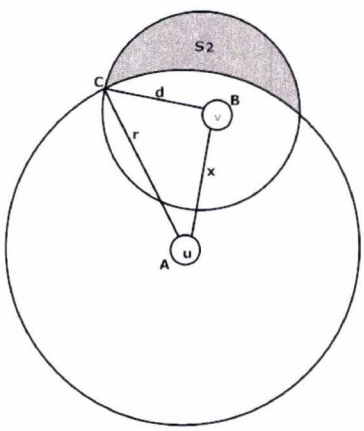

FIGURE 4.12: Probability that node v moves out of the transmission range of node u.

Given that node u has n neighbors and the total number of nodes is N, the probability that no new neighbor enters the visible neighborhood of node u is:

$$p_1 = (1 - p_{join})^{N-n-1}, \tag{4.52}$$

and the probability that no neighbor leaves the visible neighborhood of node u is:

$$p_2 = (1 - p_{leave})^n. \tag{4.53}$$

Thus, the probability that the visible neighborhood of node u changes is:

$$p_{change} = 1 - p_1 p_2. \tag{4.54}$$

Given a predetermined probability threshold p_{th}, we can determine the topology update interval t such that $p_{change} < p_{th}$. Assuming we have a high probability that the visible neighborhood of a node changes, say $p_{change} = 1 - p_1 p_2 > 0.9$, the reconfiguration interval is determined. For pedestrian nodes, (i.e. v=2 mps), the reconfiguration interval is plotted for varying node densities as shown in Figure 4.13. From Figure 4.13, we can see that as the node density increases, the reconfiguration interval decreases, which means we have to run the topology control algorithm more frequently. This is expected because the more densely the deployment region is, the greater the probability that the visible neighborhood of a node changes, so we need to reconfigure the algorithm more frequently.

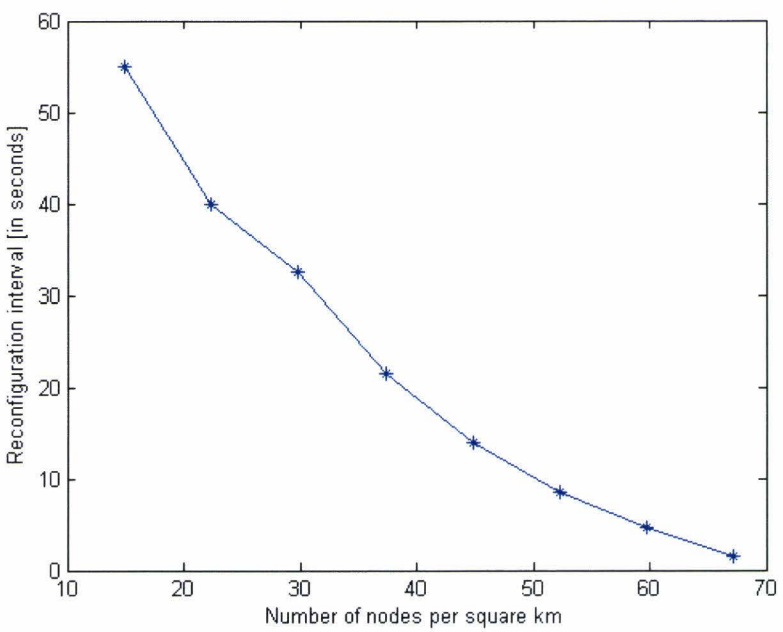

FIGURE 4.13: Reconfiguration interval for varying node densities.

It can also be interesting to see the effect of node speed on the reconfiguration interval. Figure 4.14 shows the reconfiguration interval for varying node speed (for a network density of 50 nodes per square km.). From the figure we can see that as the node speed increases, the reconfiguration interval decreases.

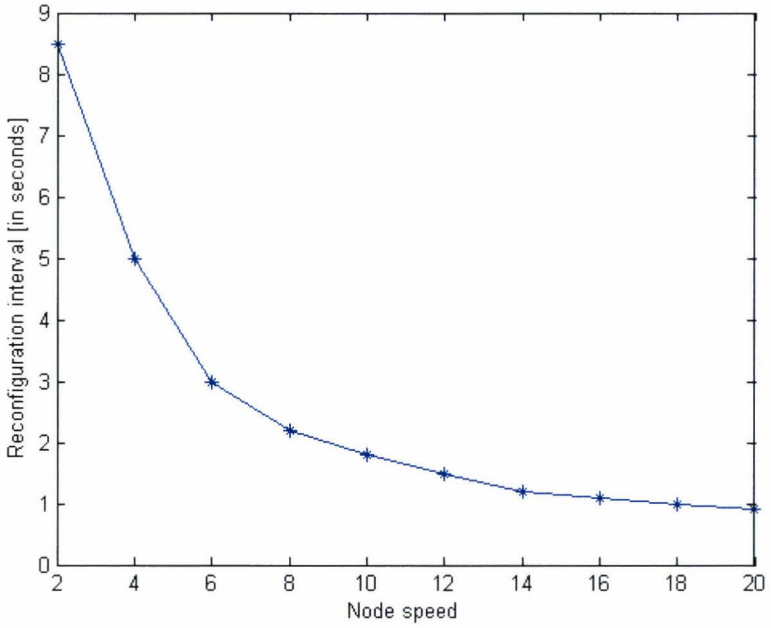

FIGURE 4.14: Reconfiguration interval for varying node speed.

The duration of the reconfiguration is the time it takes for the network nodes to exchange messages (i.e. for neighbor discovery and link quality exchange) and the time each node takes to run the algorithm (i.e. the computation time). The computation time depends on the hardware computing capability. Since the algorithm is simple, we assume that the computation time is negligible.

