ACHILLES: DESIGN OF A HIGH CAPACITY MESH NETWORK WITH DIRECTIONAL ANTENNAS

SUKANTA KUMAR HAZRA
(B.Eng. (Hons.) NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

Acknowledgments

This thesis is dedicated to Cubic, my Linux machine that worked relentlessly, day and night, to carry out the network simulations.

I would like to express my sincere gratitude to my supervisor Dr. Winston Seah for the guidance, support, and encouragement throughout my thesis work. To I^2R, for the hardware to tinker with which aided, immensely, my learning about the field of wireless communication. And to my friends Parijat and Mansi for providing the motivation to complete this thesis. Last but not least, to the open source community for the great software tools that I used throughput my research.

Abstract

This thesis presents the design of Achilles, a wireless mesh network designed for long distance communication with a typical deployment scenario of maritime mesh network. Achilles uses an antenna system made up of six fixed-beamwidth antennas. Directional antenna is used for both transmission and reception – most other directional antenna schemes use directional antenna for transmission and omni-directional antenna for reception. It uses commodity radio hardware, modified to operate as 6 Mbps transceiver. The MAC protocol used by Achilles is Spatial Time Division Multiple Access (STDMA).

In this thesis, we present practical methods, schemes, and algorithms required for neighbourhood discovery, topology broadcast, and link scheduling required for node using directional antennas. By making efficient use of directional antennas, for both transmission and reception, and spatial reuse in transmission, Achilles achieves the goal of a high capacity mesh network. In this thesis we describe in detail the various components of Achilles and evaluate its performance when compared to alternative mesh schemes. We demonstrate that Achilles performs 2 to 3 times better than IEEE 802.11 and TDMA based mesh networks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent years have seen a tremendous growth in the usage of wireless networks. The commoditisation of wireless transceivers and the availability of unlicensed band has given a boost to the deployment of wireless networks. The IEEE IEEE 802.11 has been the major driver in this arena. Today wireless access is available practically everywhere in urban areas. While in most cases, wireless LANs are deployed as last hop access, there are many successful efforts in using the same IEEE IEEE 802.11 based technology to create wireless backbones. The most well known of these are Mesh Network by Motorola [16], Tropos Networks [23] and the MIT Roofnet Project [4]. All of the mentioned wireless backbone providers use a technology known as mesh networking. In a mesh network, mesh routers route traffic for nodes that they serve directly, as well as for other mesh routers, thus forming a wireless backbone. Ad hoc wireless networks are related to mesh networks and provide similar distributed networking capability. The distinguishing feature of a mesh network, when compared to ad hoc networks, is that mesh routers are deployed in a planned way and are meant primarily as backbone nodes. Ad hoc networks on the other hand are formed when a group of nodes are configured to form a network when in the proximity (radio range) of one another and work together to route one anothers' packets to reach destination beyond direct radio range. Both mesh networks and ad hoc networks use multihop routing to extend the reach beyond direct radio range.

While mesh networks are not completely random, they differ from the conventional cellular networks in that all links are wireless and there is no centralised control. Mesh networks do not require the careful planning and co-ordination that is required in a cellular networks, thus easing their deployment. Mesh routers are also inexpensive, and work in the license free band, thus providing a excellent proposition when creating a wireless backbone. Such wireless backbones have been used in a variety of scenarios, including emergency communication, military communication, and data networks for academic and home use. In this thesis, our goal is to design a mesh network to serve as a maritime wireless communication backbone. Such a mesh network could be deployed in a port area to serve ships when they wait at the shore, or pass through the shipping lanes near the port.

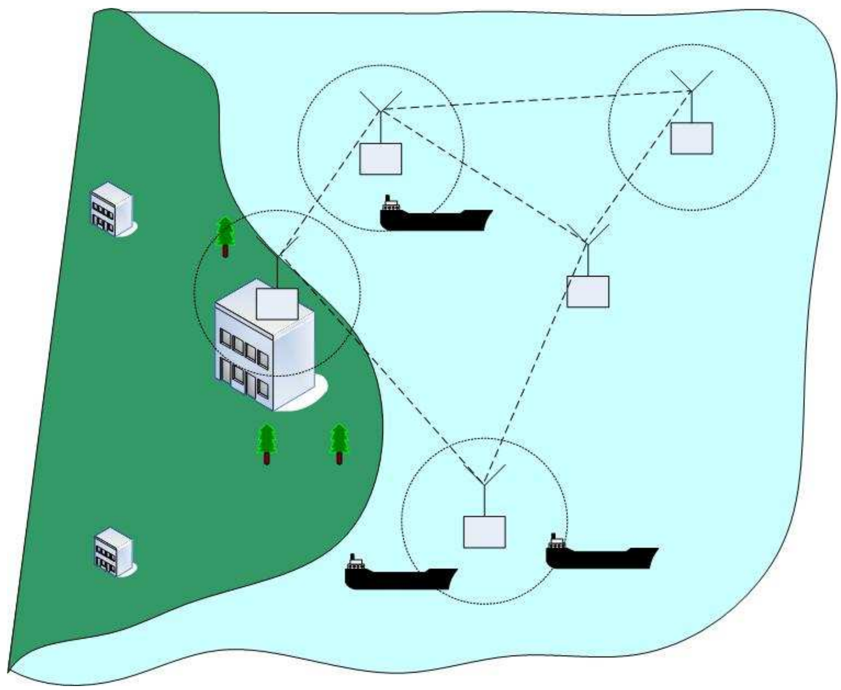


Figure 1.1: Concept of a maritime mesh network

Our usage scenario requires us to use as few mesh nodes as possible to cover a large region. Deploying mesh nodes on buoys in the sea is an expensive proposition. While the mesh node itself is inexpensive, the cost and complexity of setting up buoys is a major constraining factor. Therefore, we need the mesh nodes to be able to communicate at large distances, requiring us to use directional antenna to improve the gain and thereby the communication range. For increased communication range, we need both the transmitter and the receiver to use directional antennas.

The use of directional antennas poses several challenges, while at the same time delivering advantages. In wireless networks, the wireless medium is the most critical resource that determines the capacity of the network. Using omnidirectional antennas results in wastage of this resource by radiating energy in all directions rather than the direction of desired communication. In recent years, there has been a growing interest in the use of directional antennas to better utilize the wireless medium. Directional antennas have multiple advantages, the enhanced spatial reuse being the most obvious one. In addition, the high gain of directional antennas enables communication at greater distances; add to that the multipath mitigation properties, and we have a very compelling proposition in the use of directional antennas.

The challenges associated with the use of directional antennas stem from the fact that schemes and protocols designed for multihop wireless networks are geared towards the omnidirectional mode. Using directional antennas requires new methods for neighbourhood discovery, network-wide broadcast, transmission scheduling – to name a few. In particular, when directional transmission and directional reception are used (no omnidirectional antenna), ensuring that both transmitter and receiver antennas are pointing towards each other is a challenge. It requires the presence of link scheduling algorithms that establishes the link at the desired time by switching the antennas in the appropriate directions. To ensure good performance in the network, the link scheduling algorithm must schedule links network-wide, such that interference and collisions are minimized. In this regard, Spatial TDMA algorithm proposed by Nelson and Kleinrock in [17] is a good candidate for a link scheduling algorithm. From the practical point of view, creating the link compatibility matrix required by STDMA is not straightforward. Two popular approaches exist, a graph-based approach, and an interference-based approach. For a practical network, the interference model is suitable (further details are provided in later sections). However, calculating the interference a priori – at deployment time – is neither trivial nor accurate. Obstacles, multipath effect, hardware inhomogeneity, etc. pose difficulties in calculating the interference. To overcome this problem, we propose a scheme in which nodes perform real-time measurements to determine the compatibility matrix. This ensures that the network is self-configuring, and is not dependent on a priori knowledge of the interference characteristics.

Definition 1.0.1. This thesis presents the design of Achilles. Achilles encompasses a wireless communication system design consisting of an antenna system made up of six fixed-beamwidth antennas of 60^o beamwidth each. Directional antenna is used for both transmission and reception. It uses commodity IEEE 802.11a/b/g wireless network card, modified to operate as 6 Mbps transceiver. The MAC protocol used by Achilles is Spatial Time Division Multiple Access (STDMA). Achilles includes a neighbour discovery mechanism, a topology dissemination mechanism and a mechanism to determine the link compatability matrix required for STDMA.

1.1 Contributions of This Study

This thesis contributes the following:

- A deterministic neighbour discovery mechanism which ensures with very high probability that all the neighbours are discovered within a fixed time.
- A bootstrap mechanism to allow topology information to be disseminated to all the nodes in the network.
- A method to determine the link compatibility matrix required for link scheduling in STDMA, based on measurements.

The thesis is organised as follows: in Chapter 2, we begin by explaining some of the technologies and prior work related to this thesis. In particular, we look at basic antenna concepts in order to develop a feel for the behaviour of directional and omni-directional antennas. Our goal is to select an antenna system that is practical and affordable, while at the same time satisfying our design goals. We find that a set of six fixed-beam directional antennas together with RF switching circuit serves our purpose well. We also present the related work on transmission scheduling and the use of directional antenna for mesh networks.

In chapter 3, we present our scheme for neighbourhood discovery. The use of directional antenna complicates the otherwise straight-forward mechanism for neighbourhood discovery. Nodes send packets in certain directions, and there is no guarantee that the intended receiver is listening in the required direction. We solve this problem by devising a scheme to ensure guaranteed neighbourhood discovery within a bounded time.

In Chapter 4, we present a network-wide broadcast scheme. The same issues that make neighbourhood discovery difficult also present obstacles in broadcasting packets to all nodes in the network. Broadcast is an important requirement during the bootstrap phase, without it control packets cannot be sent to all nodes in the network. We solve this problem by proposing a method that specifies the antenna switching behaviour of the nodes in the network when broadcast is required.

In Chapter 5, the main problem of link scheduling is tackled. Link scheduling is required so that nodes know which antenna to select at what time, and whether to transmit or not. For link scheduling we use Spatial TDMA (STDMA) as proposed by Gronkvist et. al. in [9], which provides an algorithm for assignment of timeslots to links based on link priorities. The challenge is in determining a set of compatible links – links that can transmit at the same time without causing interference. Gronkvist proposes the use of interference calculation using propagation models. We differ from Gronkvist in that we believe interference calculation is very time consuming, often less than accurate, and requires extensive study of the deployment area. In order to build a practical, easy-to-deploy, mesh network, we resort to in-field testing of links to determine whether or not the links are compatible. The chapter details the testing procedure, and the required messages, and the schedule calculation algorithm.

We move on performance evaluation of Achilles in Chapter 6. To compare the performance of Achilles, we test it against IEEE IEEE 802.11 and TDMA mesh networks. The simulation results show that STDMA's performance is much better than IEEE IEEE 802.11 and TDMA, out-performing both by 2-3 times. The performance improvement is achieved at the the cost of a more complicated system using directional antennas, antenna switching circuits and algorithms as against simpler omnidirectional antenna system. However, this cost has to be incurred in order to obtain significantly higher network throughput.

Finally we conclude in Chapter 7 with some directions for future work.

Thesis Statement: Directional antennas vastly improve the capacity of wireless mesh networks by using radio resources in an efficient manner. With the use of directional antennas and spatial reuse TDMA, a high capacity wireless mesh network can be designed to serve as a wireless backbone.

Chapter 2

Background and Related Work

In this chapter, we discuss some of the technologies and techniques which form of the background of this work. In particular, we discuss the various kinds of directional antennas in use and the general theory of their operation. We follow that with a discussion of multihop networks. In our work we use Spatial TDMA as the Medium Access Control Protocol (MAC), and we provide a description of its principle in this chapter. Related work in neighbourhood discovery and transmission scheduling are also presented.

2.1 Antennas

For wireless radio communication to work, energy from the transmitter must be radiated, and then received by the receiver. Antennas perform the critical function of transmitting the radio waves, and receiving them. Two main categories of antennas are commonplace: i) *omnidirectional* antennas radiate in all directions with almost equal gain, and are usually modeled by a circular transmission radius ii) *directional* antennas, on the other hand, have a preferred direction of transmission, and are usually modeled by a sector of angle θ. The gain of the antenna is highest in the preferred direction. The directional discrimination provided by the directional antenna can be exploited to increase the spatial reuse of the wireless medium, and thus increase the network capacity. The most important characteristics of an antenna are its beamwidth and the gain. The beamwidth specifies the 3 dB width (in angle terms) of the main lobe of the antenna. The antenna gain measures the increase in signal strength as compared to a dipole antenna (dBd) or a theoretical isotropic antenna (dBi). The maximum gain of the antenna is known as the bore-sight gain. In general, the smaller the antenna beamwidth, the higher the bore-sight gain. This is because the antenna squeezes more energy in a narrow lobe thus providing higher signal strength in the bore-sight.

2.1.1 Directional Antenna Models

Before delving into directional antennas, we describe some basic terminology related to antennas. Antenna *radiation pattern* or *antenna pattern* is the most important tool to describe the performance of an antenna. It is a graphical representation of the radiation properties of the antenna as a function of space coordinates. Typically, a radiation pattern shows the spatial distribution of the radiated energy. Figure 2.1 shows a 3-D view of radiation patterns.

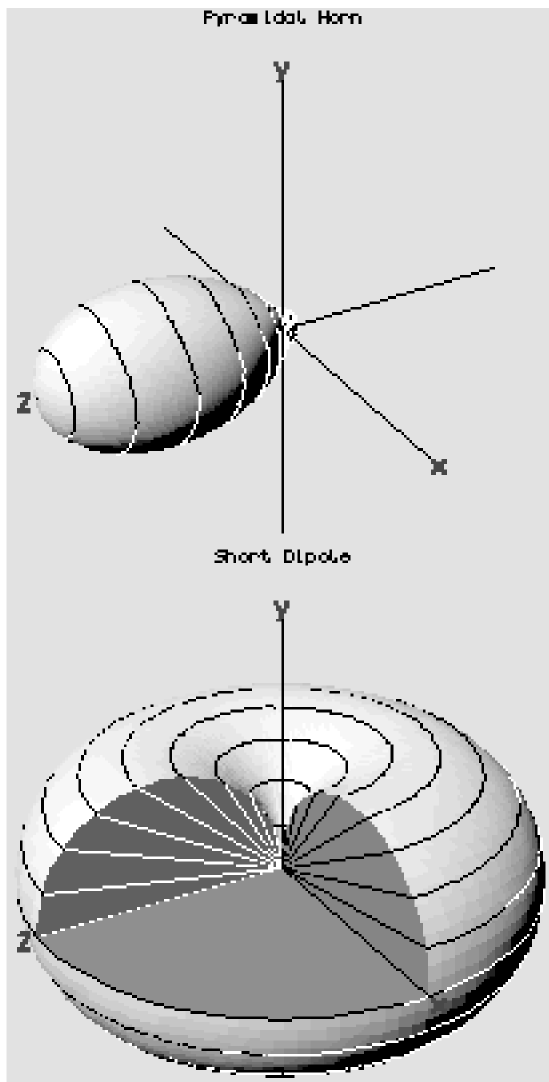


Figure 2.1: 3-D representation of antenna radiation pattern of a directional and omni-directional antenna, courtesy wikipedia.com

Antennas are often compared against a hypothetical *isotropic* radiator. An *isotropic* radiator is a loss-less antenna having equal radiation in all directions (in 3-D the radiation pattern appears as a sphere). An *omnidirectional* antenna, on the other hand, has has equal radiation in all directions in the azimuth plane, but not in the elevation plane.

An antenna pattern is used to show the behaviour of a given antenna. In general, two patterns are specified for each antenna: the azimuth pattern, and the elevation pattern. The azimuth pattern is a plot of the gain of the antenna in the horizontal plane in different directions. The elevation pattern is a plot of the gain of the antenna in the vertical plane, for different elevation angles. Together the azimuth and elevation patterns allow the calculation of the gain of the antenna at any point in 3-D space, around the antenna.

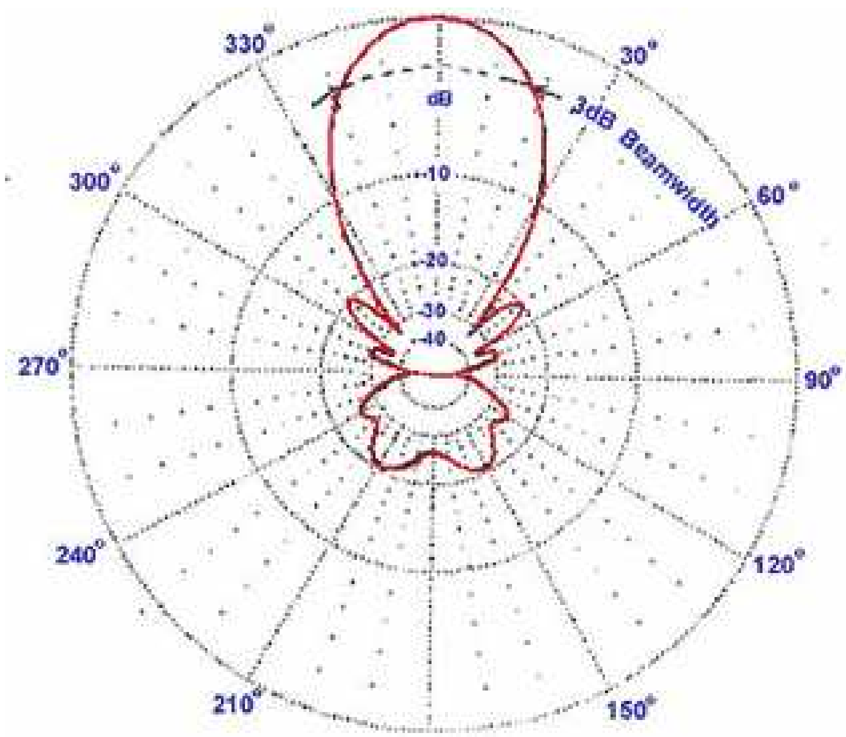


Figure 2.2: Azimuth pattern showing the 3 dB beamwidth

A term often used in directional antennas is *lobe*. A *radiation lobe* is a portion of the radiated energy bounded by regions of relatively weak radiation intensity. Directional antennas typically have one visibly large lobe, and several minor side and back lobes, c.f. Figure 2.2.

The *directivity of an antenna* is defined as the ratio of the radiation intensity in a given direction from the antenna to the radiation intensity averaged over all directions.

The average radiation intensity is equal to the total power radiated by the antenna divided by 4π. Often the directivity of the antenna is specified without mentioning any specific direction. In those cases the direction is assumed to be the one with maximum radiation intensity.

$$D = \frac{U}{U_0} = \frac{4\pi U}{P_{rad}} \tag{2.1}$$

If the direction is not specified, it implies the direction of maximum radiation intensity expressed as

$$D_{max} = D_0 = \frac{U_{max}}{U_0} = \frac{4\pi U_{max}}{P_{rad}} \tag{2.2}$$

D = directivity (dimensionless)

D_0 = maximum directivity

U = radiation intensity (W/unit solid angle)

U_{max} = maximum radiation intensity (W/unit solid angle)

U_0 = radiation intensity of isotropic source (W/unit solid angle)

P_{rad} = total radiated power (W)

Another important measure describing the performance of an antenna is the *gain*. Gain is related to the directivity, however, it takes into account the efficiency of the antenna, as well as its directional capabilities. The *relative gain* of an antenna is commonly used. In relative gain, the antenna radiation power at a point is compared to that of a isotropic radiator if the same power was fed in both the antennas:

$$G = 4\pi U(\theta, \phi) \tag{2.3}$$

θ = is the angle of azimuth

ϕ = is the angle of elevation

When the direction is not stated, the power gain is usually taken in the direction of maximum radiation.

The *half-power beamwidth* often called the *beamwidth* of the antenna describes the angle between the two directions in which the radiation intensity is one-half the maximum value of the beam. Typically, the more directional the antenna, the higher the gain, and smaller the beamwidth.

Several types of directional antennas exist. In this section, we provide a brief description of four major types:

- **Single beam:** In single beam antennas the antenna has a single major lobe. The antenna couples most of radiated energy in this lobe. Single beam antennas can have very high directivity and large gain. They are usually passive structures and do not require sophisticated signal processing. Single beam antennas are widely used in microwave and satellite communication.

- **Switched beam:** These antennas have multiple elements allowing RF power to be switched to one or more of the elements present. Switched beam antennas are simple and do not require sophisticated signal processing. The limitation is that the radiation pattern of the antenna is fixed, allowing only a choice of one of the possible patterns.

- **Steered beam:** These antennas have a radiating element with fixed pattern, however, they can be mechanically steered in different directions. Such antennas are commonly used in radars and signal scanners.

- **Beamforming:** These are the most sophisticated type of directional antennas and work on the bais of constructive and destructive interference of radio waves. By shifting the phase of the input RF wave, the radiation beam can be changed to the desired beam pattern. Such antennas use fairly sophisticated RF technology and are bulky and expensive. They are used mainly in military applications for countering radio jamming by using a technique known as null-steering.

2.2 Multihop Wireless Networks and the Issue of Network Capacity

The focus of this thesis if on multi-hop wireless networks. In these networks nodes are equipped with a wireless transceiver and able to communicate with neighbouring nodes using the wireless medium. In addition to being source and destination of packets, nodes also route packets for other nodes in the network, thus they act as routers as well. Omni-directional antennas are a popular choice for node in multi-hop wireless networks. In order to reach a destination node that is beyong the radio range, a node can solicit the support of neighbouring nodes to route packets to the destination in a multi-hop manner [12]. Such multi-hop or *ad hoc* wireless networks often do not have any centralised control, the lack of which give rise to many issues at the network, medium access control (MAC), and physical layers, which have no counterparts in wired networks like Internet, or in cellular networks.

IEEE 802.11 distributed co-ordination function (DCF) is one of the most popular MAC protocols used in multi-hop wireless networks. However, the use of a contention-based MAC such as IEEE 802.11 leads to low netowork performance due to wasted opportunity to transmit as a result of contention and backoffs [15]. The popular use of omni-directional antennas means that nodes are affected by on-going transmissions in all directions, thus worsenig the contention. Use of contention-free protocols such as TDMA is desirable, however the lack of centralised control creates new challenges in their use. Another extension to improve the performance is the use of directional antennas. However, directional antennas introduce new challenges of link scheduling and neighbourhood discovery [14]. The use of directional antennas with IEEE 802.11 resurfaces the problem of hidden terminals. The hidden terminal problem arises due to possibility that transmission from two nodes which cannot hear each other, may interfere at a third node. Modern MAC protocols for omnidirectional antennas have taken this problem into account [13, 3], and schemes to extend the solutions to directional antennas have been proposed in [14, 22, 6]. Schemes such as Directinal MAC [6] introduce the concept of a Directional Network Allocation Vector (DNAV) to solve the issue of *deafness* (a phenomenon where a node can not hear the channel reservation requests from other nodes due to its directional antenna pointing away from the requesting node) and hidden-terminal introduced as a result of use of directional antennas. These extensions to IEEE 802.11 MAC protocol improve the performance of IEEE 802.11 when used with directional antennas, however, they do not fully solve the issue of contention, which is inherent to any contention-based MAC protocol.

The contention free properties of TDMA based MAC protocols alleviates the problem of contention. In TDMA, each node (node TDMA) or link (link TDMA) is given the opportunity to transmit in specific slots, with the guarantee that there will be no other transmissions from other nodes or links. This ensures that all transmissions are contention free. However, as the number of nodes or links increase in the network, the length of the TDMA frame (proportional to number of nodes) increases, resulting in increased packet delay. Nelson and Klienrock [17] proposed a scheme that takes advantage of the fact that radio transmissions that are sufficiently separated in space do not interfere with each other. By taking advantage of the spatial diversity, multiple transmissions can be scheduled in a single time slot, thus reducing the length of the TDMA frame, and allowing more transmissions in each time slot. This observation is the basis of Spatial TDMA (STDMA). STDMA is the MAC protocol of choice for Achilles.

The capacity issue was studied extensively by Gupta and Kumar [12], where they showed that the per-node throughput of the network scales as $\Theta\left(\frac{W}{\sqrt{n\ log\ n}}\right)$, where W is the link bandwidth and n is the number of nodes in the network. The theoretical limitation is a result of the routing burden on the nodes. To improve the capacity of the network, they suggested: i) reduction in unintended interference, ii) optimal scheduling at the MAC layer, and iii) power control. In the design of Achilles we have taken into account these suggestions to enhance the capacity of the network.

2.3 Link Scheduling and Spatial TDMA

In order to avoid contention at the MAC layer (which results in back-offs and collisions), TDMA is an attractive candidate. However, TDMA is unable to benefit from spatial reuse of radio resources. Nelson and Kleinrock [17] are the first to suggest a spatial reuse TDMA scheme. The basic premise of STDMA is that transmissions that are sufficiently spatially separated do not interfere with each other and therefore are permissible in the same time slot. By allowing multiple transmissions in the same time slot, the length of the TDMA frame is shortened, resulting in lower delays. Allowing multiple transmissions in the same time slot improves network throughput. The main challenge in STDMA is to calculate which nodes or links can transmit at the same time without interfering. In order to determine the *compatible* links, Nelson and Klienrock used a graph model of the network (not taking additive interference into account). The STDMA scheme proposed forms the basis of Achilles's MAC. The spatial reuse enhances the capacity of the network while at the the same time keeps the TDMA schedules short, ensuring low delay in the network.

Gronkivst et. al. in [9, 11] extended Kleinrock's STDMA using an interference model, instead of graph model, of the network. Their work uses wave propagation library Detvag-90, to calculate the path loss between transmitter and receiver. With extensive knowledge of the terrain and wave propagation characteristics, the compatible links in each STDMA slot are determined. They also proposed an algorithm to schedule links in each STDMA slot. Achilles uses the same algorithm, however, instead of using wave propagation library, Achilles determines the compatible links by means of link test. The link test approach is more robust as it takes into account all factors including, terrain, propagation, as well as multipath and fading.

A problem with STDMA is the optimal selection of compatible link sets and the optimal assignment of time slots. The problem is shown to be NP-hard problem [5]. Grönkvist [10] proposed two assignment methods for STDMA. The first assignment method is *node assigned* schedule, in which each node is allowed to transmit to any of its neighbours in its slot. The second assignment method is *link assigned* schedule, in which each directed link is assigned a slot. A node can then use this slot to transmit to a specific neighbour. Performance analysis in [10] showed that *link assigned* schedule performs better than *node assigned* schedule. Since Achilles uses purely directional antennas, *link assigned* schedule is more suitable and we therefore use the *link assigned* algorithm proposed by Grönkvist for time slot assignment.

Sanchez et. al. [21] suggest a scheme Reuse Adaptive Minimum Hop Algorithm (RA-MHA) – an extension to link assigned schedule by taking routing into account. The goal is to minimise the number of hops to the destination. The routing in turn determines the expected traffic load on each link. The expected traffic load is taken into account when assigning time slots to each link – assigning more slots to busy links. The authors claim that by combining routing with scheduling for STDMA, substantial improvement in throughput and packet delay can be obtained. Achilles does not adopt this algorithm because in Achilles the traffic patterns are not pre-determined.

2.4 Mesh Networks using Directional Antennas

Much of the early work on directional antenna focused on extending the carrier sense multiple access/collision avoidance (CSMA/CA) scheme (in particular for IEEE IEEE 802.11) to work with directional antennas. A directional network allocation vector (NAV) is proposed by Takai et. al. in [22]. The directional NAV scheme works on the principle that, if a node receives a request-to-send (RTS) packet or clear-to-send (CTS) packet from a certain direction, then it needs to defer only for those transmissions that are in and around that direction. The node could continue to transmit in other directions. In [14], a scheme to use multiple fixed directional antennas is proposed. This scheme requires multiple radios, one radio per antenna. Ramanathan et. al. [19] proposed a fairly comprehensive scheme called utilizing directional antennas for ad hoc networking (UDAAN) which specifies neighbour discovery, MAC, as well as routing. The basic mechanism is still CSMA/CA enhanced for directional antennas. The major shortcoming of the scheme is that it requires the receiver to be in omnidirectional mode when *control* packets are transmitted, after which the the receiver can switch to directional antenna.

There is limited amount of work in the area of TDMA using directional antenna. In [8], the authors study the performance of STDMA in a network with beamforming antenna arrays. They show a capacity gain of up to 980% when using beamforming antenna for receiving. We derive much of our motivation to use STDMA from the performance improvements shown in [8]. The authors do not specify any practical method of using their results, limiting themselves to a theoretical network.

Another TDMA based scheme using directional antennas is proposed in [2]. The authors describe a scheme called Receiver Oriented Multiple Access (ROMA) which is designed to use multi-beam adaptive array (MBAA) antennas. ROMA is one of the few protocols that is able to use directional antenna for both transmission and reception. However, neighbourhood discovery is probabilistic and link schedules in ROMA are nondeterministic resulting in uncertainty about delays. ROMA is an on-demand channel access protocol, which is desirable for a mobile ad hoc network, but not particularly suited for a static mesh network.

2.5 Neighbourhood Discovery Mechanisms

In this section, we look at some of the neighbourhood discovery algorithms proposed in literature for the case when either transmitter or receiver or both use directional antennas.

- **ROMA**: In this model [2], nodes randomly select a slot in the TDMA frame to transmit with probability p. Nodes transmit the *hello* packet n times. The value of p and n are calculated to ensure discovery with high probability.

- **UDAAN**: works even when both transmitter and receiver use directional antennas. Nodes send heartbeats periodically while steering their antenna in a clockwise direction. All nodes in the network transmit heartbeats in the same direction. If a node wants to receive heartbeats, it steers the antenna 180^o, and therefore, can receive the transmission, if any from its neighbours. If, however, two transmitting nodes are close by and in line with the receiver, the heartbeat will be lost due to interference.

- **Gossip-based algorithm**: Vasudevan et. al. [24] present the analysis of neighbour discovery based on random transmission. The optimal transmission frequency and number of required transmissions for neighbourhood discovery, with high probability, is calculated. To enhance random discovery they propose a gossip-based scheme in which nodes share their neighbour information with other (already discovered) nodes.

2.6 Network Design, Notations and Assumptions

In this section, we discuss the underlying network design on which this thesis is based. There are some peculiarities of our antenna system and network deployment scenario which affect the terminology in the rest of the thesis and we will point those out.

The work in this thesis is geared towards the development of an oceanographic backbone with the intention of providing a fast and cheap data network for vessels, and sensors deployed in the coastal region. The network should cover a large region, several square kilometers, using as few backbone nodes as possible. The reason for this is the cost and the difficulty of deploying buoys in the busy shipping lanes. This calls for very sparse deployment of backbone nodes, thus requiring very long distance communication. We solve the problem of long distance communication by using high gain directional antennas. To make full use of the gain provided by the antenna, we need both directional transmission and directional reception.

Once the nodes have been deployed, the network should be self-configuring with minimal centralised control. Due to the hostile environment of the oceans, node failures do occur, and the network must have a way to recover from the failures, and reconfigure the whole system. Therefore, in this thesis, we discuss methods by which the network can bootstrap from the time they are deployed and maintain a working state throughout its lifetime.

In summary the design goals of Achille's are:

- Cover a large geographical region with minimal number of nodes
- Provide high throughput and low delay backbone for ships and oceanographic sensor networks
- Be self-configuring with mimimal centralised control

2.6.1 Node and Antenna Design

Each node in the network under consideration consists of a single transceiver operating in the 5.8 GHz band. The antenna system consists of six antennas, each a 60^o beamwidth antenna with a gain of 16 dBi. The RF output/input of the transceiver is channeled to/from one antenna at a time using an antenna switch. The antenna switch is controlled from the parallel port of the computer by sending the appropriate control signal. The antenna switching code resides in the kernel, and can switch an antenna within 2 μs. The GPS receiver provides the time synchronisation required, and is accurate within 1 μs. The GPS receiver also provides the position (location) coordinates of the node.

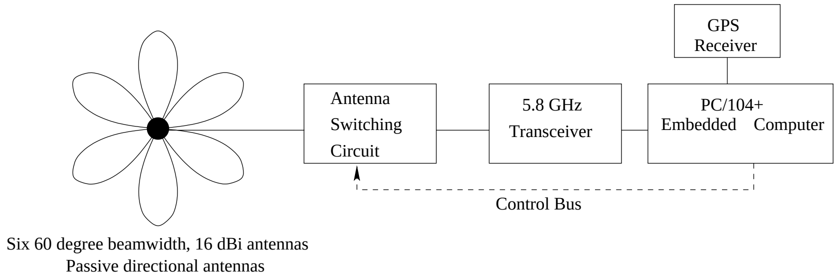


Figure 2.3: Block diagram of a node

As can be seen from the block diagram shown in Figure 2.3, the antenna system is a *switched antenna* system, as opposed to the *switched beam* antenna systems often discussed in literature. Logically, switched antenna and switched beam antenna are equivalent. Both of them can have k major antenna lobes or beams which they select one at a time. Since we switch between antennas, we often refer to *antenna switching*, which for the purpose of discussion is the same as *beam switching*. Moreover, because of the design of the antenna system, we do not have any *omnidirectional* mode of reception. All the elements in the antenna system are directional antennas (MARS Antenna Model: MA-WC50-5X). The combined pattern of the antenna system is shown in Figure 2.4.

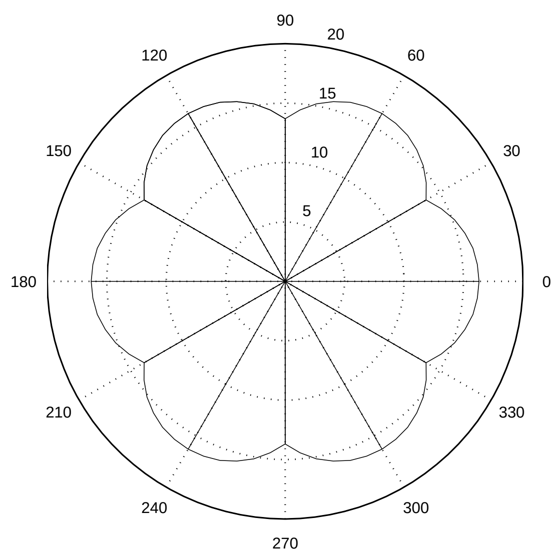


Figure 2.4: The combined pattern of the antenna system

Definition 2.6.1. *Antenna switching* or *beam switching* is the process of selecting one antenna out of an array of k possible antennas. At any given time a transceiver can have only one active/selected antenna.

Definition 2.6.2. We define in this thesis two nodes as being *tuned* when the two nodes have switched their antennas such that they are pointing to each other, c.f. Figure 2.5. We also refer to antennas being tuned as having the same meaning.

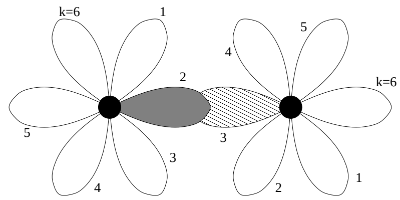


Figure 2.5: Tuned nodes – nodes that have selected their antenna such that the lobes overlap and thus communication between them is possible. The solid lobe shows antenna being used for transmission. The striped lobe shows antenna being used for reception. Unfilled lobes show antennas that have not been selected.

Definition 2.6.3. We define (or refer to) *transmission range (radius)* as the maximum distance at which communication is possible when both the nodes are tuned. The transmission range depends on the transmitter power, antenna gain and the receiver sensitivity.

2.6.1.1 Antenna Gain and Orientation

The gain of a directional antenna is not constant in the 3 dB beamwidth region. There is in fact a difference of 3 dB (by definition) between the gain at the bore-sight and the gain at the 3 dB angle. Thus, even when two nodes are tuned, depending on their orientation and position, there can be a difference of up to 6 dB in the antenna gain. Figure 2.6 illustrates this.

This 6 dB difference in antenna gain results in the variation of transmission range. When following the free space propagation model, a 6 dB gain results in (roughly) doubling the transmission range. To capture this variation, we define two transmission ranges, a *minor transmission range* and a *major transmission range*. Figure 2.7 illustrates the concept of two transmission ranges.

Definition 2.6.4. *Minor transmission range* is the maximum distance at which two nodes can communicate, irrespective of their orientation or relative position.

Definition 2.6.5. *Major transmission range* is the maximum distance at which two nodes can communicate with favourable orientation and/or relative position. That is, when they can look at each other through their antenna bore-sight.

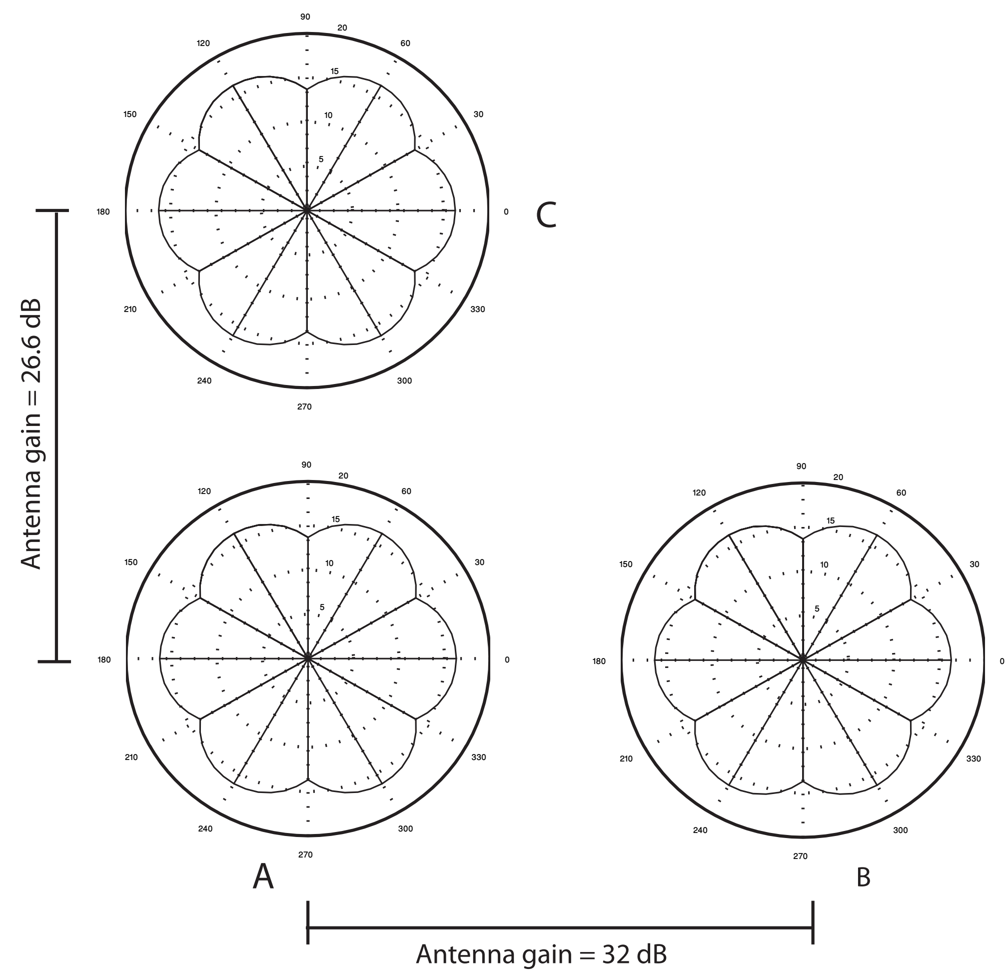


Figure 2.6: Difference in antenna gain as a result of gain variation in the major lobe. Node A and B are tuned, so are A and C. However the antenna gain between A and B is 6 dB larger than the antenna gain between A and C, because A and B look at each other through bore-sight while A and C look through the 3 dB angle.