The time for packet exchange can be determined in the same way we calculate the delay on a link as the same back-off procedure is performed by each node. However, in this case, T_t (Eq. 4.23) and T_c (Eq. 4.24)are equal and are given by:

$$T_t = T_c = T_{DIFS} + T_{beacon} + T_{SIFS} + \gamma, \tag{4.55}$$

where T_{beacon} is the duration of the beacon frame. Since a node exchanges such a frame twice, the total time a node takes to reconfigure is twice the delay on a link:

$$E[t] = 2\frac{1}{\mu_i - \lambda_i}. \tag{4.56}$$

Taking the length of the beacon message to be 50 bytes and at maximum transmission range, we plot the amount of time for reconfiguration versus the network density.

From Figure 4.15, we can see that the time required for reconfiguration (based on the assumption that the computing time is negligible) is small (compared to the time it takes a node to send a packet successfully), but increases with the network density. Since the reconfiguration interval is relatively large, the reconfiguration period is negligible compared to the reconfiguration interval.

The energy consumed during the reconfiguration duration can be calculated in the same way as the energy consumed to send a data packet (Section 4.3). However, in this case,

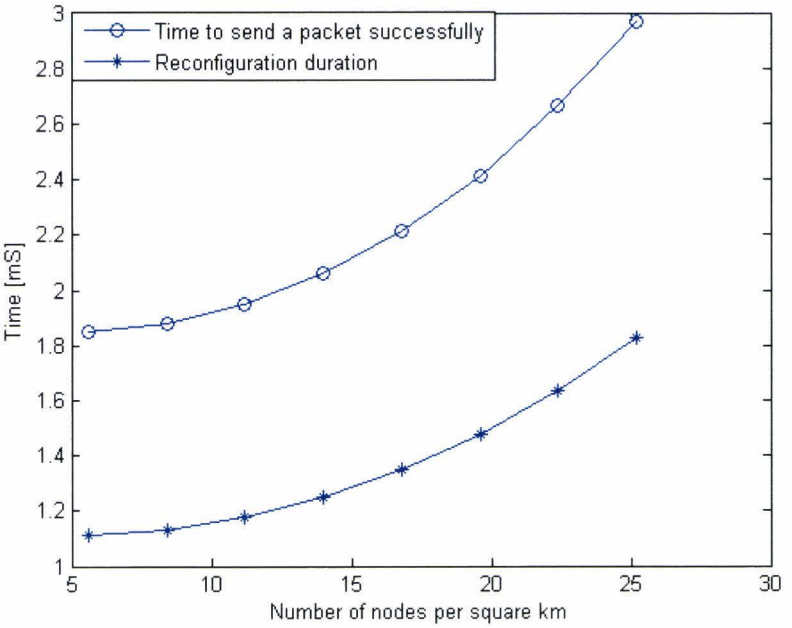

FIGURE 4.15: Reconfiguration duration for varying node speed.

T_t and T_c are given by Eq. 4.55. Taking the length of the beacon message to be 50 bytes and at maximum transmission range, we plot the consumed energy for reconfiguration versus the network density.

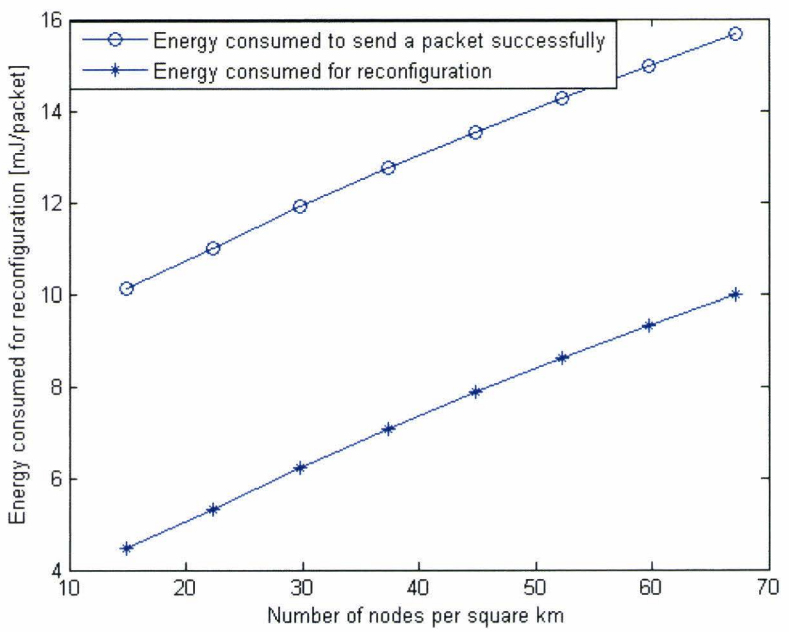

FIGURE 4.16: Energy consumed for reconfiguration.

From Figure 4.16, we can see that the energy consumed for reconfiguration per node is

considerably less than the energy consumed to send a data packet successfully from its source to its destination. Hence, we can see that the network takes a small duration of time and costs less amount of energy to compute the topology, and uses this topology for a relatively longer time (i.e. the reconfiguration interval, e.g. 8 seconds for a walking speed and a network density of 50 nodes per square km.).

4.7 Summary

In this chapter, we analyzed the effect of the transmission range, node degree and the message complexity on the performance of mobile ad hoc network including energy consumption, capacity and delay.

The numerical results of the analyses show that a small transmission range (which also gives a low physical node degree) is necessary to reduce energy consumption and to limit interference (and hence to increase the network capacity). This results in communication along multiple hops, and consequently, the network delay increases. The numerical results also show that, as the message complexity increases, the energy consumption and the network delay increase, but the increase is not significant.

We also studied the reconfiguration interval, the duration of the reconfiguration and the energy consumed during the reconfiguration for a proactive topology control algorithms using statistical model. The numerical results show that, typically, a network takes a small duration of time and costs less amount of energy to compute a topology, and uses this topology for a relatively longer time (i.e. the reconfiguration interval, e.g. 8 seconds for a walking speed and a network density of 50 nodes per square km.)