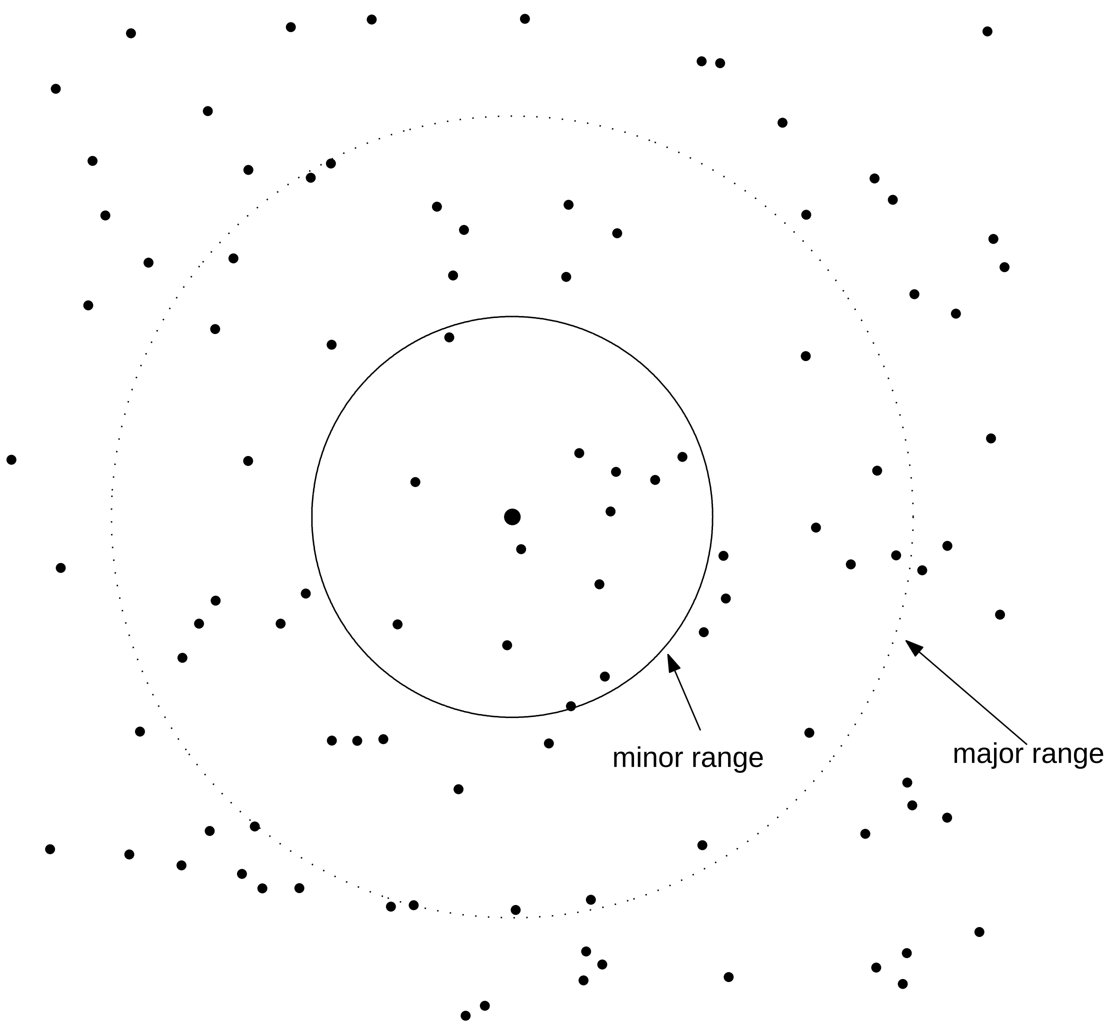


Figure 2.7: Minor and major transmission ranges. Nodes within the minor transmission range have at least one way to tune to the centre node such that communication is possible. For the nodes in the region between minor and major ranges, ability to communicate is probabilistic.

The use of directional transmission and reception provides the system with a higher gain than omnidirectional mode, and thus allows the transmission radius to be much longer as illustrated in example 2.6.1.

Example 2.6.1. Consider the system with a OFDM transceiver (5.8 GHz) with transmit power 13 dBm, and receive sensitivity of -88 dBm. The transmit and receive antennas have a gain of 16 dBi. We consider the Friis free space model [1] for calculating path loss. Distance d is is km, frequency f is in MHz, and antenna gains are in dB. The system should maintain a SNR of 10 dB for proper operation.

(a) Path Loss at distance d km,

$$\begin{aligned} L_f &= 32.44 - G_T - G_R + 20\ log\ f + 20\ log\ d \quad dB \\ &= 32.44 - 16 - 16 + 20\ log\ 5800 + 20\ log\ d \\ &= 75.71 + 20\ log\ d \end{aligned}$$

(b) Maximum acceptable path loss,

$$\begin{aligned} L_{max} &= TxPower - RxSensitivity - SNR \\ &= 13 - (-88) - 10 \\ &= 91\ dB \end{aligned}$$

(c) Maximum separation,

$$\begin{aligned} 20\ log\ d &> 91 - 75.51 \\ d &\approx 5.8 \end{aligned}$$

If we used omnidirectional antennas with a typical gain of 12 dBi. The transmission range obtainable is less than 3.3 km.

Definition 2.6.6. A *transmission schedule* is a schedule that specifies for each time slot the triplet $< node, antenna, mode >$. Given this schedule, each node in the network knows whether to transmit or to be in receive state, and which antenna to select.

2.6.2 Notations

Commonly used symbols and notations in the thesis are listed in Table 2.2

n	The number of nodes in the network.
k	The number of antenna beams in the antenna system.
id	A unique ID available to each node in the network.
t	The time slot number.
A	Antenna index, each antenna/beam has an index from $\{0, 1, 2, \ldots, k-1\}$, assigned clockwise.
$t_{NbrStart}$	The time slot in which neighbour discovery is started.
$t_{TpBcastStart}$	The time slot in which topology broadcast is started.
β	SNR ratio required for successful reception of transmitted signal by receiver.
$G(\mathcal{V}, \mathcal{E})$	The graph representation of the network.
$\mathcal{V}$	The set of vertices in the graph. Used when the network is represented as a graph. Each vertex represents a node in the network.
$\mathcal{E}$	The set of edges in the graph. Edges are directional. Represents a link from one node to another.
$SINR_{ij}$	is the SINR at node j when trying to receive signal transmitted by node i.

Table 2.2: Commonly used symbols and notations

2.6.3 Assumptions

The major assumptions made in this thesis which impact simulations and analysis are:

- Static network: We do not consider mobile networks. Our target network is a static maritime network.
- Homogeneous nodes: all nodes have same capability in terms of memory buffers, radio capability, etc.
- Flat terrain: All nodes are at the same altitude. Thus, we only consider azimuth pattern in deciding the antenna gain.

2.7 Summary

In this chapter we presented the work that forms the background of this thesis. We started the chapter by considering the various antenna technologies available and their potential use. We then moved on to present the issue and solutions pertaining to use of directional antennas in wireless mesh networks. Much of the work using directional antenna is geared towards solving the problem of using directional antenna with IEEE 802.11 networks. The main problems facing the use of directional antennas in mesh networks are neighbour discovery and link scheduling. We then presented the work on STDMA which is used as the MAC protocol for Achilles. We then considered the prior work in mesh networks using purely directional antennas. We could not find an existing solution that would fit the design goals of a purely directional antenna system and a contention-free MAC, that provides a complete system solution. Our background study showed that a *link schedule* based STDMA MAC best suits Achille's design goals. The chapter also presented the node and antenna design for Achilles, along with the assumptions in the design of Achilles.

Chapter 3

Neighbourhood Discovery

3.1 Overview

In this section, we describe our proposed algorithm for neighbourhood discovery. Our algorithm ensures that all the nodes in the network will discover their neighbours in a deterministic time. At the end of a fixed time period, the neighbourhood discovery phase will terminate. This feature is important in the start-up phase of the network, when nodes do not have a transmission schedule yet and depend on an alternate method to set up the link with neighbours. The algorithm works as follows:

- At the start of the neighbourhood discovery phase, which occurs at a pre-determined time (wired in), all nodes in the network wake-up and enter the neighbourhood discovery phase. The start time is chosen such that all the nodes in the network are switched on (which can be determined when deploying the system). The exact time is not important.
- Each node has a unique ID and has the knowledge of the number of nodes in the network (n), both of which are programmed at the time of deployment.
- Depending on its ID, a node can determine whether to be in passive scan mode, or in the active transmit mode (described later)
- If the node is in the passive scan mode, it switches between its antennas at a predetermined rate in the clockwise direction.
- When a node receives a hello message from a neighbouring node, it notes down the neighbour's ID and location, and maintains a neighbour table.

- Each node in the network is assured a period to be in active transmit mode, and is also assured that all the other nodes in the network would be in passive scan mode when it is active, thus preventing collisions.
- The neighbourhood discovery phase lasts for nk^2 time slots, at the end of which all nodes in the network would have discovered all their neighbours.

3.2 Random Discovery

Before we discuss the algorithm in detail, let us consider the general neighbourhood discovery problem. In a self-configuring network, nodes do not have a priori knowledge of their neighbours. In addition to communicating with neighbours, nodes depend on their neighbours to relay packets on their behalf. Thus, the discovery of neighbours is one of the first tasks that a self-configuring multihop network needs to perform. Neighbour discovery is typically done by listening for *hello* packets. Nodes periodically advertise their presence by broadcasting a short message known as a *hello* packet. Any neighbour that hears the *hello* packet is thus made aware of the presence of the advertising node. If all nodes in the network use omnidirectional antennas, then a single *hello* packet transmitted by a node makes all the nodes in the neighbourhood aware of the node's presence (collisions and packet loses make it a bit more complicated). If nodes use directional antennas then an advertisement (by means of hello) is successful only if the neighbour(s) are tuned to the transmitter. In the absence of some form of scheduling, the random discovery is an option available to nodes. In the next few paragraphs, we will look at the probabilities associated with random discovery.

Definition 3.2.1. A node A is said to discover another node B when A successfully receives a special kind of broadcast packet from B which contains the hello message. The first time such a packet is received is the instant of discovery.

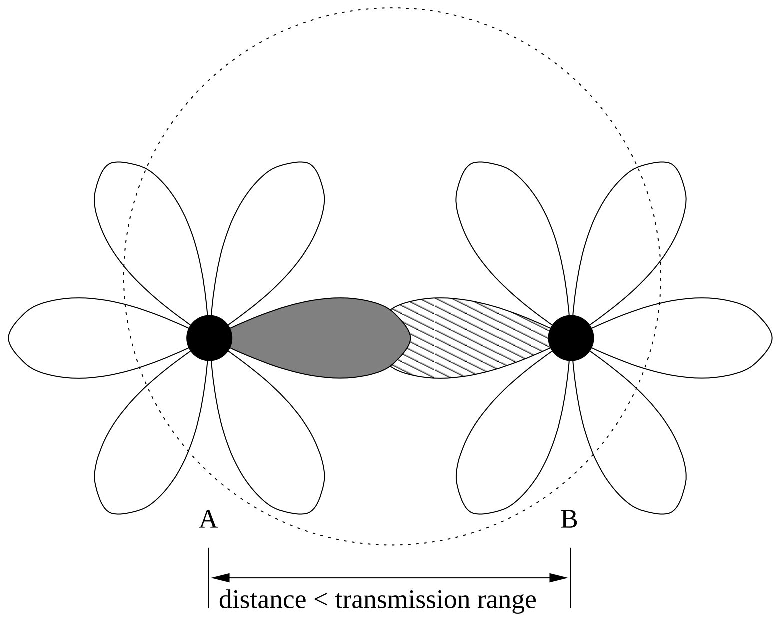


Figure 3.1: Two neighbouring nodes

Consider two nodes in a plane, c.f. Figure 3.1, placed randomly in a circle of diameter equal to the transmission radius. Each node has a random orientation. To be able to communicate the nodes must be tuned to each other. However, in the neighbourhood discovery phase, the nodes are not aware of the presence of one another, and don't have a transmission schedule. For node B to successfully receive node A's hello packet in a particular time slot, the following conditions must be met:

- Node A selects the antenna that points towards node B
- Node B selects the antenna that points towards node A
- Node A transmits the hello packet
- Node B is in receive mode

Let p be the probability that a node transmits in a given slot, and q the probability that node is in receiving mode $q = 1 - p$. The probability of a particular antenna being selected is $\frac{1}{k}$. Thus the probability that node B would discover node A in a particular slot (d) is given by:

$$d = \frac{1}{k} p \frac{1}{k} q \tag{3.1}$$

Equation 3.1 presents the probability of discovery in a particular slot where there are just two nodes. However, if there are other nodes whose transmission range cover B, then an additional condition is required to take into account the interference from those neighbours of B. Then, the following additional condition is required for successful discovery:

- None of the other neighbours of B should be transmitting in the direction of B.

Assuming that a node has m neighbours on average, and that the neighbours are distributed uniformly in all directions, the probability of successful discovery in a particular time slot is thus given by:

$$d = \frac{1}{k}p\frac{1}{k}q(1-\frac{p}{k})^{\frac{m}{k}-1} \tag{3.2}$$

Equation 3.2 is calculated by multiplying the probability of successful reception with the probability that none of the other nodes in the selected antenna direction of the receiver is transmitting. $\frac{p}{k}$ is the probability that a node is transmitting in a particular direction. $\frac{m}{k}$ is the average number of nodes in the receiver's currently selected antenna direction. For there to be no interference, none of the other nodes in the receiver's selected antenna sector should be transmitting towards the receiver.

In calculating the above, we made the simplifying assumption that only nodes that are neighbours interfere with the transmission. This is not true in general. The interference range[1] is almost always larger than the transmission range, and so nodes that are not neighbours of the receiver can still result in interference at the receiver.

From Equation 3.2 we can see that the probability of discovery in a given time slot depends on p in addition to other system and network parameters such as number of antennas and density. The selection of optimal p is treated in [24] for steerable antennas. In Figure 3.2, we present the plots of d vs p for various neighbour densities (number of neighbours per antenna sector). It shows that at moderate neighbour densities $p = \frac{1}{2}$ is a fair choice. At high neighbour densities, collisions from neighbours increases and thus nodes need to reduce their transmission of hello messages to avoid excessive collisions. We are interested in the question: how many slots should a node be in the neighbour

[1]The maximum distance at which a transmitted signal can cause interference and collision at a receiver but the signal itself is not sufficiently strong enough to be decoded

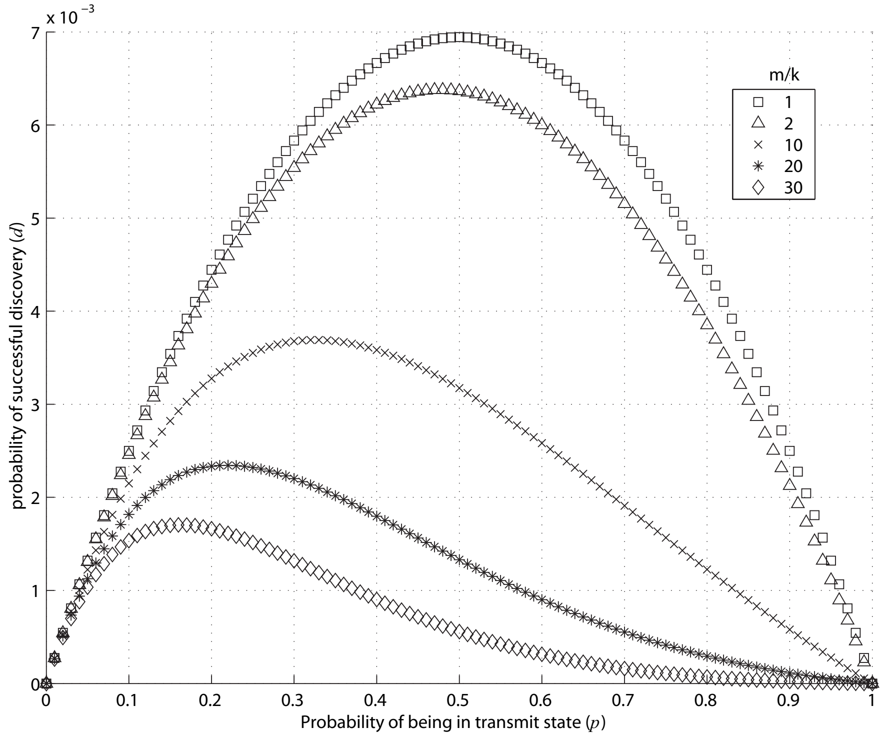


Figure 3.2: Probability of discovery for various transmission probabilities. Note that as the neighbour per antenna sector (m/k) increases, the optimal transmission probability decreases. (analytical)

discovery phase such that it is assured with a high probability (say 99%) that it has discovered a particular neighbour? The probability that discovery is successful in α tries is given by:

$$d_\alpha = 1 - (1 - d)^\alpha \tag{3.3}$$

Assuming that the discovery of neighbours is independent of one another, the probability that a node discovers exactly w of its neighbours in α tries is given by:

$$P_{w,\alpha} = \binom{m}{w} d_\alpha^w (1 - d_\alpha)^{m-w} \tag{3.4}$$

And that it discovers w or more ($w \leq m$) of its neighbours in α tries is given by:

$$P_{w,\alpha} = \sum_{i=w}^{i=m} \binom{m}{i} d_\alpha^i (1 - d_\alpha)^{m-i} \tag{3.5}$$

Example 3.2.1. Let us look at an example to get a feel for the above equations. We consider a network with 100 nodes distributed in a square area of dimensions 30 km $\times$ 30 km. Each node has six 60^o beamwidth antennas ($k = 6$). The average number of neighbours $m = 14$. Nodes transmit and receive with equal probability $p = q = \frac{1}{2}$.

(a) From Equation 3.2, the probability of discovering a particular neighbour in a given time slot is:

$$d = \frac{1}{12}\frac{1}{12}(1 - \frac{1}{12})^{\frac{14}{6}-1}$$
$$\approx 0.00618$$

(b) Number of tries required for 99% success can be caluclated from Equation 3.3:

$$0.99 = 1 - (1 - 0.00618)^{\alpha}$$
$$\alpha = \frac{log(1 - 0.99)}{log(1 - 0.00618)}$$
$$\approx 743$$

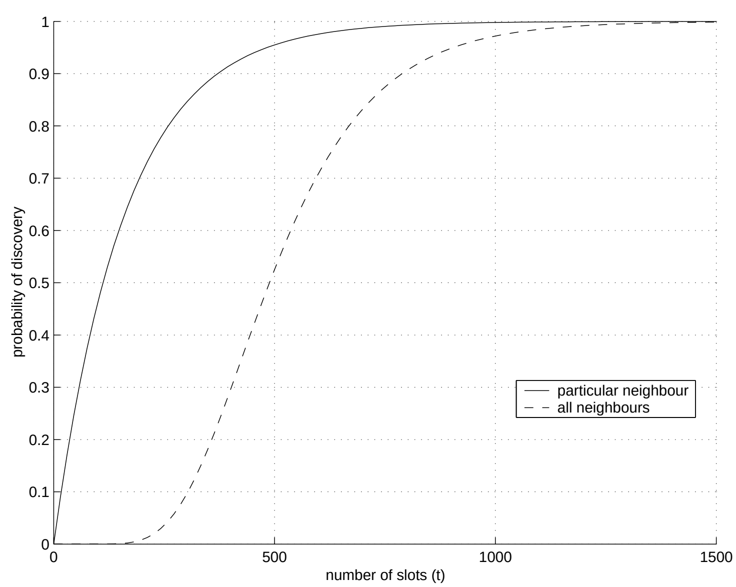


Figure 3.3: Probability of neighbour discovery with α tries (analytical)

Figure 3.3 shows the plot of the elapsed time (number of slots) vs. the probability of discovery of a particular node, and of the complete neighbourhood. It can be seen that in about a 1000 time slots, the complete neighbourhood is discovered with very high probability.

Parameter	Value
Terrain dimension	30 km × 30 km
Number of nodes	100
Node placement	Random
Time slot duration	8 ms
Number of slots simulated	10,000
Average node density	15
Transmission probability	0.5
Radio bandwidth	6 Mbps
Transmission range minor	4.4 km
Transmission range minor	8.8 km
Radio required SNR	10 dB

Table 3.1: Simulation parameters

Next, we look at the results from simulations of the same network. We implemented the random discovery protocol for the Qualnet Simulator. Figure 3.4 shows the position of the nodes. Table 3.1 lists the parameters used in the simulations. In addition the following hold true:

- Free-space radio propagation model is used.
- The network considered is a static network. Nodes are placed randomly in the square of size 30 km × 30 km. Node 1 is placed at the centre of the square.
- The simulation runs for 10,000 time slots.

From the simulation, we find out the statistics of the time required for neighbourhood discovery. We run the simulations with 100 different seeds. First we focus on node 1, which is at the centre of the square. We want to look at how node 1 discovers its neighbours. We choose four different neighbours (nodes 30, 14, 66 and 83) with increasing distance from node 1. Figure 3.4 shows the position of the nodes in the square, as well as the neighbours of interest. In Figure 3.5, we plot the probability of discovery of each of the four chosen neighbours with increasing number of time slots.

Some of the immediate observations from the simulation results are:

- The number of slots required for neighbour discovery with high probability (99%) is much higher than the analytical value in all cases except when the neighbour is very close. The analytical value of approximately 743 slots for 99% discovery probability is close for the neighbours 30 and 14 (0.35 and 1.43 km resp.), however,

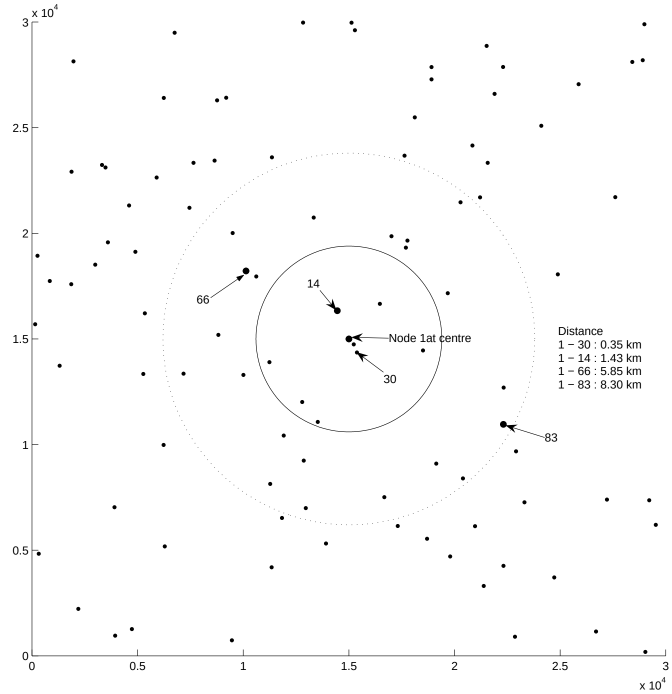


Figure 3.4: 100 nodes scattered in a 30 km × 30 km square. Node 1 is at the centre of the square. The inner circle is the minor transmission radius and the outer circle is the major transmission radius.

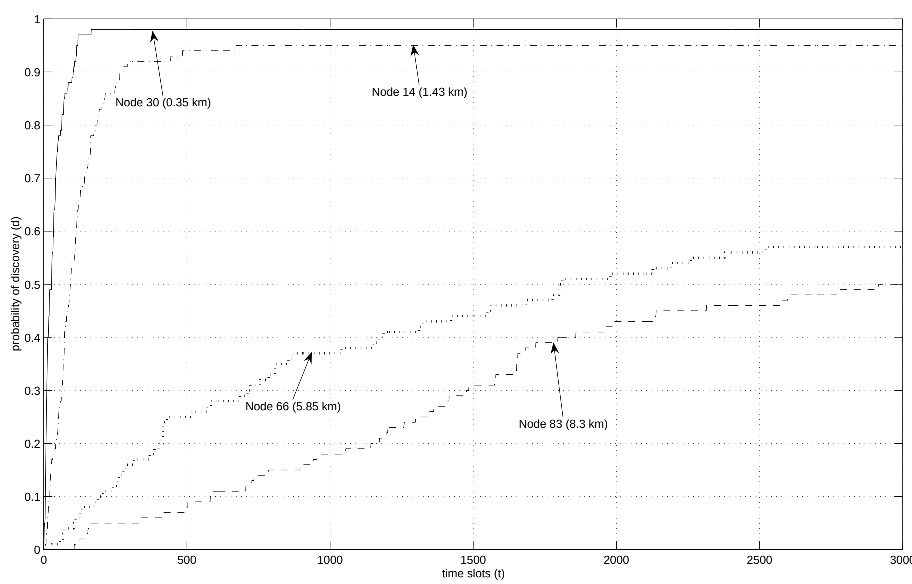


Figure 3.5: Probability of discovery of neighbours at increasing distance. Note that each of the curves should asymptotically approach 1 for sufficiently large number of time slots. However we plot only up to 3000 slots so that the trend of the top two curves is visible.

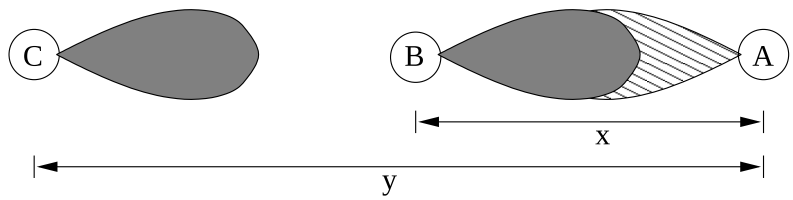


Figure 3.6: Interference from distant nodes. Even though node C can not communicate with node A, it can interfere with the reception at node A.

for neighbours 66 and 83, the probability of discovery is less than 0.6 even for 3000 time slots.

- For any given number of time slots, the probability of discovery decreases monotonically with increasing distance between the nodes.

The reason for the discrepancy of the simulation results from the analytical results is the interference observed from nodes beyond the communication range. The analytical results considered a graph model with binary relationship where as long as the neighbours of the receiver did not transmit in the direction of receiver, communication was successful. However, in the simulation, a realistic radio model is used. The model takes into account the interference from all nodes, including those that are beyond communication range.

The radio model considered requires a Signal-to-Noise Ratio (SNR) of 10 dB to be able to successfully decode a signal. Consider three nodes A, B, and C placed in a line as shown in Figure 3.6. Node C and node B transmit a packet at the same time. Node A can successfully receive the packet if the SNR of the signal from Node B is above a threshold ($\beta = 10\ dB$). Now, the transmission from node C acts as noise to the receiver A, and thus raises the noise floor.

In general, a receiver can successfully decode the transmission from a node k if the condition in Equation 3.6 is satisfied. P_k is the power of the signal received from node k and N is the thermal noise power[2].

$$\frac{P_k}{N + \sum_{i=0, i\neq k}^{n-1} P_i} > \beta \tag{3.6}$$

Since all the nodes transmit at the same power and follow the same path loss model,

[2] $N = kTB$, where k is *Boltzmann's constant* given by 1.38×10^{-23} Joules/Kelvin, T is the ambient temperature in Kelvin, and B is the effective bandwidth of the receiver in Hz.

whether or not node A can receive the transmission from node B depends on the distance ratio of the distance between node A–B and A–C. Recalling *Friis free-space path loss model* $(L)f = 32.4 - G_T - G_R + 20log(d) + 20log(f))$, and noting that the SNR at receiver A, would depend only on the distance, we can say that the following condition must be met for successful reception:

$$20(log(y) - log(x)) > \beta$$
$$\frac{y}{x} > 10^{\beta/20}$$

For $\beta = 10\ dB$, this gives us

$$\frac{y}{x} > 3.162 \tag{3.7}$$

Equation 3.7 implies is that nodes up to three times the distance to the neighbour that fall in a sector formed by the directional antenna (interference sector) can interfere with the *hello* message from the neighbour. Therefore, the further the neighbour is, the more susceptible it is to interference. This increased number of interferers with increasing distance is the reason for the decrease in probability of discovery. In Figure 3.7, we can see that the transmission from node 14 is interfered by far fewer nodes than the transmission from node 66. As the number of interfering nodes increases, *hello* packets are lost due to interference, and thus many more tries are required (on an average) for successful neighbour discovery.

3.3 Deterministic Discovery

We no present our proposed algorithm to enable complete neighbourhood discovery within a predetermined time. To enable this, we need to i) minimize the collisions and interference, ii) modify the antenna switching so that nodes can send their *hello* packets to all their neighbours within a fixed number of slots.

To avoid collisions (first requirement), we need some form of preliminary schedule. As neighbourhood discovery forms the very first part of network boot-up process, only the simplest and primitive form of synchronization mechanism is desirable and practical. For this, we use a value $t_{NbrStart}$ as the time slot when the neighbourhood discovery process should start. When nodes are deployed in the network, this time can be programmed

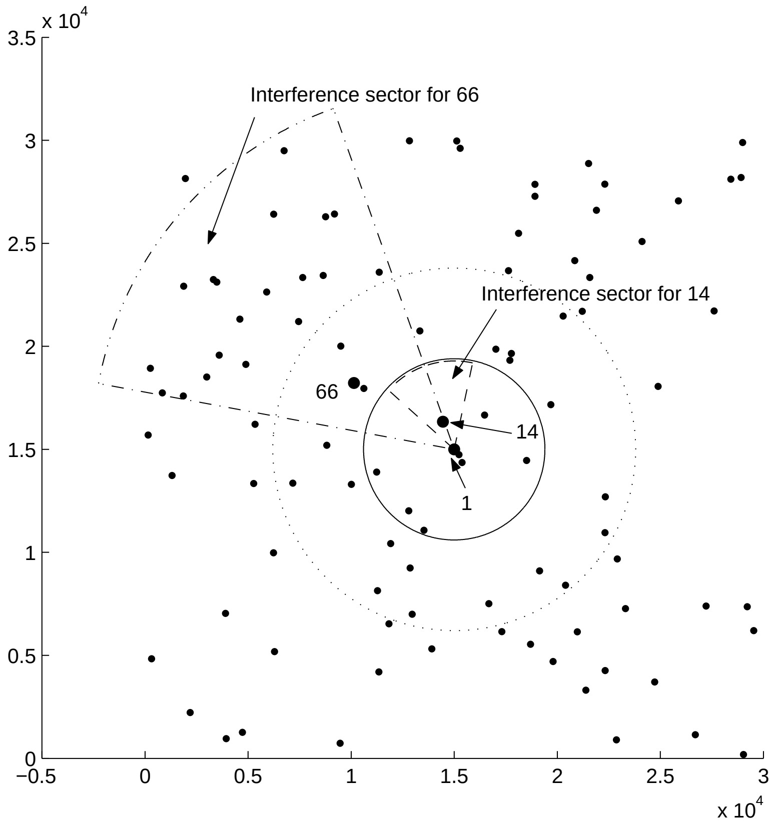


Figure 3.7: Candidate interfering nodes for the transmission from node 14 and node 66 to node 1. The greater the distance from the destination node, the larger is the interference sector.

into the nodes. As an example, $t_{NbrStart}$ could be the commissioning time of the network. When the network is operational and new nodes are to be added to the network, then a new $t_{NbrStart}$ could be communicated to the existing nodes using the available network facilities; this new $t_{NbrStart}$ will be used by the existing nodes and new nodes to start the next neighbourhood discovery phase. The other requirement is an unique ID for each node in the network from $\{0, 1, 2, \ldots, n-1\}$.

To satisfy the second requirement, we propose a new antenna switching algorithm that ensures that the *hello* packet from a node will be received by all its neighbours within a fixed number of time slots. The algorithm has two distinct ways of switching the antenna depending on the state of the node. As mentioned before, during the neighbourhood discovery phase, nodes are either in *active transmit mode* or in *passive scan mode*. In the passive scan mode, at the beginning of each time slot, the node switches to the next antenna in a clockwise direction and listens for hello packets, therefore, in k time slots a node would have listened to all the possible directions. In the active transmit mode, the node switches the antenna, also in clockwise direction, but only every k time slots. The active transmit state lasts for k^2 time slots (where k is the number of antennas) at the end of which a node would have discovered all its neighbours. Since the transmitting node transmits for k consecutive time slot in each direction, any neighbour in the currently active antenna sector will surely hear the *hello* packet within k time slots. Figure 3.8 shows an illustration of the antenna switching scheme.

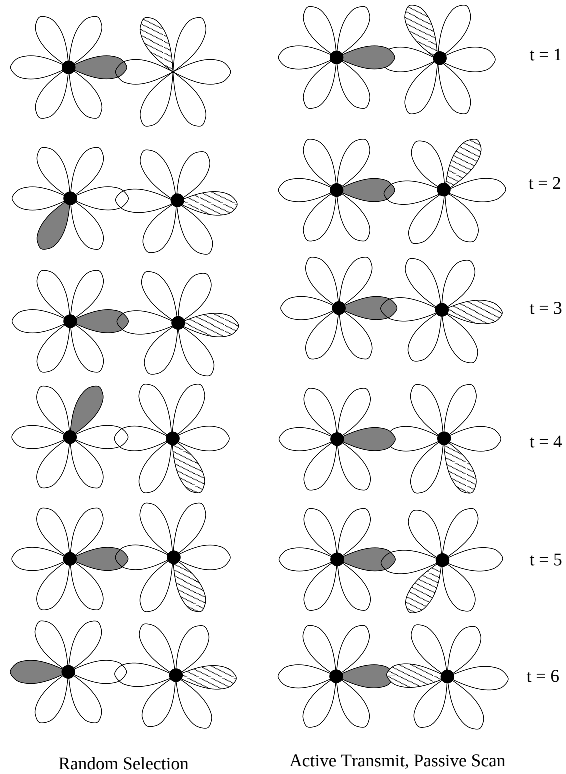


Figure 3.8: Antenna switching. The left column shows random switching, the right column shows switching with active transmit and passive scan. In the Active Transmit Passive Scan the transmitter keeps the same antenna selected for the period and the receiver switches the antenna clockwise every time slot.

Algorithm 1 Antenna switching algorithm SWITCH_ANTENNA_N (neighbourhood discovery)

Require: The node is in neighbourhood discovery phase

1: **procedure** SWITCH_ANTENNA_N(*state*, *A*, *aslotcount*) ▷ aslotcount is the current time slot number
2:     **if** $state = activetransmit$ **then**
3:         **if** aslotcount mod $k = 0$ **then**
4:             $A \leftarrow (A+1) \bmod k$ ▷ A is the antenna index, k is the number of antennas
5:         **end if**
6:     **else if** $state = passivescan$ **then**
7:         $A \leftarrow (A + 1) \bmod k$
8:     **end if**
9: **end procedure**

Algorithm 1 presents the antenna switching procedure executed during the neighbourhood discovery phase. The procedure determines which antenna to select based on the current state. A is the index of the currently selected antenna, modifying A automatically triggers the antenna switching circuitry. The procedure is called by the *neighbour_discovery* procedure, towards the beginning of a time slot. Lines 2–5 ensure that if the node is in *activetransmit* state then the antenna is switched only every k

slots. Thus the node sends *hello* packet in each direction for k time slots. Lines 6–8 control the switching of the antenna when the node is in *passivescan* mode; the next antenna in clockwise direction is selected.

Algorithm 2 Neighbour discovery

1: **procedure** NEIGHBOUR_DISCOVERY(id, n, $t_{NbrStart}$, k, A))
2:     **if** $t = id \times k^2 + t_{NbrStart}$ **then** ▷ check if it is my time to start active scan
        ▷ t is the current slot number, incremented by one every time slot
3:         $state = activetransmit$
4:         $aslotcount \leftarrow 0$ ▷ number of time slots that the node has been active
5:     **end if**
6:     **if** $aslotcount \geq k^2$ **then**
7:         $state = passivescan$
8:     **end if**
9:     **if** $state = activetransmit$ and $aslotcount < k^2$ **then**
10:         call SWITCH_ANTENNA($state$, A, $aslotcount$)
11:         transmit hello
12:         $aslotcount \leftarrow aslotcount + 1$
13:     **else if** $state = passivescan$ **then**
14:         call SWITCH_ANTENNA($state$, A)
15:         listen for hello
16:     **end if**
17: **end procedure**

The neighbour discovery algorithm, Algorithm 2, becomes active at time $t_{NbrStart}$ and is executed at the beginning of each time slot during the neighbour discovery phase by every node. Each node in the network is assured i) a period to send its *hello* packets and ii) that all the other nodes in the network would not interfere. Each node is assigned a k^2 time slot period, during which it broadcasts its *hello* packets. The time at which a node becomes active is easily determined by the node by using its ID and is given by $id \times k^2 + t_{NbrStart}$. It can be seen that the unique ID assumption ensures that each node would go into the *activetransmit* state at a unique and non-overlapping period. Line 2–4 check whether it is time for the node to get into *activetransmit* state, and initialise the slot counter if it is. Lines 6–8 ensure that the node exits from the *activetransmit* state after exactly k^2 time slots of being in *activetransmit* state. In lines 9–12, *hello* message is transmitted by a node in *activetransmit* state, incrementing the slot counter that tracks the number of slots of activity. Lines 13–15 keep the transceiver in the receiving mode, if the node is in *passivescan* state.

Example 3.3.1. Consider a network consisting of four nodes with ID from 0 to 3, as shown in Figure 3.9. The lines joining the nodes show that the nodes are within

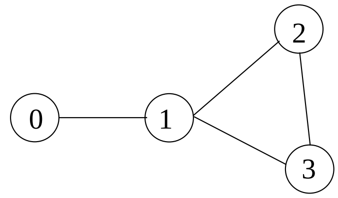


Figure 3.9: A simple network

transmission range of each other. Let t_0 be the slot in which neighbour discovery is programmed to begin. Each node has a six beam antenna system and therefore, the Active Transmit state lasts for 36 time slots ($k^2 = 36$). The following sequence of events take place:

$t = t_0$: Node 0 begins Active Transmit, and nodes 1,2 and 3 are in Passive Scan state.

$t = t_0 + 36$: Node 1 would have heard the *hello* from node 0. Node 0 node returns to Passive Scan state and node 1 enters Active Transmit state.

$t = t_0 + 72$: Node 0, 2 and 3 would have heard the *hello* from node 1. Node 1 now returns to Passive Scan state and node 2 enters Active Transmit state.

$t = t_0 + 108$: Node 1 and 3 would have heard the *hello* from node 2. Node 2 now returns to Passive Scan state and node 3 enters Active Transmit state.

$t = t_0 + 144$: Node 1 and 2 would have heard the *hello* from node 3. Node 3 now returns to Passive Scan state. The neighbour discovery phase ends at this time.

3.3.1 Implementation Details

The neighbourhood discovery protocol is used to create and maintain the Neighbour Table. The format of the *hello* packets is shown in Figure 3.10. The message exchange takes place over UDP, and all messages are sent to a port PSTDMAPORT.

type	length	id	timestamp	latitude	longitude	altitude

Figure 3.10: Structure of hello packet (not to scale)

Hello packets have the following fields:

type : Type of packet, set to HELLO. Allows the protocol to distinguish messages of different types.

length : The length of the packet in bytes (size = 1 byte).

id : The ID of the node sending the packet (4 bytes).

timestamp : The timestamp when packet was created. The timestamp value is the time since epoch (00:00:00 UTC, January 1, 1970), measured in milli seconds. (8 bytes).

latitude : The latitude in thousandths of a second of arc. 2^{31} represents the equator; numbers above that are north latitude (4 bytes).

longitude : The longitude in thousandths of a second of arc. 2^{31} represents the prime meridian; numbers above that are east longitude (4 bytes).

altitude : The altitude above mean sea level measured in centimeters (4 bytes). At present, this field is not used, but kept for possible use in future.

Each node maintains a neighbour table which keeps a record of its neighbours, their locations as well as the index of the antenna from which the *hello* packet was received. It may appear that the location fields maintained in the neighbour table are redundant as the antenna index is sufficient to determine the antenna to use to communicate with the neighbour. However, due to the peculiarities of our network deployment (nodes on floating buoys), due to wave motion and wind, the orientation of nodes may change due to rotation or lateral movement. The node then needs to recalculate which antenna to use to communicate with its neighbour and the location information is required for this. A node may receive multiple *hello* packets from a neighbour, the latest message (determined by comparing the timestamp) is used to populate the neighbour table.

Id	Timestamp	Latitude	Longitude	Altitude	AntennaIndex
10	...	...	...	...	2
4	...	...	...	...	3
7	...	...	...	...	0

Table 3.2: Example neighbour table

New entries in the neighbour table are only added during the neighbour discovery phase. Between two neighbour discovery phases, no entries are deleted from the neighbour table. Only at the initiation of a new neighbour discovery phase is the neighbour table cleared. Node failures, if any, are handled by the routing protocol, by using alternate routes to destinations.