Chapter 5

Simulation Results and Performance Evaluation

In Chapter 3, we explained the various steps of topology control algorithm and the properties that a topology control algorithm should have. In the previous chapter, we analyzed the effect of the various parameters, such as node degree, transmission range and message complexity on the performance of MANETs including power consumption, capacity and delay.

In this chapter, simulation results of our proposed scheme are presented along with the simulation results of two (modified) protocols, the XTC protocol and the LTCA protocol, for comparison. We use a java simulator package developed by Jan Stemerdink at WMC. The performance metrics we are interested in include:

- Connectivity - percentage of connected nodes in a given time interval.
- Sparsity - average physical node degree vs network density.
- Throughput - the average number of correctly received packets in a given time interval vs network density.
- Energy consumption vs network density.

5.1 Simulation setup

The number of nodes per square km is varied from 0 to 70. The nodes are randomly placed in the simulation area. Nodes are allowed to move in the simulation area in a direction randomly chosen from the interval $[0, 2\pi]$ with a speed of $2m/s$ (walking speed). When the nodes reach at the boundary of the simulation area, they again choose a direction randomly from the interval $[0, 2\pi]$ and move in that direction. The maximum power used is $18dBm$. A radio link between a transmitter unit u and a receiver unit v is established if and only if the power of the radio signal received by node v is above a certain threshold, called the sensitivity threshold.

5.2 Simulation Results

5.2.1 Topology

For inspection of the topologies generated with the various algorithms, we set the number of nodes equal to 50. The simulation area is a rectangular region of sides $1000X1000$ meters.

Figures 5.1,5.2,5.3 and 5.4 show a snapshot of the resulting topologies produced using maximum power, after application of the XTC protocol, after application of the LTCA protocol and after application of our proposed algorithm, respectively. The maximum power topology many links many links. The topology produce using application of the XTC protocol is the most sparse of all the topologies. The topology produced by the proposed scheme has more sparse topology than the maximum power topology and the topology produced using the LTCA protocol, but a little bit denser than that produced by XTC. However, it usually consists of disconnected nodes if it is not shown in this snapshot.

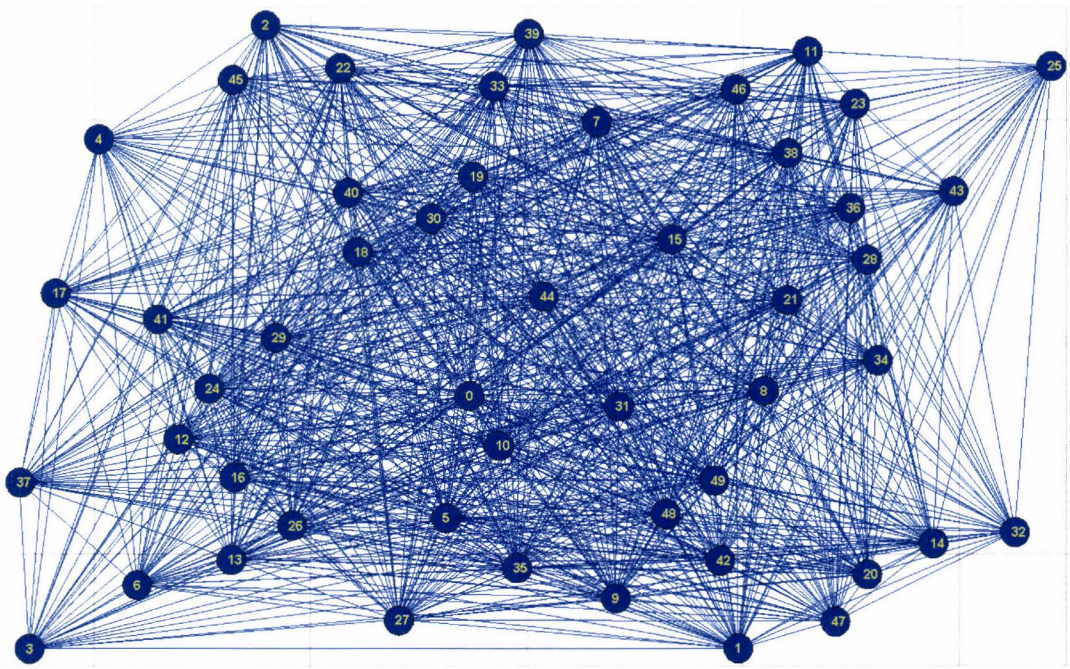

FIGURE 5.1: Network topology at maximum power.

5.2.2 Connectivity

Figure 5.5 shows the connectivity, i.e., the fraction of nodes which are connected in a given simulation time versus network density. When the network density is very low, all the protocols have small connectivity This is because, many nodes may be out of the maximum transmission range of each other. For XTC protocol, we can see that up to 20% of the nodes become disconnected. For LTCA and the proposed scheme, almost 100% of the nodes remain connected.

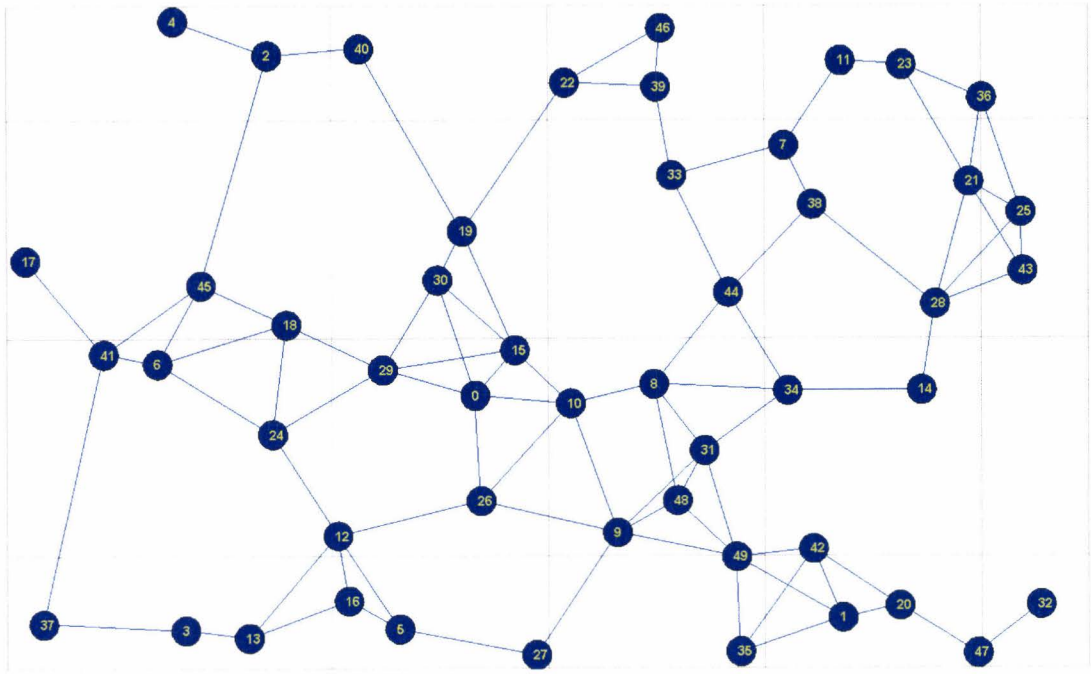

FIGURE 5.2: Network topology after application of XTC topology control protocol.

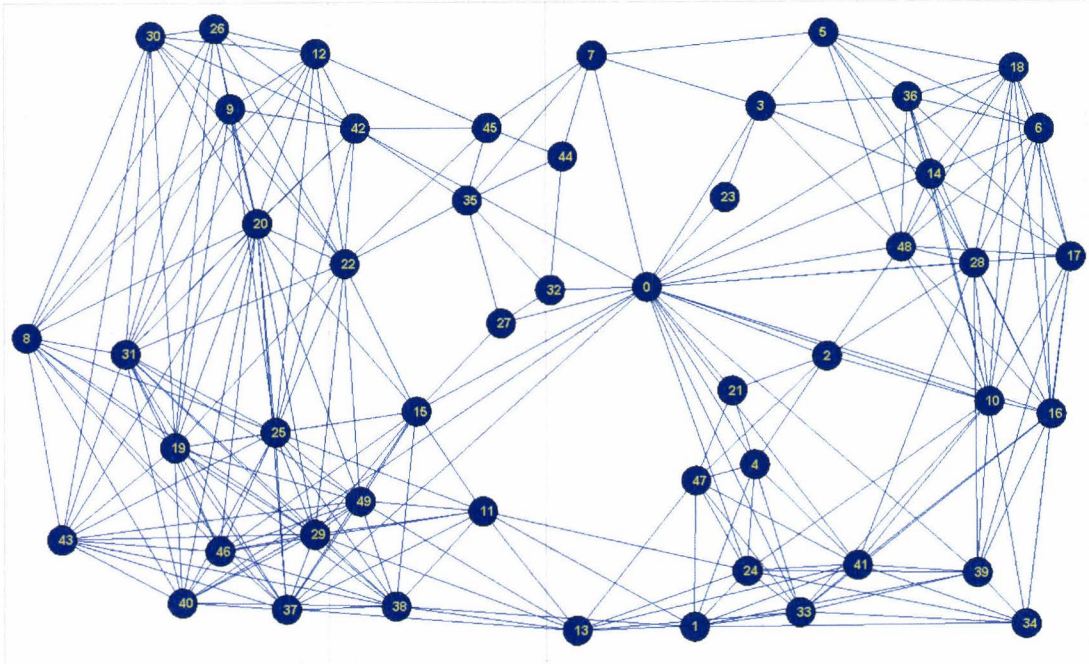

FIGURE 5.3: Network topology after application of LTCA topology control protocol.

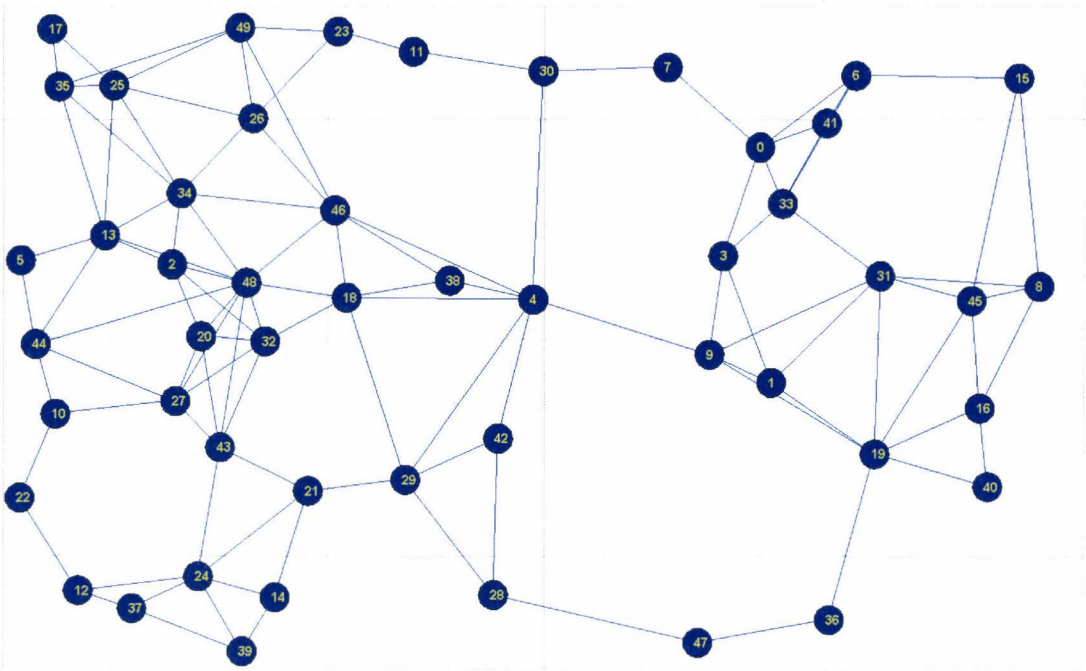

FIGURE 5.4: Network topology after application of proposed algorithm.