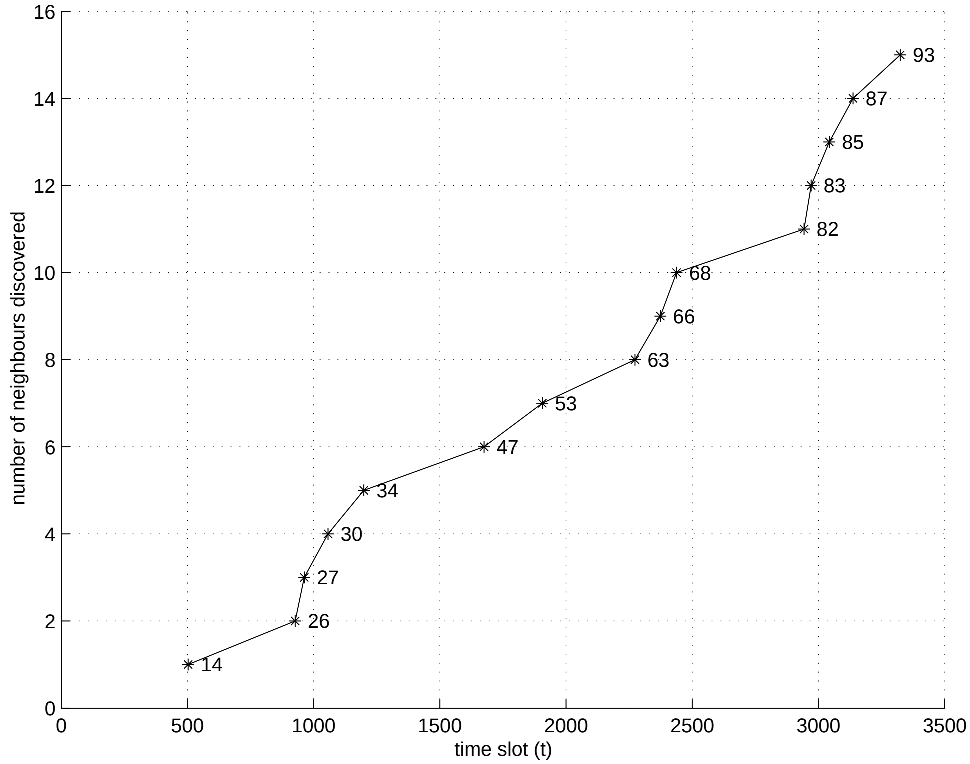


Figure 3.11: Neighbour discovery in a deterministic way. Node 1 discovers all 15 of its neighbours in a predetermined time. Note also the ascending order in which nodes are discovered due to scheduling based on node ID.

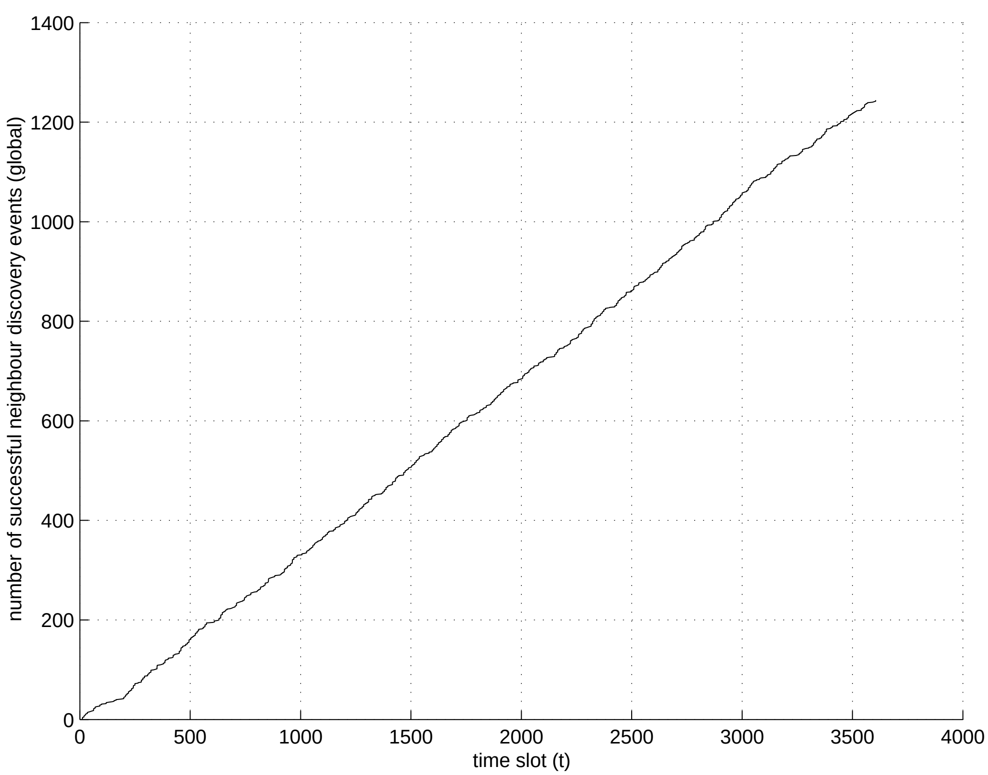


Figure 3.12: Plot showing the number of successful neighbour discovery events in the network with time. Note the linear growth as nodes discover their neighbours and the neighbour table build up.

Example 3.3.2. We consider the same network as in example 3.2.1, and simulate deterministic neighbour discovery. Recall that node 1 has 15 neighbours and if there are no collisions, then all the neighbours should be discovered within $k^2n = 3600$ time slots. Figure 3.11 shows the time slot in which each neighbour of node 1 was discovered. Some of the immediate observations from the simulation are:

- All the neighbours of a node are discovered within a fixed time.
- The order in which neighbours are discovered depend on their ID. Neighbours are discovered in ascending order which is a consequence of the node ID based scheduling.
- The number of successful discovery events grows linearly with time. This is due to the fact that at any given time only one node is actively transmitting *hello* packets, and thus the rate of growth is limited by the number of neighbours of the active node.
- The neighbour discovery phase finishes within a predetermined time, in this case, 3600 time slots after it started.

3.4 Conclusion

In this chapter, we considered the problem of neighbour discovery in a network where nodes used directional antennas. The main problem is that nodes do not have a schedule and thus do not know how to co-ordinate their antenna switching. We looked at the random discovery scheme in which nodes point their antennas in random directions and discover neighbours in a probabilistic manner. We saw that the random discovery scheme suffers from interference and may fail to discover neighbours even after several tries. To alleviate the problem, we presented a deterministic neighbour discovery scheme suitable for static wireless networks, especially those deployed to form a backbone network, as is our driving scenario. Nodes can perform the neighbour discovery at regular programmed intervals (e.g. midnight) or any other time as determined by the network administrator. Nodes are assured that they will discover all their neighbors. A question that we did not address is what happens when there are packet losses in the network? An answer to this is that each time slot of 8 *ms* is long enough to send multiple *hello* packets and thus is a possible error recovery strategy. Another method is to perform the neighbour discovery multiple times.

Chapter 4

Topology Broadcast

4.1 Topology Broadcast

The next major phase in network set up is the topology broadcast phase. Once the neighbour discovery phase has completed, all nodes in the network are aware of their own neighbours, each node now needs to inform the rest of the nodes in the network about their neighbour set, so that a topology map of the whole network can be constructed at each node. The topology map is required to derive the link schedules. To allow the topology to be broadcasted network-wide, a process akin to simple flooding is required. To enable this, we propose a scheme which uses node TDMA. The key features of the algorithm are as follows:

- Each node in the network is assigned k contiguous time slots (*k-frame*) in the TDMA frame of size kn, where n is the number of nodes in the network.

- During its assigned time slots, the *active node* can broadcast any queued packets that it has.

- The node broadcasts the same packet k times, selecting a different antenna for each successive packet. Node does not rebroadcast a packet that it has seen before.

- All the other nodes in the network that have the currently active node in their neighbour table, tune their antenna to the active node.

- Thus, after broadcasting the packet for k time slots, the active node is assured that all of its neighbours would have received the packet.

The algorithm provides the basic broadcast mechanism that is used for topology dissemination.

4.1.1 Forming the Global Topology Map

Each node forms the global topology map by collecting broadcasts from other nodes that contain information about their neighbour set. This information is carried in a special packet called *nbrinfo* packet. Each node begins with a $n \times n$ matrix referred to as *topology matrix* in which all the elements are zero except the diagonal elements which are 1. The value i, j of the topology matrix represents the connectivity of node i to node j. The size of the topology matrix is n^2. The topology matrix is constructed by each node from the $nbrinfo$ packets it has received, is not transmitted directly. The size of the $nbrinfo$ packet depends on the number of neighbours that a node has. The size of the $nbrinfo$ packet is limited by the MTU (maximum transmission unit) of the link. With a typical MTU of 1500 bytes, information about more than 300 neighbours can be transmitted.

The first task the node does is to update the *topology matrix* with information from its own neighbour table. Next, the node collects *nbrinfo* packets and progressively updates the matrix. Finally, when *nbrinfo* packets from all thea nodes in the network are received, the *topology matrix* represents the global topology. Figure 4.2 shows the process of how node 1 build up the topology matrix by updating the initial matrix when it receives the *nbrinfo* packets from the other nodes in the network.

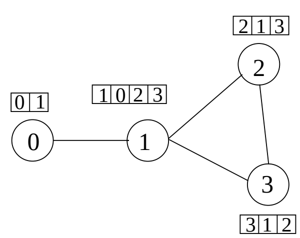
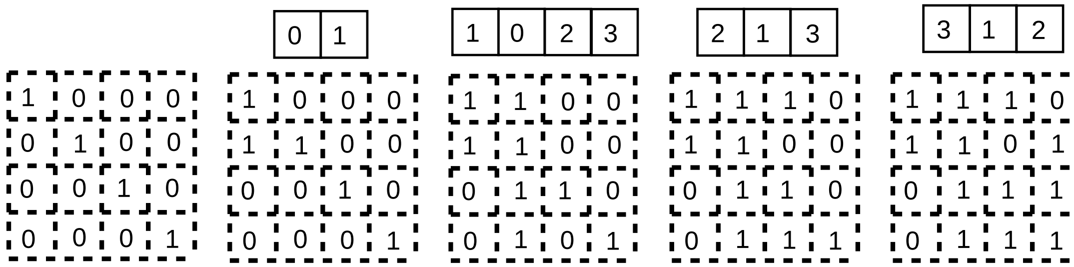


Figure 4.1: Neighbour Information ($nbrinfo$) packets broadcasted by each node in the network. Node forms the complete network topology matrix by collecting these packets from all the other nodes in the network.

As long as each node in the network successfully broadcasts its *nbrinfo* packet to all the other nodes in the network, the final topology map created by every node will be the same consistent network topology. In the event that some *nbrinfo* packets are lost, there could be inconsistencies in the topology map generated by different nodes.



(a) initial (b) from 0 (c) from 1 (d) from 2 (e) from 3

Figure 4.2: The topology matrix is updated as node 0 receives receives *nbrinfo* packets from other nodes in the network. Each node in the network follows the same procedure to end up with a consistent topology map of the entire network.

To tackle this, the *nbrinfo* packet can be transmitted multiple times.

4.2 Broadcast Algorithm

Similar to the neighbour discovery problem, the issue now too is that there is no schedule present in the network to ensure that neighbours can receive the broadcast packet (and subsequently forward it) when a node sends the packet. While the nodes know who their neighbours are, they do not have any schedule to tune to their neighbours. A possible solution here too is to do a random broadcast. However, as we found out in the neighbour discovery section, random broadcast is highly unreliable, and does not guarantee that all the broadcasted packet will be received by all the nodes in the network. To solve this, as before, we form a global schedule based on a start time ($t_{BcastStart}$) and the node IDs. There are at least two ways to specify $t_{BcastStart}$ time. It could be programmed into the nodes (for the very first time when network is set up, i.e. network commissioning) or if the network is already operational, then the nodes can a priori communicate and agree on a suitable value. Another way is to calculate it from the $t_{NbrStart}$ by noting that the broadcast phase starts at the end of the neighbour discovery phase. Thus $t_{BcastStart} = t_{NbrStart} + \nu k^2 n + \epsilon$, where ν is the number of times the neighbour discovery phase is repeated[1] and ϵ is a settling down or guard time.

The basic idea is to give each node a chance to broadcast within a TDMA frame, and to ensure that when the node is broadcasting, all its neighbours are tuned to it, and no other node in the network is broadcasting at the same time. This is similar to

[1] To mitigate the impact of packet errors if any, the neighbour discovery phase could be repeated a number of times

a simple TDMA scheme used in a fully connected network, where every node gets at least one slot to transmit within a TDMA frame. However, there are with two main differences:

- A node transmits the same packet for k contiguous time slots.
- Node selects a different antenna for each of the k time slots.

This algorithm aims to ensure that all the nodes in the neighbourhood receives the broadcast packet. When a node is transmitting i.e. it is *active*, the algorithm ensures that all of its neighbours are tuned to it for the period. This ensures that if the *active* node transmits the same packet once using each of its antennas, then all the neighbours will receive the packet. For an illustration see figure 4.3.

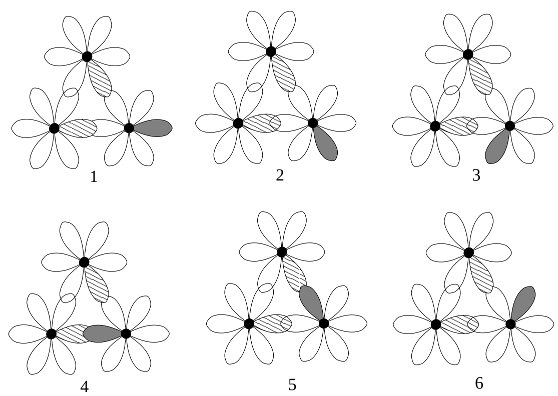


Figure 4.3: Node behaviour during the broadcast phase. The node to the bottom right is transmitting a broadcast packet. All its neighbours are tuned to it. Note the clockwise shift in the antenna selected for each time slot.

4.2.1 Broadcast Delay for a Single Packet

Let us analyse the number of frames required to broadcast a packet to the whole network. Let $\mathcal{N}_i$ be the neighbour set of node i (node with $id = i$), and let $\mathcal{N}_i^a$ be the augmented (original neighbour set plus the member i) neighbour set of node with i, that includes node i. During each TDMA frame, the broadcast packet progresses at least one hop from the originator. It can progress more than one hop away from the originator, in the same TDMA frame, if the following conditions are met:

$$(\exists j \in \mathcal{N}_i \text{ such that } j > i) \wedge (\exists k \in N_j \text{ such that } k \notin \mathcal{N}_i^a)$$

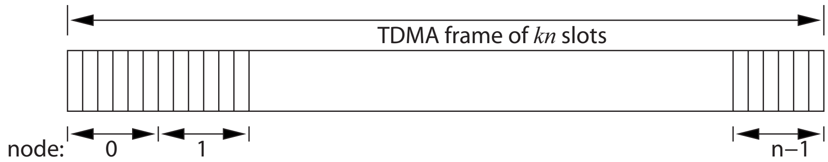


Figure 4.4: TDMA frame during the broadcast phase. Each node has k contiguous slots assigned to it. The slots assigned depend on the node ID. Node 0 has the first k slots, node 1 the next k slots, and so forth.

We can apply the above condition repeatedly to determine the maximum distance of propagation of the broadcast packet in one TDMA frame. The random distribution of the nodes, and the fact that neighbourhood of two nodes are not independent of each other, complicate the analysis. We therefore provide the upper bound on the number of frames required, and look at the propagation by means of simulations. The maximum number of TDMA frames required to flood a packet to the whole network is equal to the network diameter[2]. The worst case network diameter for a connected network is $n - 1$, and this happens when the nodes are arranged in a chain.

Theorem 1. *In the absence of other traffic in the network, the maximum number of TDMA frames required to broadcast a packet to the whole network is equal to the network diameter.*

Proof. Let D be the network diameter. Let O be the originating node of the broadcast. The distance between the originating node and the furthest node is $\leq D$ hops.

At the end of the first TDMA frame, at least all the neighbours of O would receive the packet. At the end of the next TDMA frame, the packet from O would have reached all its 2-hop neighbours, and at the end of the third TDMA frame all the 3-hop neighbours of O would have received the packet. Continuing this at the end of D^{th} TDMA frame, all the D-hop neighbours of O would have received the packet.

Now, suppose there is a node that has not received the packet at the end of D TDMA cycles. This would imply that the number of hops required to reach the node from the originating node is greater than D. This is in contradiction to the definition of network diameter D. Hence, the maximum number of frames required equals the network diameter. □

[2]network diameter is the maximum length in terms of hop count of the shortest path between any two nodes in the network

4.2.2 Calculation of Lower and Upper Bounds on Number of TDMA Frames Required for Topology Broadcast

The topology broadcast phase ends when all nodes have sent their *nbrinfo* packet to all other nodes in the network. Each node broadcasts its own *nbrinfo* packet and re-broadcasts the *nbrinfo* packet of each of the $n-1$ other nodes. Thus in total, during the topology broadcast phase, a node transmits n packets (Note that each packet is actually duplicated and transmitted k times. For the analysis, these k transmissions are considered as a unit). We now calculate the lower and upper bound on the number of TDMA frames required. Since each node must transmit n packets during the topology broadcast phase, and a node can transmit only one packet in a TDMA frame, the lower bound is clearly:

$$f^b_{min} = n \tag{4.1}$$

Equation 4.1 is applicable when the network is fully connected.

To calculate the upper bound we consider the worst case scenario of nodes arranged in a line. The case of nodes arranged in a line is the worst case scenario because in any other connected network the average length of shortest path to the furthest node is less than that in a line. For nodes in a line, the average distance to the furthest node in the network is given by:

$$\overline{d} = \begin{cases} \dfrac{(3n+1)(n-1)}{4n} & \text{if } n \text{ is odd,} \\ \dfrac{3n-2}{4} & \text{if } n \text{ is even.} \end{cases} \tag{4.2}$$

If nodes perform the broadcast process sequentially, that is first node 0 completes the process, then node 1, etc., then the maximum number of frames required would be $\overline{d} \times n$ (from Theorem 1. If nodes perform the broadcast process in parallel, that is all nodes transmit their *nbrinfo* packet without waiting for other nodes to finish, then the number of frames required to complete the topology broadcast phase is less than or equal to the sequential case. This holds because there are no collision in the network, and we assume each node has an infinite queue size. Thus, the number of TDMA frames required to complete the topology broadcast phase is upper bounded by:

$$f^b_{max} = \overline{d}n \tag{4.3}$$

The upper bound in Equation 4.3 is very loose and in every simulated network, the number of TDMA frames required was far less. Referring again to Example 3.2.1 for illustration, Figure 4.5 shows the propagation of *nbrinfo* packets of selected nodes in the network (Figure 3.4. The network has 100 nodes, therefore the TDMA frame is of 600 time slots ($kn = 6x100$). We start the broadcast phase begins at $t = 4000$ (after neighbourhood discovery is over), and it continues for a little over 100 TDMA frames, which is equivalent to 60000 time slots. Note that broadcast phase doesn't end with the reception of the last new *nbrinfo* packet, it continues a little while after than as the nodes are clearing up their transmit queue. From the plots, we can make the following observations:

- The number of TDMA frames required by different nodes to broadcast their *nbrinfo* packet is different. This is reasonable because the broadcast propagation depends on the neighbourhood (the number and the IDs of neighbours), which is essentially random.

- A node which is centrally placed and well connected, such as node 1, requires less frames than a node at the edge of the network, such as node 90. This is expected because a node at the centre of the network has a lower average path length to other nodes and thus requires less number of frames to send its *nbrinfo* packet to the rest. Also, being in the centre, it propagates its packet in all directions, covering maximum number of nodes per broadcast.

- Notice that in some time slots a large number of new nodes receive the $nbrinfo$ packet of the originator, as compared to others. This depends on the neighbourhood of the node transmitting the $nbrinfo$ packet. If the neighbourhood is distinctly different from the previous nodes that transmitted the $nbrinfo$ packet, then a large number of new nodes receive the $nbrinfo$ packet.

Figure 4.6 shows the propagation of $nbrinfo$ packets for a network in which 100 nodes are arranged in a line with increasing node IDs. The separation between the nodes is equal to the transmission range. The broadcast phase ends in a little more than 140 TDMA frames, which is equivalent to 84000 time slots ($600x140$).