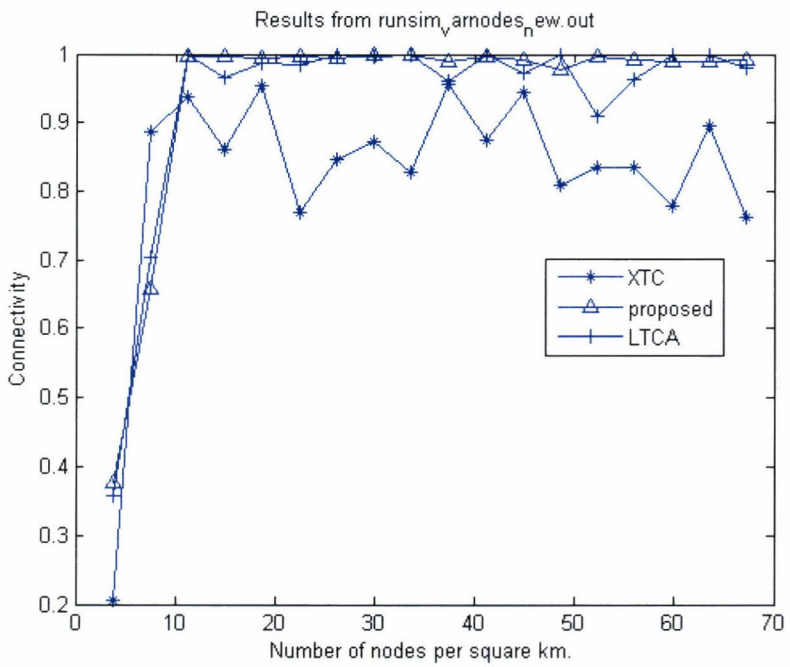

FIGURE 5.5: The fraction of connected nodes in a given simulation time versus density of nodes.

5.2.3 Node Degree

Figure 5.6 shows the average physical node degree versus the network density for the various protocols. The figure shows that the proposed algorithm and the XTC protocol have a low degree which is almost constant, particularly less than 5, when the network density increases. The XTC protocol produces protocol has the lowest average physical node degree; however, this is at the cost of disconnectivity. The LTCA protocol has the highest average physical node degree. It also increases as the network density increases. This is reasonable because, LTCA algorithm works based on node identity, and hence there is a chance that a node v which is farther in distance may be chosen as a neighbor by node u, and consequently, all the nodes within the transmission range of node u to reach node v are also neighbors of node u, because we are now talking about physical node degree.

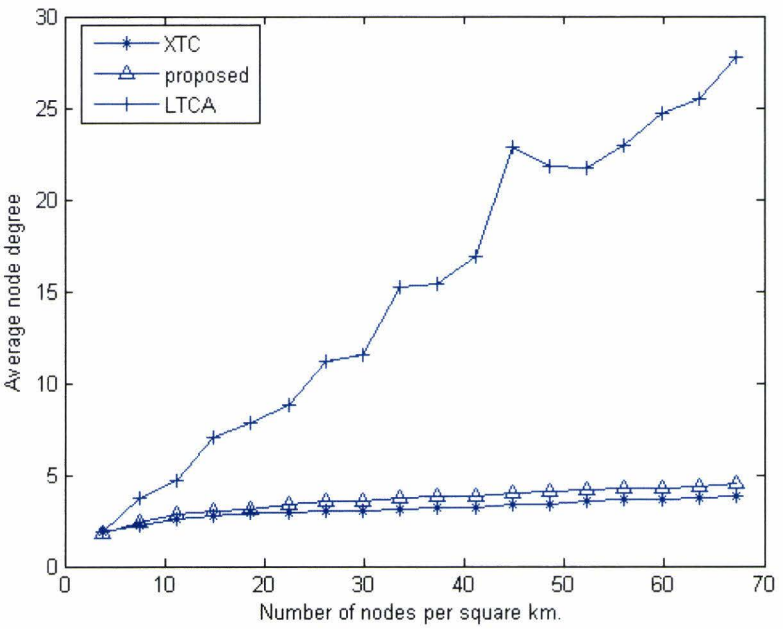

FIGURE 5.6: The average node degree versus density of nodes.

5.2.4 Throughput

Figure 5.7 shows the average throughput per node versus network density. The figure shows that the throughput per node first increases and then decreases as the network density increases. It finally tends to be constant. The first increasing part is due to the fact that, at the beginning, there were disconnected nodes and the per node throughput for disconnected nodes is zero. Then it increases to the point where the network density is low but where the network becomes connected. After this point, the per node throughput decreases as the network density increases. This is because, as the network density increases, both the node degree and the interference from other nodes on a receiving node increase. The fact that the node degree increases means that the node spends a longer time in the MAC procedure to send a packet successfully. The fact that the interference on a receiving from other nodes increases means that the packet is

not received correctly. Both of these factors reduce the number of successfully received packets in a given slot time.