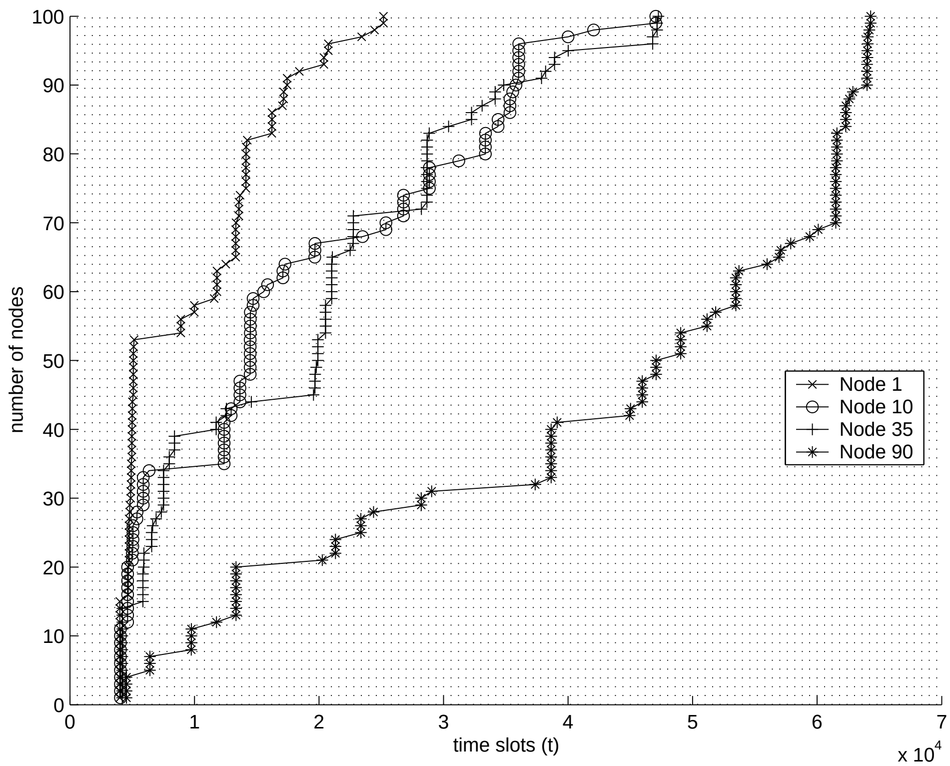


Figure 4.5: Broadcast propagation in the network shown in Figure 3.4. Each point represents a node receiving the *nbrinfo* packet of the originating node for the first time. The time required for all the nodes in the network to receive the *nbrinfo* packet of a particular node depends on the location of the originating node as well as the network topology. The y-axis is the number of nodes that have received the $nbrinfo$ packet. The dotted vertical lines represent the TDMA frame boundaries. Each TDMA frame is of 600 time slots

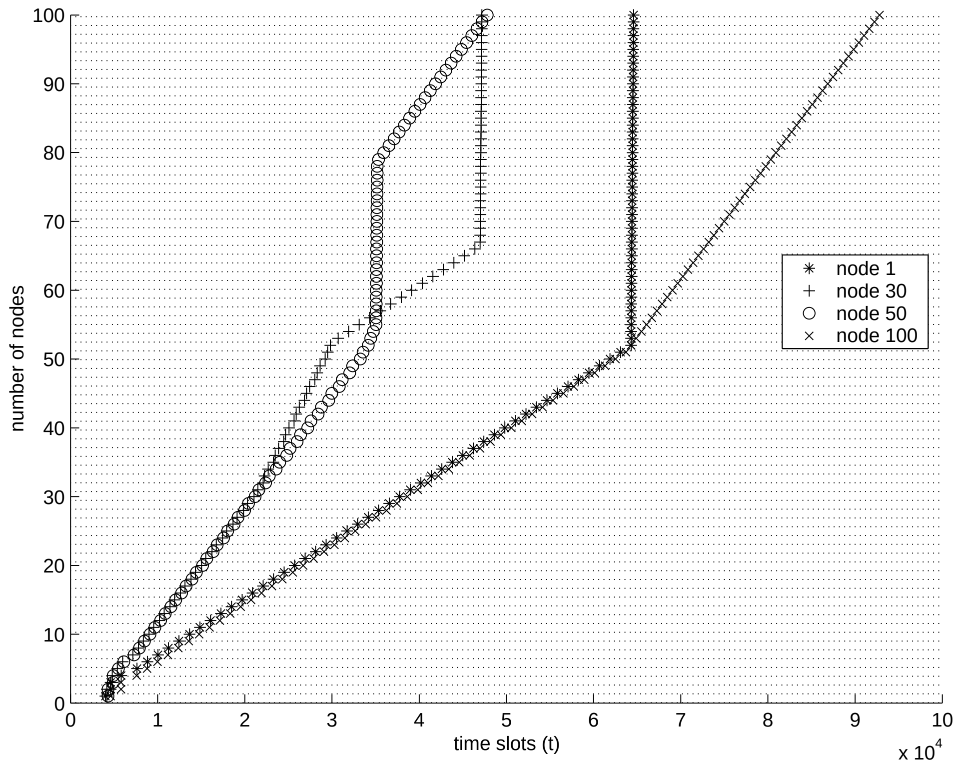


Figure 4.6: Broadcast propagation in the network with nodes arranged in a line. Each point represents a node receiving the *nbrinfo* packet of the originating node for the first time.

4.2.3 Termination of topology broadcast

Nodes have complete topology information when they have received the *nbrinfo* packets of all the nodes in the network. If there are are no node failures and no transmission errors, then it is trivial; nodes have complete topology information when they have received all the n *nrbinfo* packets. However, in the even that there are node failures or transmission erros, the other nodes in the network may wait perpetually to receive the *nbrinfo* packets from the failed nodes. To work around this we propose the concept of a consistent topology information state.

Definition 4.2.1. A node has consistent topology information when it has received the *nbrinfo* packet from every node that appears as a neighbour in any of the *nbrinfo* packets that it has received so far. Let $\mathcal{A}$ be the set of all the nodes whose *nbrinfo* packet the node has received plus itself. Let $\mathcal{N}_A = \bigcup_{\forall i \in A} \mathcal{N}_i^a$ (all the nodes from the *nbrinfo* packets received). Then the topology is consistent if and only if $\mathcal{A} = \mathcal{N}_A$, i.e. $\forall j \; [\; j \in \mathcal{A} \Longleftrightarrow j \in \mathcal{N}_A \;]$.

Theorem 2. *A node (A) can have consistent topology information only when it has* *received the nbrinfo packet from all the active nodes in the network that are part of the connected component to which the A belongs.*

Proof. Let $\mathcal{V}$ be the set of all the nodes that form the connected component (network), $|\mathcal{V}| > 1$, $A \in \mathcal{V}$. $\mathcal{A}$ is the set of all the nodes whose *nbrinfo* packet A has received. Then, $\mathcal{O} = \mathcal{V} \setminus \mathcal{A}$ is the set of nodes whose *nbrinfo* packets A has not received. Let $K \in \mathcal{O}$ be the node nearest to A (in terms of hopcount) such that $K \notin \mathcal{A}$. Let d be the distance from A to K.

Let us assume that the topology information is consistent. For the topology information to be consistent it requires that if $K \notin \mathcal{A}$ then $K \notin \mathcal{N}_A$. This means that K is not a neighbour of any of the $d-1$ hop neighbours of A. This is a contradiction because we assumed that all the nodes in $\mathcal{V}$ are connected. Thus a node that is d hops away from A must be a neighbour of some node that is $d-1$ hops away from A (c.f. Figure 4.7) □

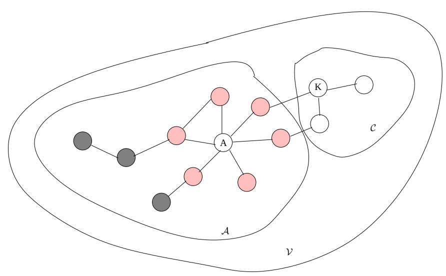


Figure 4.7: In a connected network, Node K, 2 hops away from A, must be the neighbour of one of the 1-hop neighbours of A.

Once the topology information available at a node is consistent, then the node has completed the topology discovery process.

4.2.3.1 Antenna switching during broadcast phase

The procedure *switch_antenna_b* is used to switch the antennas during the broadcast phase. Nodes can be in two states depending on the current time slot, *active* or *passive*. A node gets into the *active* state at $t_{TpBcastStart} + k \times id$ and remains active for the next k slots, after which it returns to the passive state. The cycle repeats every kn slots (a broadcast TDMA frame) till the end of the broadcast phase. If the node is in *active* state, then *switch_antenna_b* would choose the next antenna in a clockwise direction. If, however, the node is in *passive* state, then *switch_antenna_b* would lookup (from the

Algorithm 3 Antenna switching algorithm SWITCH_ANTENNA_B (broadcast)

Require: The node is in broadcast phase

procedure SWITCH_ANTENNA_B(*state*, *j*)
    if $state = active$ **then**
        $A \leftarrow (A + 1) \bmod k$ ▷ A is the antenna index, k is the number of antennas
    else if $state = passive$ **then**
        if $j \in NbrTable$ **then**
            $A \leftarrow NbrTable[j][AntennaIndex]$
        end if
    end if
end procedure

neighbour table) the antenna index that tunes the node with the currently active node, and switch to that antenna. If the currently active node is not a neighbour, then the selected antenna remains unchanged.

The *broadcast* algorithm begins by calculating the current TDMA frame C. TDMA frames are numbered from 0 onwards. In lines 3–5, the node checks if the broadcast phase has started, and if so, it creates an *nbrinfo* packet from the neighbour table and places it on the transmit queue, a FIFO queue. Lines 6–10 check whether it is time for the node to move into active state from passive state. If it is, then the node changes the state to *active* and resets the *aslotcount* counter. The *aslotcount* counter keeps track of the number of time slots that the node has been in *active* mode. If the node is in *active* state, and this is the first time slot of the current *active* state, then the node makes k duplicates of the packet at the transmit queue head. This is because the node will transmit the same packet (duplicates) in k directions. Lines 18–20 ensure that the node will return to the *passive* state after k contiguous time slots. If however, the node is in *passive* state then line 23 determines the currently active node in the network. This is required so that the node can tune itself to the currently active node, if that happens to be its neighbour. The node then listens for broadcast packets. If the node receives a broadcast packet that it has not seen before, then it will add it to the tail of the transmit queue (line 30). Each received packet is sent to an appropriate packet handler, based on its type. As an example, if the received packet is an *nbrinfo* packet then the packet is added to the list of received *nbrinfo* packets that the node maintains.

Algorithm 4 Broadcast algorithm BROADCAST

```
procedure BROADCAST
    $C = \lfloor \frac{t - t_{BcastStart}}{k \times n} \rfloor$                ▷ $C$ is the current TDMA frame number
    if $t = t_{BcastStart}$ then
        queue own nbrinfo packet at the transmit queue head.
    end if
    if $t = t_{BcastStart} + C \times k \times n + id \times k$ then
        $state \leftarrow active$
        $aslotcount \leftarrow 0$
        $inBcast \leftarrow True$
    end if
    if $state = active$ then
        call SWITCH_ANTENNA_B($state, id$)
        if $aslotcouont = 0$ then
            make $k$ duplicates of the packet at the head of transmit queue if any.    ▷ We will transmit the same packet $k$ times
        end if
        transmit packet at transmit queue head if any
        $aslotcount \leftarrow aslotcount + 1$
        if $aslotcount \geq k$ then
            $state \leftarrow passive$
        end if
    end if
    if $state = passive$ then
        $j = \lfloor \frac{t - t_{BcastStart} - C \times k \times n}{k} \rfloor$                ▷ $j$ is the currently active node.
        call SWITCH_ANTENNA_B($state, j$)
        listen for broadcast packets
        if received new broadcast packet then
            send a copy of packet to packet handler
            queue packet at the tail of transmit queue
        end if
    end if
end procedure
```

4.3 Implementation Details

The topology broadcast phase uses *nbrinfo* messages. The structure of the *nbrinfo* packet is shown in Figure 4.8. The message exchanges takes place over UDP, and all messages are sent to port PSTDMAPORT. A *nbrinfo* packet has the following fields:

type	length	id	timestamp	neighbour	neighbour	• • •	neighbour

Figure 4.8: Structure of *nbrinfo* packet (not to scale)

type : Type of packet, set to NBRINFO. Allows protocol to distinguish messages of different types. (size = 1 byte)

length : The length of the packet in bytes including the type and length fields. (size = 1 byte)

id : The ID of the originator of the message. (size = 4 bytes)

timestamp : The timestamp when the packet was created. The timestamp value is the time since epoch (00:00:00 UTC, January 1, 1970), measured in milliseconds. (size = 8 bytes)

neighbour : The ID of neighbour node. (size = 4 bytes)

In the event that the *nbrinfo* packet is larger than the MTU, the node can break the packet into multiple packets. Each *nbrinfo* packet acts independently on the topology matrix and generates the links (c.f. Figure 4.2). Thus, breaking a large *nbrinfo* packet into multiple smaller packets would still result in the same final topology matrix.

Each node maintains a data structure that contains the topology matrix. The topology matrix is a $n \times n$ binary matrix, with a 1 in position (i, j) if i appears in the *nbrinfo* packet of j. All the diagonal elements of the topology matrix are set to 1.

4.4 Conclusion

In this chapter, we described the method for broadcasting in the the network when a link schedule is not present. The chapter considered two approaches: random broadcast and a simple scheduled broadcast that uses node ID to ensure that nodes do not interfere with each other. Random broadcasts have a very low delivery rate, with many nodes failing to receive the broadcasted packets. We solved the problem by using the simple node ID based scheduling. The section also presented the algorithms for controlling the antenna during the broadcast phase. The antenna switching algorithm and the broadcast algorithm ensure that at any given time only one node is transmitting packets, and all its neighbours are tuned to it so that they can receive the broadcast packet. Topology broadcast, whose goal is to ensure that all the nodes in the network have the complete, consistent, topology information was also discussed. We showed that nodes would discover the topology in a finite time, and all the nodes in the network will have the same topology information.

Chapter 5

Link Scheduling

5.1 Link Scheduling

In this chapter we will discuss our proposed link scheduling algorithm for Achilles. The goal of link scheduling is to ensure that:

- Nodes are allocated time slots in which they can transmit
- Nodes know which antenna to switch to during each time slot so that they can activate the links. Two nodes (neighbours) have to tune to each other in order to activate the link between them.

To solve this problem Achilles uses a link based Spatial TDMA algorithm. Spatial TDMA has the advantage of providing a link schedule with the additional benefit of scheduling multiple links during the same time slot if the transmissions do not interfere with each other.

5.1.1 Spatial TDMA

In this section we describe Spatial TDMA. We use arguments and notations as used in [17, 9, 11].

Consider the network to be represented by a directional graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices (nodes) and $\mathcal{E}$ is the set of edges (links). If only one link is activated during a time slot, then it is obvious that there is no collision in the network. However, as the nodes are distributed in space, it is possible that more than one link can be activated at the same time without any collisions if the nodes are sufficiently separated in space, or use directional antennas. Spatial TDMA takes advantage of this possibility to schedule multiple links in the same time slot and thus gain capacity while reduce the length of the TDMA cycle, resulting in lower packet delays.

The main task is to determine the set of links $\mathbf{C}$, $\mathbf{C} \subseteq \mathcal{E}$, that can be activated or switched on during a time slot. Such a set of links are called *compatible* links. In general, a network would have many such sets of compatible links. Each of the sets considered is a *maximal* set in the sense that in any given set of compatible links, no more links can be accommodated [17].

When a link is *activated*, transmission can take place over the link. We represent a link by (i, j), with i being the tail (transmitter) and j the head (receiver). For a link to be active, nodes i and j must tune to each other. Some of the constraints on the set $\mathbf{C}$ are as follows:

All the transceivers are half-duplex, and nodes cannot transmit and receive at the same time:

$$(i, j) \in \mathbf{C} \wedge (k, i) \in \mathbf{C} \quad \text{is False}, \quad \forall i, j, k. \tag{5.1}$$

The second constraint is due to unicast nature of transmissions. A transmission has only one intended recipient:

$$(i, j) \in \mathbf{C} \wedge (i, k) \in \mathbf{C} \quad \text{is False}, \quad \forall i, j, k, j \neq k \tag{5.2}$$

The third constraint is based on the limitation that a node can receive packets from only one transmitter at a time:

$$(i, j) \in \mathbf{C} \wedge (k, j) \in \mathbf{C} \quad \text{is False}, \quad \forall i, j, k, i \neq k. \tag{5.3}$$

The final constraint takes into account the interference resulting from ongoing simultaneous transmissions. For packet reception to be successful, the interference from the other activated links should be limited such that the SINR (Signal to Interference Noise Ratio) for each link is greater than the threshold for reception β.

$$SINR_{ij} > \beta, \quad \forall (i, j) \in \mathbf{C} \tag{5.4}$$

Where $SINR_{ij}$ is the SINR at node j when receiving signal transmitted by node i.

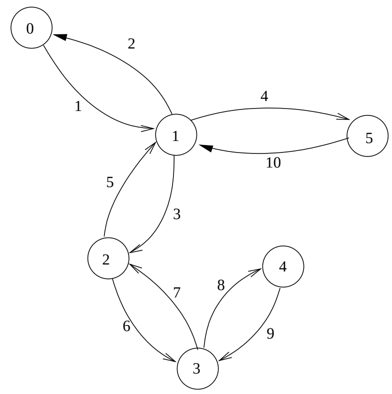


Figure 5.1: A sample network with numbered links

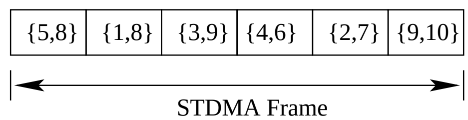


Figure 5.2: STDMA frame specifying the active links in each of the slots.

Example 5.1.1. Let us look at example network as shown in Figure 5.1. Nodes in this network use *omnidirectional* antennas. We use the graph model of collision to illustrate the concept of compatible links.

$$\mathcal{E} = \{(0,1),(1,0),(1,2),(1,5),(2,1),(2,3),(3,2),(3,4),(4,3),(5,1)\}$$

$$\mathcal{E} = \{1,2,3,4,5,6,7,8,9,10\} \quad \text{with numbered edges}$$

Edges are ordered in ascending order, with edge $(a,b) > (c,d)$ if $a > c$, or if $a = c$ then $(a,b) > (c,d)$ if $b > d$. Edges are numbered according to their position in the edge set. Thus, edge $(0,1)$ is numbered 1, $(1,0)$ is 2, and so on. The following are the compatible edge sets:

$$C_1 = \{5,8\} \quad C_2 = \{1,8\} \quad C_3 = \{3,9\} \quad C_4 = \{4,6\} \quad C_5 = \{2,7\} \quad C_6 = \{9,10\}$$

$$C_7 = \{8,2\} \quad C_8 = \{8,10\} \quad C_9 = \{6,1\} \quad C_{10} = \{7,1\} \quad C_{11} = \{9,1\} \quad C_{12} = \{9,4\}$$

As we can see, links 5 and 8 can be simultaneously turned on, similarly links 3 and 9, and so on. Our goal is to ensure that each link has the opportunity to be active at least once during the TDMA cycle. Thus, we make a collection of compatible link sets which include all the links. Such a collection is called a *cover*. An example *cover* is $\{C_1, C_2, C_3, C_4, C_5, C_6\}$. A simple transmission schedule would be to assign links in C_1 slot 1, C_2 slot 2, and so on, until we have covered every link that we want to schedule. Thus, the TDMA frame would have six slots and each link would have an opportunity to be active at least once during the TDMA frame (c.f. Figure 5.2).

5.1.2 Basic Steps for STDMA

Definition 5.1.1. A *cover* is a set of compatible link sets such that every link in the network is contained in at least one of the included sets.

The basic steps in creating a STDMA schedule are:

1. Generate compatible link sets. Except for the situation in which all the nodes are isolated, there are, in general, multiple compatible link sets.

2. Select a number of compatible link sets and make a combination which provides a *cover*. Assign each link set in the *cover* a number of time slots in the frame. It is not necessary that each link set have the same number of time slots assigned to them. Based on traffic demand, node priority, etc. some link sets may be assigned more time slots than others.

We look at the two steps one by one. The first step is to generate the compatible link sets. Unlike Example 5.1.1 where nodes used *omnidirectional* antennas and the graph model of collision was assumed, determining compatible links is much more difficult when directional antennas are used and interference model of collision is applied. Spatial TDMA using the interference model is covered by Grönkvist in [9], in which the author proposed two algorithms for STDMA scheduling. We use Algorithm I from [9] as the basis with one important difference. Grönkvist used interference calculations (using Detvag-90 software) to determine if a set of links is compatible, while we replace the calculation by actual measurements performed in the live network. The algorithms proposed by Grönkvist do not create the compatible link sets as a separate step, instead it is done along with the time slot assignment.

The second step is one of assignment of time slots to compatible link sets. The are two main problems here: i) given a number of compatible link sets, how to make a collection of link sets that is optimal and ii) what is the order and number of time slots that should be assigned to each of the compatible link sets in the collection such that it is optimal. Obviously, the meaning of optimal here depends on what we are trying to optimise. A popular definition of optimal assignment is the assignment that generates the shortest possible TDMA cycle. A short TDMA cycle ensures that packet delays are kept low. To generate the shortest possible TDMA cycle, we have to select the minimum number of compatible sets that provide a *cover*. The optimal selection of compatible link sets and the optimal assignment of time slots is a NP-hard problem [5], we therefore use the algorithm proposed by Grönkvist for time slot assignment.

5.1.3 Algorithm for link scheduling

The STDMA link scheduling algorithm, Algorithm 5, creates a STDMA frame with a number of time slots. Each link in the network is assured at least one time slot in the frame. The algorithm is greedy in the sense that it tries to accommodate as many links in each time slot as possible. The algorithm begins by creating two sets of links *assigned* and *notassigned*. At the start, all the links are in the *notassigned* set. As soon as a link is assigned a time slot, it is moved to the *assigned* set. If a link cannot be assigned during a particular time slot, the link is said to be delayed. The algorithm maintains a counter *skipped*, that keeps track of the number of consecutive time slots that a link has been delayed. At the end of each loop, the links in *assigned* and *notassigned* sets are sorted in descending order of the *skipped* value. Thus, a link that has been delayed more than others has the higher priority in the next round of allocation. This ensures that links are assigned time slots as soon as possible. A critical part of the algorithm is the test to determine whether or not a set of links is compatible. The original algorithm [9] used offline interference calculations to determine the compatibility. We replace the offline calculation by a procedure to do live measurements, thus enabling the operation of the algorithm even when nodes are deployed randomly. The algorithm terminates when all the links are assigned a time slot. In the worst case, the number of time slots in the STDMA frame equals the number of links in the network. The array *links* contains the links assigned to each slot in the STDMA frame.

Algorithm 5 STDMA link assignment algorithm

```
procedure LINK-SCHEDULE
    skipped[i] ← 0      ∀i ∈ S   ▷ For all the links in the sorted edge list the skipped counter is initialised to 0
    notassigned ← sortededges, assigned ← ∅   ▷ At the beginning at the links are in the notassigned list
    slotnumber ← 0
    repeat
        slotnumber ← slotnumber + 1
        links[slotnumber] ← ∅   ▷ links[slotnumber] has the list of links assigned a particular slot number
        for each link i ∈ notassigned do
            Append i to links[slotnumber]
            if Links in links[slotnumber] are compatible then
                skipped[i] ← 0
                append i to assigned
            else
                remove i from links[slotnumber]
                skipped[i] ← skipped[i] + 1
            end if
        end for
        for each link i in assigned but not in links[slotnumber] do
            append i to links[slotnumber]
            if Links in links[slotnumber] are compatible then
                skipped[i] ← 0
            else
                remove i from links[slotnumber]
                skipped[i] ← skipped[i] + 1
            end if
        end for
        sort lists assigned and notassigned in descending according to skipped value for the link.
    until notassigned is empty
end procedure
```

5.1.4 Live measurements to determine link compatibility

Suppose you want to determine if a set of links $\mathbf{c} \subseteq \mathcal{E}$ are compatible. If all the node positions, node orientations, and complete topology information is known a priori then offline calculations using a suitable path loss tools may be possible. However, in a real life deployment, interference calculations are complicated by the several factors:

- Variations in node orientation at the time of deployment. Even small variations in the node orientation can lead to very different interference statistics, especially when using directional antennas.
- In general, the node hardware has some inhomogeneity, such as different transmission power, cable loss, antenna gains, etc. These factors acting together, can often lead to significant differences in the transmitted power from the nodes.
- The complex multipath characteristics of the region in which nodes are deployed are difficult to capture. Multipath characteristics have significant impact on the interference faced by nodes.
- Obstacles in the deployment region such as islands and buildings, complicate offline calculations.
- Finally, when nodes are deployed in a random manner, offline calculations are not possible.

In view of the above limitations, we resort to live measurements in the network to determine compatibility. So, for example, a way to find if a set of links $\mathbf{C}$ are compatible, is to enable all the links in $\mathbf{C}$ and measure the packet loss rate in each of the links in set $\mathbf{c}$. If the packet loss rate is below a certain system specified threshold (ρ), we reach the conclusion that the links in $\mathbf{C}$ are compatible. To implement this method, we designed a mechanism that allows the network to test the compatibility of the links online. The topology matrix provides the candidate compatible links. The measurements choose a subset of the topology matrix based on loss rate.

Consider line 10 and 20 of Algorithm 5. To check if a given set of links are compatible, we break the statement into the following stages (c.f. Figure 5.3). First, the set is tested to see if the links satisfy the constraints in (5.1),(5.2) and (5.3). If they pass the first

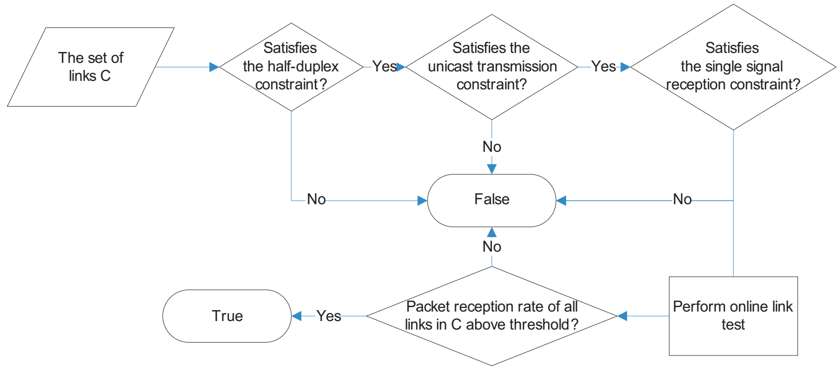


Figure 5.3: Flowchart showing how a set of links is tested for compatibility

stage, then an online test is performed on the links. To perform the online test, the orchestrating node (for simplicity, node with the smallest ID) broadcasts the link set (to be tested) and the start time of the test $LinkTestStart$ to all the other nodes in the network. At $LinkTestStart$ the links are activated for a predetermined number of time slots ($TestSlotCount$). The transmitters of the active links send packets of size equal to MTU of the link. The receivers keep a count of the number of packets received during the test period. At the end of the test period, the receivers broadcast the results to the network. After collecting the results from all the receivers, the orchestrating node determines the compatibility of the links. If any of the links has a loss rate higher than a system specified threshold ρ, then the links are considered to be incompatible.

5.1.4.1 Measuring the Broadcast Delay

The orchestrating node needs to specify the $LinkTestStart$ time when a set of links would be tested. How do we ensure that all the nodes, at least the nodes which are required for the particular test, have received the $linktest$ message before $LinkTestStart$ time? The time required for a broadcast packet, originated by the orchestrating node, to reach all the other nodes is easily calculated by sending a probe packet ($delayprobe$), and requiring all the remaining nodes in the network to send a reply packet containing the slot number in which the probe packet was first received by the node. From the replies obtained the orchestrating node can calculate the number of time slots required for a message to reach all the nodes in the network. However, the delay value calculated is specific to the state of the network at the time when the probe packet was sent – state here being the size of the transmit queue of each node in the network – it is clear that a different state of the network could result in a different number of required time slots.

To tackle this problem we introduce a new class of packets called *prio.e*. The packets of *prio.e* class have the following properties:

- A *prio.e* packet is always placed at the head of the transmit queue.
- There is only one *prio.e* packet in the queue. If a new *prio.e* packet needs the be queued, then all other packets of *prio.e* class, if any, are flushed from the transmit queue. This property ensures that the packet is never delayed, though it may fail to reach all the nodes in the network.

A consequence of the above mentioned properties is that a packet of *prio.e* class is propagated through the network as if all the transmit queues were empty. The *delayprobe* packet is of class *prio.e*. Packets of *prio.e* class are sent only by the orchestrating node. The other packet in *prio.e* class is the *linktest* packet.

A *delayprobe* packet is like a ping packet. It is sent by the orchestrating node and it travels through the network using the broadcast mechanism. A node upon receiving the *delayprobe* packet broadcasts a response packet (*delayresp*) containing the identifier for the *deplayprobe* packet it is replying to, its ID, and the time slot in which the *delayprobe* packet was received. The *delayresp* packets are of normal priority (*prio.n*). The orchestrating node waits until it has received the response packets from all the nodes in the network. By finding the node which received the *delayprobe* packet the last, the broadcast delay for packets of class *prio.e* is calculated.

Example 5.1.2. Here we illustrate the operation of the *delayprobe* packet with the help of an example network. We use the network as shown in Figure 3.9. Node 0 is the orchestrating node. Each node has six directional antennas. Let t_l be the time when node 0 sends the *delayprobe* packet. At this point of time, the nodes in the network are in the broadcast phase. The state and activity of the nodes are shown below:

$t_l + 0$ Node 0 sends the *delayprobe* packet.

$t_l + 6$ Node 1 has received the *delayprobe* packet. Node 1 queues the *delayresp*.1 packet in the transmit queue and forwards the *delayprobe* packet.

$t_l + 12$ Node 2 and Node 3 have received the *delayprobe* packet. Nodes 2 and 3 queue the *delayresp*.1 packet in the transmit queue. Node 2 forwards the *delayprobe* packet.

$t_l + 18$ Node 3 forwards the *delayprobe* packet.

$t_l + 24$ No action.

$t_l + 30$ Node 1 transmits the *delayresp*.1 packet queued in its transmit queue.

$t_l + 36$ Node 0, 2, and 3 have recived the *delayresp*.1 packet and queued it in their transmit queue. Node 2 transmits its *delayresp*.2 packet.

$t_l + 42$ Node 1 and 3 have received the *delayresp*.2 packet from node 2 and queued it in their transmit queue. Node 3 transmits its *delayresp*.3 packet.

$t_l + 48$ Node 1 and Node 2 have received the *delayresp*.3 packet and queued it in the transmit queue. Node 0 forwards the *delayresp*.1 packet.

$t_l + 54$ Node 1 forwards the *delayresp*.2 packet.

$t_l + 60$ Node 0 has received the *delayresp*.2 packet. Node 2 forwards the *delayresp*.3 packet.

$t_l + 66$ Node 3 forwards the *delayresp*.2 packet.

$t_l + 72$ Node 0 forwards the *delayresp*.2 packet.

$t_l + 78$ Node 1 forwards the *delayresp*.3 packet.

$t_l + 84$ Node 0 has received the *delayresp*.3 packet. Node 0 has received the response packet from all the nodes in the network.

5.2 Processing a *linktest* packet

On receiving a *linktest* packet, nodes schedule a test event for the period starting from the $LinkTestStart$ time slot to the $LinkTestStart + TestSlotCount$ time slot (referred to as test period). The behaviour of the node during this test period depends on whether or not it is part of any of the links specified for the test. If it is a source or destination in any of the specified links, then the node will take the appropriate steps to i) select the required antenna to set up the link and ii) transmit test packets, or be in reception mode and count the number of packets received, respectively. Nodes that are not part of any of the links specified in the test will not transmit any packets during the test period.

Algorithm 6 Processing a *linktest* packet

```
if t < LinkTestStart then
    testtime ← LinkTestStart
    testslots ← TestSlotCount
    if (id, j) ∈ C then    ▷ Node is a transmitter in one of the links specified in the test set C.
        testtx ← True, testrx ← False
        testantenna ← NbrTable[j][AntennaIndex]
    else if (i, id) ∈ C then    ▷ Node is a receiver in one of the links specified in set C.
        testtx ← Flase, testrx ← True
        testantenna ← NbrTable[i][AntennaIndex]
    else
        testtx ← False, testrx ← Flase
        testantenna ← 0
    end if
end if
```

5.3 Performing Link Test

The following events take place in order to test a set of links C for compatibility:

- The orchestrating node (ON) broadcasts a *linktest* message to the network specifying the desired link set C.
- Nodes receive the packet and schedule the test.
- During the test period, nodes tune their antenna such that the links in set C are set up.
- Nodes that are transmitters in set C transmit a known test packet of size equal to MTU of the link.
- Nodes that are receivers in set C count the number of test packets received.
- Other nodes remain silent.
- At the end of the test period, the receiver nodes in set C broadcast a *linkresult* packet informing the orchestrating node of the number of test packets received.
- ON waits till it receives the *linkresult* packets from all the receiver nodes in C. ON then determines whether or not the links are compatible based on the performance of the links. If the packet reception rate for all the links is greater than a threshold ρ then the links are declared compatible.

During the test period the antenna selection scheme changes from the default scheme for the broadcast phase. This is required to ensure that the links being tested are set up for the whole test period. At the end of the test period the node behaviour returns to the same state as it would have been if the test period did not occur. This is achieved by nodes continuing to run the antenna switching procedure, but the physical action of switching the antenna is masked and overridden to ensure that the appropriate links to be tested are set up. The transmit queue of the node is saved just before the node enters the test period, and is restored at the end of the test period. At the beginning of the test period $TestSlotCount$ copies of the test packet are copied to the transmit queue of transmitter nodes. Algorithm 7 shows the behaviour of the node during the test period.

Algorithm 7 Node behaviour during test period

```
if t = testtime then
    A ← testantenna
    if testtx = True then
        queue testslots copies of test packet in the txqueue
    else if testrx = True then
        testrcvd ← 0
    else
        do nothing
    end if
end if
if testtime ≤ t < testtime + testslots then
    if testtx = True then
        transmit packet at transmit queue head
    else if testrx = True then
        Listen for test packet
        if packet received from peer then
            testrcvd ← testrcvd + 1
        end if
    else                                        ▷ Node not involved in the test
        do nothing
    end if
end if
```

5.4 Broadcasting the STDMA Schedule

When the STDMA link assignment algorithm terminates, Orchestrating Node (ON) has to broadcast the information about the STDMA Frame to all the nodes in the network. The STDMA Frame is completely specified by i) the start time of the first STDMA

Frame, ii) the number of time slots in each frame, and iii) the links active during each time slot. ON broadcasts this information using $frameinfo$ packets. $Frameinfo$ packets are of class *prio.e* and are sent at a period of $EDelay$. All the $frameinfo$ packets other than the final packet contains 0 in the *stdmaactivate* field. For each time slot in the STDMA Frame, a $frameinfo$ packet is sent containing the links active during that specific time slots. If however, the list of links is large and cannot be accommodated in one packet, then the list is split in several packets and the receiver rebuilds the final list by concatenating the packets.

The final $frameinfo$ packet contains a non-zero value for the *stdmaactivate* field. This specifies the time slot from which the STDMA schedule will become active. The time specified in *stdmaactive* is at least $EDelay$ slots in future.

5.5 The Operational Phase

At the start of *stdmaactivate* time slot, nodes begin to use the STDMA schedule to determine antenna selection and transmission scheduling. The network now enters the *operational* phase. The antenna selected during each time slot is determined by the link that is scheduled during that particular time slot. Nodes keep the STDMA schedule in the $STDMASched$ table. An example table is shown in Table 5.1 – for node 1 in Figure 5.1. The $STDMASched$ table specifies the behaviour the the node during each time slot. For those time slots in which the node is neither a scheduled transmitter, nor a scheduled receiver, the node selects a random antenna, and is in receiving mode. This allows nodes (already in operation mode) to receive *hello* messages from new nodes. A process can then be started to introduce the new node in the network. In this thesis we do not describe the process of introducing of a new node, but leave this facility for further work.

Slot	Antenna	Peer	Tx/Rx
0	4	2	Rx
1	5	0	Rx
2	3	2	Tx
3	1	5	Tx
4	5	0	Tx
5	1	5	Rx

Table 5.1: An example of an $STDMASched$ table in which the STDMA schedule is maintained.

For each outgoing link, the node maintains a transmit queue. When the schedule activates the outgoing link, packets from the transmit queue of that particular link are transmitted. Figure 5.4 illustrates the formation of the multiple transmit queues in the MAC layer.

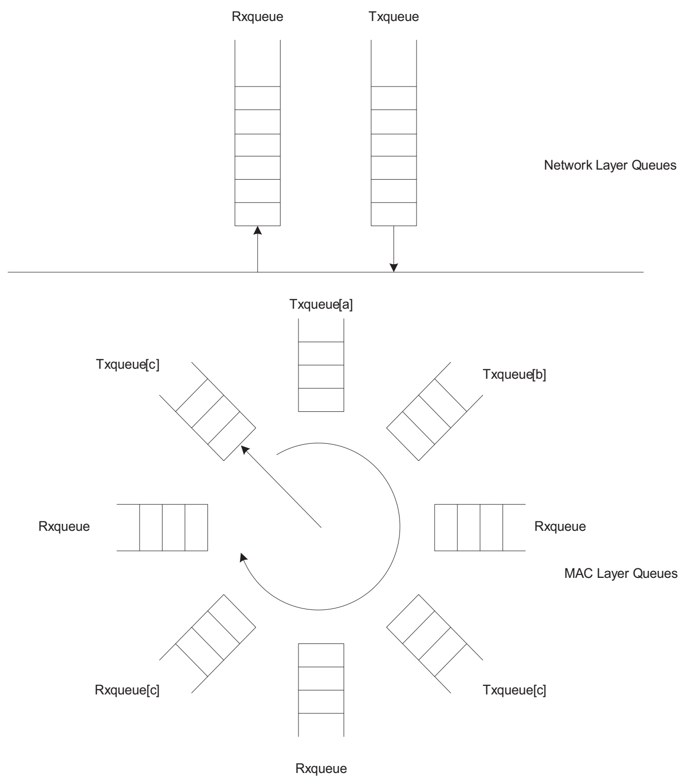


Figure 5.4: MAC layer queues for STDMA. Each outgoing link has its own transmit queue. Queues are selected based on the STDMA schedule.

Routing protocol can now be started to discover routes and to begin network operation for data transport.

5.6 Implementation Details

The schedule design phase uses five types of messages: *delayprobe*, *delayresp*, *linktest*, *linkresult*, and *frameinfo*. All the messages are encapsulated in UDP packets and are sent to the port DSTDMAPORT.

5.6.1 *Delayprobe* message

The *delayprobe* message is sent to measure the broadcast delay of packets of class *prio.e*. The structure of the packet is shown in Figure 5.5.

type	length	id	timestamp

Figure 5.5: Structure of a *delayprobe* packet (not to scale)

type : Type of packet, set to DELAYPROBE. Allows protocol to distinguish messages of different types. (size = 1 byte)

length : The length of the packet in bytes including the type and length fields. (size = 1 byte)

id : The ID of the originator of the message. (size = 4 bytes)

timestamp : The timestamp when the packet was created. (size = 8 bytes)

The *id* and *timestamp* fields are used to uniquely identify a *delayprobe* packet. On receiving a new *delayprobe* packet, the node will place it at the head of the transmit queue. The node will flush out packets of class *prio.e*, if any, from the transmit queue.

5.6.2 *Delayresp* message

On receiving a new *delayprobe* packet, a node creates a response packet, *delayresp*. This packet informs the originator node of the *delayprobe* packet, the time when the *delayprobe* was received by the node. The structure of the packet is show in Figure 5.6

type	length	oid	otimestamp	rid	rtimestamp

Figure 5.6: Structure of a *delayresp* packet (not to scale)

type : Type of packet, set to DELAYRESP. Allows protocol to distinguish messages of different types. (size = 1 byte)

length : The length of the packet in bytes including the type and length fields. (size = 1 byte)

oid : The ID of the originator of the *delayprobe* message to which this message is a reply. (size = 4 bytes)

otimestamp : The timestamp from the *delayprobe* message to which this message is a reply. (size = 8 bytes)

rid : The ID of the node which created this message. (size = 4 bytes)

rtimestamp : The timestamp when the *delayprobe* message was received by the node. (size = 8 bytes)

Delayresp messages are of normal priority, *prio.n*.