From Figure 5.7, we can also see that the LTCA protocol has the lowest throughput. This is obvious because it has the highest node degree, which also means that it has the highest transmission range and a higher interference on receiving nodes from other nodes. The proposed algorithm has the highest throughput. The XTC protocol has lower throughput than the proposed one although it has the lowest average physical node degree. This is because it has a large proportion of disconnected nodes, since the throughput for such disconnected nodes is zero, the average per node throughput becomes small.

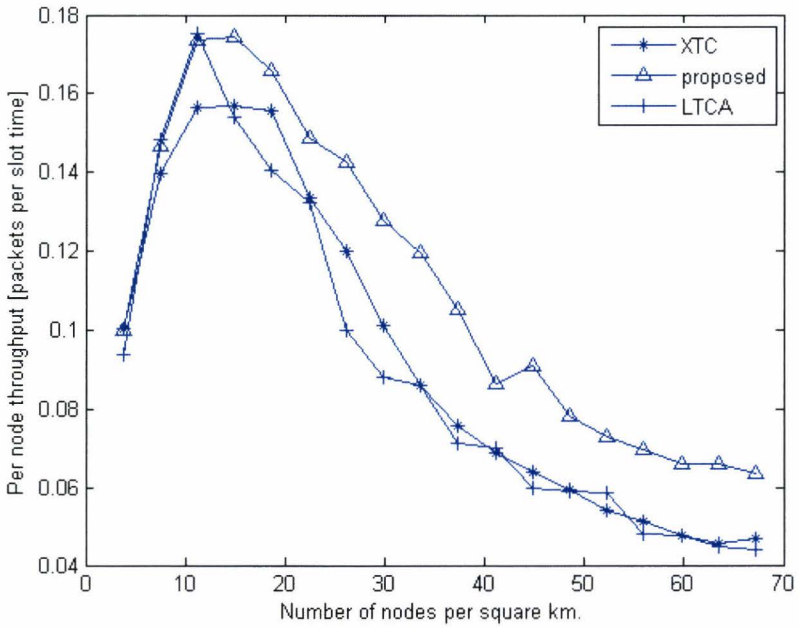

FIGURE 5.7: The average number of correctly received frames per node per slot time versus density of nodes.

5.2.5 Energy Consumption

Simulation results for the energy consumed per packet per node is shown in Figure 5.8. In this simulation, the instantaneous power consumption by a node is equal to the transmit power level. The figure shows that the energy consumed per successfully transmitted power for the *XTC* protocol and the proposed algorithm first decreases and then tends to be constant. This is because, as the network density increases, the transmission range becomes smaller and the transmit power used becomes smaller. So, this reduces the energy consumed even if the average physical node degree slightly increases. For LTCA protocol, the energy consumption tends to increase. This is obvious because the average physical node degree for the LTCA protocol increases significantly as the network density increases.

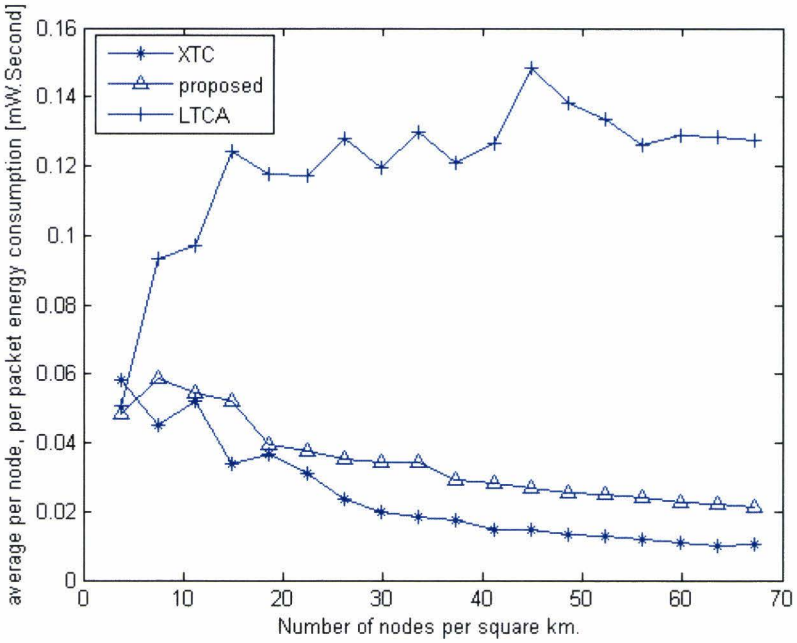

FIGURE 5.8: The average consumed energy per node per frame versus density of nodes.

5.3 Summary

In this chapter, simulation results of our proposed scheme along with the simulation results of two (modified) protocols, the XTC protocol and the LTCA protocol were presented and the performances were evaluated. The performance metrics of our interest were the connectivity, the node degree, the throughput and the energy consumption.

In terms of connectivity, the proposed scheme and the LTCA protocol perform well. In terms of node degree, the XTC protocol produces a topology with the smallest physical node degree; however, this is at the cost of disconnectivity. The LTCA protocol produces the highest physical node degree.

Regarding the average throughput, the proposed scheme gives the highest throughput (i.e. the average number of correctly transmitted packets per slot time). The XTC protocol gives a lower average throughput despite the fact that it has the lowest average physical node degree. This is because, the XTC protocol produces some proportion of disconnected nodes and the throughput per node for those nodes is zero.

The XTC protocol outperforms the others in terms of energy consumption per successfully transmitted packet. This is expected because it has the lowest average physical node degree.

Generally, the simulation results show that a lower transmission range (and hence a small physical node degree) is important to reduce the energy consumption and to increase the throughput, but with the constraint that the network connectivity is maintained.

Chapter 6

Conclusion and Future Work

In the subsequent chapters, we studied a power-based topology control solution for mobile ad hoc networks. This chapter summarizes what has been studied in those chapters and gives a recommendation on possible extension of this work.

6.1 Conclusion

In this thesis, we studied a power-based topology control solution which can be practically implemented, is mobility adaptive and takes into account the available power levels for the wireless card. We study the various steps of power-based topology control solution from the practical point of view using XTC algorithm as our link selection algorithm.

The topology control starts with neighbor discovery. For neighbor discovery, a broadcast beacon message at maximum power is used. Beacon messages are also sent periodically to discover newly introduced neighbors. The next phase is the determination of link quality. Distance between nodes and RSSI value are energy aware link quality metrics. Packet delivery ratio (PDR) and expected transmission time (ETT) are throughput-aware link quality metrics. After neighbor discovery and link quality calculation, every node exchanges this information at maximum power (using the beacon frame). Based on this information, every node locally decides which neighbors it has to choose based on some rule. This is called the link selection phase. We use XTC algorithm as our link selection algorithm. After a node selects its neighbors, the next step is the determination of the transmit power needed to send a message to any neighbor node. We took into consideration our available transmit power levels for this phase.

Two approaches have been presented to make the algorithm adaptive to node mobility: reactive and proactive. In the reactive method, a node is triggered, based on a certain condition, to run the topology control algorithm and establish a new set of links. The advantage of this approach is that, since all the nodes don't need to run the algorithm, reconfiguration control traffic is kept low. A disadvantage is that reactive methods are vulnerable to disconnectivity. In the proactive method, we set a reasonable time interval and the algorithm is re-executed at the beginning of each interval. An advantage of this approach is that connectivity is maintained at any instant in time. A disadvantage is that we will have a high reconfiguration control traffic.

A mathematical analysis is done for the various performance parameters such as energy consumption, capacity and network delay. The numerical results of the analyses show that a small transmission range (which also gives a low physical node degree) is necessary to reduce energy consumption and to limit interference (and hence to increase the network capacity). This results in communication along multiple hops, and consequently, the network delay increases. The numerical results also show that, as the message complexity increases, the energy consumption and the network delay increase, but the increase is not significant.

We also studied the reconfiguration interval, the duration of the reconfiguration and the energy consumed during the reconfiguration for proactive topology control methods using statistical model. The numerical results show that, typically, a network takes a small duration of time and costs less amount of energy to compute a topology, and uses this topology for a relatively longer time.

The simulation results of our proposed scheme along with the simulation results of two (modified) protocols, the XTC protocol and the LTCA protocol were presented and the performances were evaluated. The performance metrics of our interest are the connectivity, the node degree, the throughput and the energy consumption. In terms of connectivity, the proposed scheme and the LTCA protocol perform well whereas the XTC protocol has some proportion of disconnected nodes. In terms of node degree, the XTC protocol produces a topology with the smallest physical node degree whereas the LTCA protocol produces the highest physical node degree. Regarding the average throughput, the proposed scheme gives the highest throughput (i.e. the average number of correctly transmitted packets per slot time). The XTC protocol gives a lower average throughput despite the fact that it has the lowest average physical node degree. This is because, the XTC protocol produces some proportion of disconnected nodes and the throughput per node for those nodes is zero. The XTC protocol outperforms the others in terms of energy consumption per successfully transmitted packet. This is expected because it has the lowest average physical node degree.

Generally, the simulation and analytical results show that a lower transmission range (and hence a small physical node degree) is important to reduce the energy consumption and to increase the throughput, but with the constraint that the network connectivity is maintained.

6.2 Future Work

Topology control works in cooperation with the MAC layer and the routing layer. In Section 3.3, we indicated that, in the case of using ETT as a link quality metric, which is also a routing layer link quality metric, the links (called essential links) selected are kept in the form of a table so that it can be used by upper layer routing algorithm, which can be an advantage in reducing overhead.

Further study can be done so that, instead of two different layers (i.e. the topology control layer and the routing layer) working in coordination, they can be incorporated into a single layer. This means that a protocol will be developed which selects best links and use those best links for routing and keep the transmission range to cover the selected nodes.

Appendix A

Logical versus Physical Node Degree

One of the motivations for topology control is its potential to reduce interference between concurrent transmissions [2]. A typical measure used to quantify the expected interference is the node degree of the communication graph: if the transmitting node u has small degree, relatively few nodes will experience interference during us transmission. For this reason, it is desirable to generate topologies with small average node degree.

In the literature, authers use the term node degree as the number of logically selected nodes by the topology control algorithm. This is referred to as *logical node degree.* However, there may also be other nodes in the transmission range of node u which are not selected by the topology control algorithm, but which can affect its communication. *Physical node degree* refers to the total number of nodes which are in the transmission range of a node u. For instance, in Figure A.1, node u selects two neighbors v and w by a certain topology control algorithm. Thus the logical node degree is two. However, there are four other nodes which are in the transmissio range of node u. Thus node u has a physical node degree equal to six.

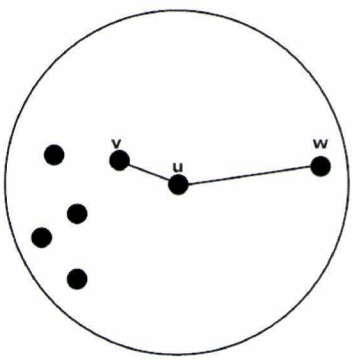

FIGURE A.1: Logical versus physical node degree.

Bibliography

[1] Paolo Santi. Topology control in wireless ad hoc and sensor networks. In *ACM Computing Surveys*, pages 164 – 194, 2005.

[2] Paolo Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks.* John Wiley and Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2005.

[3] Piyush Gupta and P. R. Kumar. The capacity of wireless networks. In *Information Theory, IEEE Transactions on Volume 46, Issue 2.*, pages 388 – 404, 2000.

[4] Ram Ramanathan and Regina Resales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Volume 2*, pages 404 – 413, 2000.

[5] Limin Hu. Topology control for multihop packet radio networks. In *Communications, IEEE Transactions on Volume 41, Issue 10*, pages 1474 – 1481, October 1993.

[6] Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. In *Selected Areas in Communications, IEEE Journal on*, pages 1333 – 1344, August 1999.

[7] Roger Wattenhofer, Li Li, Paramvir Bahl, and YiMin Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Volume 3*, pages 1388 – 1397, April 2001.

[8] Ning Li, Jennifer C. Hou, and Lui Sha. Design and analysis of an mst-based topology control algorithm. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Volume 3*, pages 1702 – 1712, 2003.

[9] Douglas M. Blough, Mauro Leoncini Giovanni Resta, and Paolo Santi. The k-neighbors protocol for symmetric topology control in ad hoc networks. In *Proc. ACM MobiHoc 03, Annapolis, MD*, page 141152, 2003.

[10] Roger Wattenhofer and Aaron Zollinger. Xtc: a practical topology control algorithm for ad-hoc networks. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, page 216, 2004.

[11] Xiang-Yang Li. Approximate mst for udg locally. In *In Proc. COCOON*, pages 364–373, 2003.

[12] Ning Li and Jennifer C. Hou. Localized topology control algorithms for heterogeneous wireless networks. In *Networking, IEEE/ACM Transactions on Volume 13,Issue 6.*, pages 1313 – 1324, December 2005.

[13] Kevin. Lillis and Siriram V. Pemmaraju. Topology control with limited geometric information. In *9th International conference, OPODIS*, pages 427–442, 2005.

[14] Kamrul Islam and Selim G. Akl. Localized topology control algorithm with no geometric information for ad hoc sensor networks. In *Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on*, pages 65 – 72, August 2008.

[15] Huang Chuanhe, Cheng Yong, Li Yuan, Shi Wenming, and Zhou Hao. An interference-aware and power efficient topology control algorithm for wireless multihop networks. In *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pages 330 – 335, 2008.

[16] Li Li and Joseph Y. Halpern. Minimum-mobile wireless networks revisited. In *Communications, 2001. ICC 2001. IEEE International Conference on Volume 1*, pages 278 – 283, 2001.

[17] Kevin J. Krizmant, Thomas E. Biedkatt, and Theodore S. Rappaportt. Wireless position location: fundamentals, implementation strategies, and sources of error. In *Vehicular Technology Conference, 1997 IEEE 47th Volume 2*, pages 919 – 923, May 1997.

[18] R. C. Prim. Shortest connection networks and some generalizations. In *The Bell System Technical Journal 36*, page 13891401, November 1957.

[19] Roger Wattenhofer. Ad-hoc and sensor networks: Worst-case vs. average-case. In *in Proceedings of International Zurich Seminar on Communications, 2004*, pages 156–159, 2004.

[20] S. M. Mousavi, H. R. Rabiee, M. Moshref, and A. Dabirmoghaddam. Mobility aware distributed topology control in mobile ad-hoc networks with model based adaptive mobility prediction. In *WIMOB '07: Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, page 86, 2007.

[21] Zeeshan Hameed Mir, Deepesh Man Shrestha, Geun Hee Cho, and Young Bae Ko. Mobility-aware distributed topology control for mobile multi-hop wireless networks. In *LECTURE NOTES IN COMPUTER SCIENCE*, pages 257–266, 2006.

[22] Prakash Ravi. Unidirectional links prove costly in wireless ad hoc networks. In *DIALM '99: Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 15–22, 1999.

[23] Michel Lammertink. Flame design report release 0.1.4. In *(ti-WMC)*, 2008.

[24] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multihop wireless mesh networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128, 2004.

[25] Amendment 5: Spectrum and transmit power management extensions in the 5 ghz band in europe, 2003. URL `http://grouper.ieee.org/groups/802/11/`.

[26] D. Bertsekas and R. Gallager. *Data Networks, 2nd edition.* Prentice Hall, 1992.

[27] Node delay analysis of routing protocols in mobile ad hoc networks, 2008. URL http://mpc.ece.utexas.edu/Papers/TR-UTEDGE-2008-009.pdf.

[28] Laura Marie Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *In IEEE Infocom*, pages 1548–1557, 2001.

[29] Xiaodong Wang and Dharma Agrawal. Analysis and optimization of energy efficiency in 802.11 distributed coordination function. In *IEEE International Conference on Performance, Computing, and Communications*, pages 707–712, 2004.

[30] Luca Negri, Jan Beutel, and Matthias Dyer. The power consumption of bluetooth scatternets. In *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE Volume 1*, pages 519 – 523, January 2006.

[31] Giuseppe Bianchi. Performance analysis of the ieee 802.11 distributed coordinationfunction. In *Selected Areas in Communications, IEEE Journal on Volume 18, Issue 3*, pages 535 – 547, March 2000.

[32] Ieee standard forwireless lan medium access control (mac) and physical layer (phy) specifications, 1999. URL http://www.cmi.ac.in/~sdatta/networks/standards/802.11-1999.pdf.

[33] Christian Bettstetter and Jorg Eberspacher. Hop distances in homogeneous ad hoc networks. In *The 57th IEEE Semiannual Vehicular Technology Conference, Volume 4*, pages 2286 – 2290, 2003.

[34] A.M. MATHAI, R MOSCHOPOULOS, and G. PEDERZOLI. Random points associated with rectangles. In *Rendiconti del Circolo Matematico di Palermo, Volume 48, Number 1*, pages 163–190, February 1999.

[35] Jan Stoter. Test results transmit power adjustement and power consumption, study report. In *(ti-WMC)*, 2009.

[36] Ramin Hekmat and Piet Van Mieghem. Interference in wireless multi-hop ad-hoc networks and its effect on network capacity. In *Wireless Networks Journal*, page 389399, 2004.