5.6.3 *Linktest* message

Linktest message is used to perform a compatibility test for links. It contains the time of test, the number of consecutive slots that should be tested, and the list of the links that should be tested. These packets are of priority *prio.e*.

type	length	id	timestamp	linkteststart	testslotcount	nlinks	links

Figure 5.7: Structure of a *linktest* packet (not to scale)

type : Type of packet, set to LINKTEST. Allows protocol to distinguish messages of different types. (size = 1 byte)

length : The length of the packet in bytes including the type and length fields. (size = 1 byte)

id : The ID of the originator of the packet. (size = 4 bytes)

timestamp : The timestamp when the packet was created. (size = 8 bytes)

linkteststart : The time when the link test period starts. (size = 8 bytes)

testslotcount : The number of time slots for which the link test is carried out. (size = 2 bytes)

nlinks : The number of links specified in the links field. (size = 2 bytes)

links : A bitmap specifying the links that should be tested for compatibility. (variable size)

Each node has the topology map and the set of links in the network $\mathcal{E}$. Nodes create a sorted list of links (*sortededges*). Links are sorted in ascending order with link $(a, b) > (c, d)$ if $a > c$, or if $a = b$, then $(a, b) > (c, d)$ if $b > d$. A 1 at position j in the bitmap *links* implies that the link referred to by *sortededges*[*j*] will be active in the test.

5.6.4 *Linkresult* message

On completion of a compatibility test, the nodes that were receivers in the test broadcast the result of the test using the *linkresult* message. These packets are of normal priority *prio.n*.

type	length	oid	otimestamp	rid	rcount

Figure 5.8: Structure of a *linkresult* packet (not to scale)

type : Type of packet, set to LINKRESULT. Allows protocol to distinguish messages of different types. (size = 1 byte)

length : The length of the packet in bytes including the type and length fields. (size = 1 byte)

oid : The ID of the originator of the *linktest* message to which this message is a reply. (size = 4 bytes)

otimestamp : The timestamp from the *linktest* message to which this message is a reply. (size = 8 bytes)

rid : The ID of the node which created this message. (size = 4 bytes)

rcount : The number of test packets received successfully. (size = 2 bytes)

5.6.5 *Frameinfo* message

These messages are used by the ON to inform the result of the nodes in the network about the STDMA schedule. The message contains information about the links that are active in each STDMA slot.

type	length	timestamp	stdmaactivate	nslots	slot	nlinks	links

Figure 5.9: Structure of a *frameinfo* packet (not to scale)

type : Type of packet, set to FRAMEINFO. Allows protocol to distinguish messages of different types. (size = 1 byte)

length : The length of the packet in bytes including the type and length fields. (size = 1 byte)

timestamp : The timestamp when this message was created. (size = 8 bytes)

stmaactivate : The time when the STDMA schedule will become active. The specified time is always at a time slot boundary. (size = 8 bytes)

nslots : The number of time slots in the STDMA frame. (size = 2 bytes).

slot : The time slot in the STDMA frame for which this packet specified the active links. (size = 2 bytes)

nlinks : Number of links specified in the links field. (size = 2 bytes)

links : A bitmap specifying the links that should be active in the particular time slot. (variable size)

5.7 An Example STDMA Schedule

Table 5.2 shows an example STDMA schedule generated for a network of 20 nodes with an average node degree of 6. Note the multiple links (tx,rx) scheduled in each time slot.

5.8 Conclusion

In this chapter, we presented our method for generating a link schedule based on STDMA. We solved of the problem of determining whether a set of links can transmit at the same time without interfering with each other – called compatible links. Our method uses online tests to determine the compatibility of links in the network, allowing the nodes to form an STDMA schedule. Once we have the set of compatible links we can use the STDMA link scheduling algorithm proposed by Gronkvist et. al. in [9] to form an optimal link schedule. Note that because our method uses online tests, it can form STDMA schedules in randomly deployed networks – something that is not possible with offline schedules – and allows the network to be auto-configurable. We presented

Slot ID	Scheduled Links
0	(1,2) (3,10) (4,7) (5,12) (6,8) (9,16) (11,17) (13,15)
1	(1,4) (2,6) (3,11) (5,14) (7,13) (8,9) (10,12) (18,15) (20,16)
2	(1,6) (2,4) (3,13) (5,19) (7,18) (10,11) (15,9)
3	(1,7) (2,8) (3,17) (4,9) (10,13) (12,5) (14,18) (15,16)
4	(1,8) (2,9) (3,18) (4,15) (11,10) (13,7) (14,5) (16,20)
5	(1,9) (2,7) (5,12) (8,6) (10,3) (13,14) (15,4) (17,11) (20,16)
6	(1,15) (4,2) (6,7) (9,20) (11,3) (12,10) (13,18) (14,19)
7	(1,16) (4,20) (5,19) (6,2) (7,8) (10,12) (11,17) (13,3) (18,14)
8	(1,18) (4,16) (6,9) (7,2) (13,10) (17,3) (19,14)
9	(1,20) (3,10) (5,14) (6,15) (7,4) (9,2) (13,19) (17,11)
10	(2,1) (7,6) (9,8) (10,11) (12,5) (14,13) (15,18) (16,4) (17,3)
11	(2,15) (3,11) (4,1) (5,12) (7,19) (9,6) (10,13) (14,18)
12	(2,16) (3,17) (7,1) (8,9) (11,10) (14,5) (18,13) (20,4)
13	(2,8) (5,19) (6,1) (7,16) (9,4) (11,17) (12,10) (13,14) (18,3)
14	(4,20) (5,14) (7,15) (8,2) (9,1) (10,12) (11,3) (18,19)
15	(3,13) (8,1) (9,15) (12,10) (16,2) (17,11) (18,7) (19,14)
16	(2,6) (3,18) (4,16) (8,7) (10,11) (14,13) (15,1) (19,5) (20,9)
17	(2,6) (8,15) (9,20) (10,3) (11,17) (12,5) (16,1) (18,14) (19,7)
18	(1,20) (3,10) (4,9) (5,12) (6,8) (15,2) (16,7) (17,11) (19,13)
19	(1,20) (4,7) (12,10) (14,5) (15,6) (16,9) (17,3) (19,18)
20	(2,9) (5,14) (8,6) (11,10) (13,3) (15,7) (16,20) (18,1)
21	(3,17) (6,2) (7,4) (9,16) (10,12) (13,18) (14,19) (15,8) (20,1)
22	(1,2) (6,7) (8,9) (10,11) (12,5) (14,18) (15,13) (16,4) (17,3)
23	(1,6) (2,9) (3,11) (5,19) (7,13) (12,10) (15,20)
24	(1,8) (4,20) (6,9) (7,18) (11,3) (12,5) (13,10) (16,15) (19,14)
25	(1,4) (8,6) (9,2) (10,12) (11,17) (13,7) (18,3) (19,5) (20,15)

Table 5.2: STDMA schedule for 20 nodes

the protocol messages required for link tests, creation of the schedule, and disseminating of the schedule to all the nodes in the network. At the end of the schedule generation phase, the nodes in the network have a schedule which determines the antenna selection and radio behavior in each time slot. The network now enters the operational phase and is ready for data transmission.

Chapter 6

Evaluation

6.1 Overview

In this chapter, we present the performance evaluation of Achilles, in particular the performance of STDMA link scheduling and directional antenna system used. We compare the performance of Achilles's STDMA MAC with IEEE IEEE 802.11 mesh network and simple TDMA mesh network. The performance is measured using the following three metrics:

- **Total Network Throughput:** The total number of bytes successfully transported at the *application layer* per unit time. Since the simulations involve neighbour discovery and route discovery, we allow a settling down time of 100 seconds upon initialisation. We do not include the traffic generated in the initialisation phase in the throughput calculation. Throughput is computed as the total number of packets delivered successfully at the destination divided by the time taken.

- **Average Packet Delay:** The average delay (in seconds) incurred from the time a packet is transmitted at the source node to the time when it is received at the destination node. Only packets that are successfully received are used in the packet delay calculation.

- **Packet Delivery Ratio:** The ratio of the packets received at the destination node to the packets transmitted at the source node.

In order to test the performance over varied network sizes and density, we use network of three sizes: 20, 40, and 100 nodes. The networks used have node degrees of 6 (low density) and 12 (high density). All networks simulated are connected.

6.2 Simulation Setup

All simulations are carried out using the Qualnet Simulator (v3.8) [18]. We implemented the neighbour discovery, topology broadcast and link scheduling required by Achilles. In addition, the MAC layer was implemented to use the STDMA link schedules as required by Achilles. For IEEE IEEE 802.11 mesh and TDMA mesh, we used the existing facilities provided by Qualnet.

- Networks of sizes 20, 40, and 100 are generated with densities resulting in average node degree of six and 12. Nodes are placed randomly in a square plane. The terrain size varies from 1000 x 1000 square metres to 50000 x 50000 square metres, depending on the number of nodes and required node degree.

- The wireless transceiver characteristics such as transmission power, receive sensitivity etc. are chosen to be identical to the ones of the IEEE IEEE 802.11a/b/g radio CM9 [7]. Link rate of 6 Mbps is used throughout the simulations. This is the IEEE IEEE 802.11a basic rate.

- IEEE 802.11, TDMA, and STDMA MAC protocols are used. The IEEE IEEE 802.11 MAC uses the standard IEEE IEEE 802.11 DCF function. In TDMA each node in the network is allocated one slot per TDMA frame in which it can transmit, in all other slots the nodes is in receive mode. In STDMA, which Achilles uses, multiple nodes in the network can transmit in each TDMA slot; the Tx nodes and Rx nodes in each slot are determined by the generated STDMA schedule.

- IEEE IEEE 802.11a and TDMA nodes use omni-directional antenna, whereas Achilles nodes use a set of 6 directional antennas, each with a beam-width of 60^o.

- The gain of the antennas don't impact the simulations as we create network topologies with a fixed average node degree as the design parameter. There is no argument that directional antennas have higher gain and therefore greater range than omni-directional antennas. IEEE IEEE 802.11 and TDMA mesh networks are unable to use multiple directional antennas. Therefore, for the purpose of evaluation, we choose average node degree as the topology parameter and not terrain dimensions.

- Signal propagation in the channel follows the Two-Ray model [20].
- Only static networks are considered in the simulation. Achilles is designed as a static mesh network.
- Each node has a maximum IP queue size of 100 packets. Once the queue is full, incoming packets are dropped. The IP queue is FIFO.
- For IEEE IEEE 802.11 and TDMA mesh, we use the Distributed Bellman-Ford unicast routing protocol. For Achilles, we calculate the routes using the Dijkstra shortest path algorithm; links which are scheduled more often have a lower weight than those scheduled less often.
- CBR application is used to generate traffic. Packets of fixed size (512 bytes) are generated at a regular interval. To increase network load, the number of connections is increased. Connections are uniform, that is, any source-destination pair is equally likely. Network load is increased from 0.5 Mbps to 5 Mbps in steps of 0.5 Mbps. An additional point is taken at 10 Mbps to study the impact of heavy load on the network.
- Each simulation is run for a duration of 900 seconds, with an initialisation time of 100 seconds.
- For each data point, we carry out 20 simulation runs with different seeds.

Parameter	Value	Description
Number of Nodes	20, 40, 100	Number of nodes in the mesh
Node Placement	Random	Nodes are placed randomly over the square region
Node Degree	6, 12	The average number of neighbours of each node in the mesh
Terrain Dimension	1000 x 1000 to 50000 x 50000 sq. m	Terrain dimension varied to ensure node degree requirement
Link Bandwidth	6 Mbps	The radio link bandwidth (IEEE IEEE 802.11a base-rate)
Tx Power	18 dBm	The transmit power of the radio
Rx Sensitivity	-88 dBm	The minimum Signal-to-Noise ratio required for the radio to decode the received signal
Antenna	Omni-directional, Fixed beam directional	Omni-directional for IEEE IEEE 802.11 and TDMA mesh. Fixed beamwidth of 60^o for Achilles.
MAC	IEEE 802.11, TDMA, STDMA	MAC layers used in the simulation. Achilles uses STDMA
TDMA Slot Duration	8 ms	Duration of a transmit/receive slot used in TDMA and Achilles
Routing Protocol	Distributed Bellman-Ford	Routing protocol used by IEEE 802.11 and TDMA mesh. Achilles uses static routes calculated using Dijkstra shortest path algorithm.
IP Queue Size	100 packets	The size of the IP queue. The queue is FIFO queue.
Packet Size	512 bytes	Size of packet generated by CBR application
CBR Packet Arrival Interval	100 ms	Interarrival time between CBR packets
Traffic Type	Uniform	Each source-destination pair is equally likely. All connections generate an equal amount of load
Network Load	0.5 – 10 Mbps	Total traffic load generated by all nodes in the network
Simulation Time	900 s	The duration of the simulation

Table 6.2: Simulation parameters used in evaluation

6.3 Throughput

We evaluated the achieved throughput of the network as a result of increasing traffic load on the network. In general, when the network is lightly loaded the throughput is close to the offered load. However, with an increase in load the throughput saturates, reaching the network capacity. In general, contention for radio channel and delays introduced by queueing, affect the achievable throughput. As the average number of hops in the network increases, the throughput decreases due to contention faced at multiple nodes in the path. In a contention free MAC scheme, such as TDMA, where transmission is slotted, the throughput is limited by the fact that the opportunity to transmit is divided amongst the nodes in the network. Thus, each node can transmit only for a fraction of time. STDMA improves on TDMA by allowing nodes that are spatially separated to transmit at the same time, thus increasing the opportunity to transmit, and therefore, the network capacity.

6.3.1 Throughput for Network of 20 Nodes

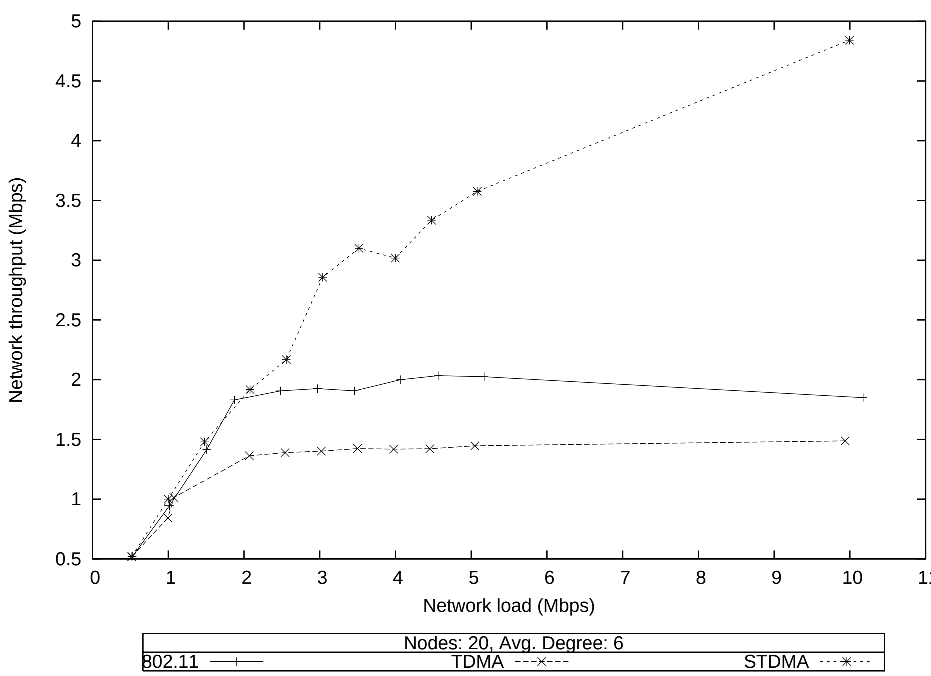


Figure 6.1: Throughput for a network of 20 nodes with an average node degree of 6

We first look at the throughput of the network with 20 nodes with an average node degree of 6 (c.f. Figure 6.1). With IEEE 802.11 the peak throughput achieved is 2 Mbps. This is one-third of the link bandwidth of 6 Mbps. In IEEE 802.11 networks, the throughput is limited by the behaviour of the IEEE 802.11 MAC which is susceptible to interference from far away nodes, a phenomenon identified by Li et. al. in [15]. Even though the network is of low density, the large interference range of nodes (almost twice the transmission range) results in excessive back-offs in all the nodes in the networks.

The throughput achieved by TDMA is 1.5 Mbps. The reason for the low throughput is that each node in the network has an opportunity to transmit only once every 20 slots (1/number of nodes). While there is no contention at the MAC layer resulting in back-off, the link bandwidth available to each node is 0.3 Mbps (6/20 Mbps). Nodes use this available bandwidth to transmit their own packets, as well as to route packets for other nodes in the network. In a low density network, the average distance to neighbours (hopcount) is larger, this results in increased routing burden on each node in the network, consequently, low network throughput.

The peak throughput achieved by STDMA is about 5 Mbps. In fact, there is further capacity available in the network as shown by the rising curve. The throughput is more than twice that achieved by IEEE 802.11 and TDMA. The reason for the higher throughput than IEEE 802.11 is that STDMA is contention free at the MAC layer and is not susceptible to MAC layer back-offs. The improvement over TDMA is a result of the fact that multiple links are active during each time slot in STDMA, while in TDMA only one link is active in each time slot.

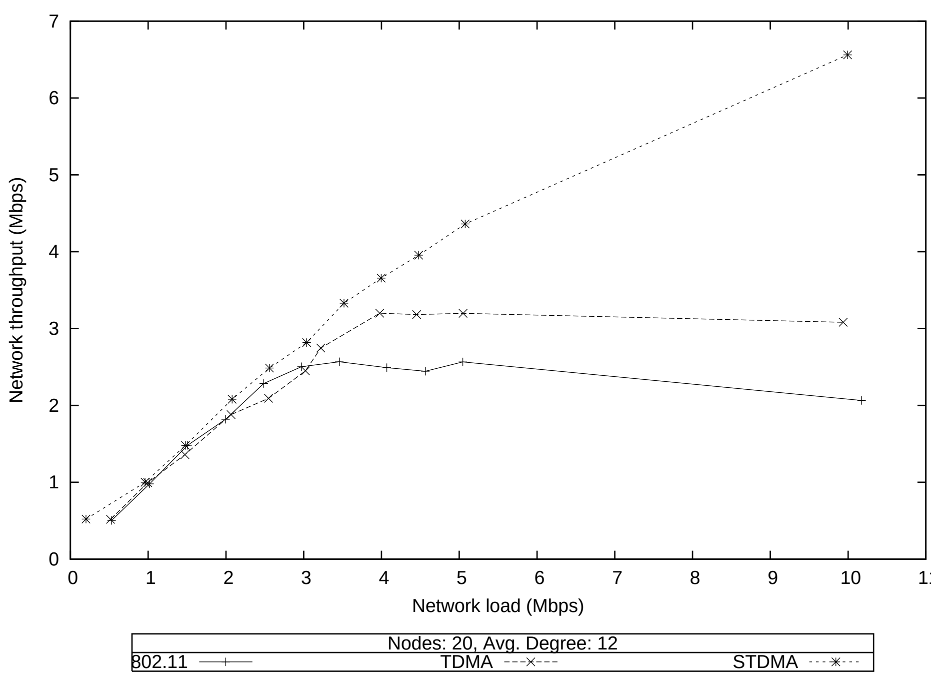


Figure 6.2: Throughput for a network of 20 nodes with an average node degree of 12

In Figure 6.2, we show the results for the higher density network with an average node degree of 12. Since the network has 20 nodes, for each node more than half the nodes in the network are direct neighbours and therefore, the average hop count is lower. It is clear that the increase in density improves the network throughput for all the three MAC schemes. TDMA benefits the most from the increase in density as the routing load on nodes is reduced. The peak throughput achieved by TDMA is increased to 3 Mbps (from 1.5 Mbps in 6 neighbour case). IEEE 802.11 suffers from increased interference in a high density network. The benefit from reduced hop count is cancelled by the increased interference and thus the peak throughput (2.5 Mbps) remains almost the same as for 6-neighbour case (2 Mbps).

6.3.2 Throughput for Network of 40 Nodes

Next we look at the throughput performance in a network with 40 nodes as shown in Figure 6.3. As expected, the performance of IEEE 802.11 has degraded. Increased interference from the additional nodes in the network results in increased contention at the MAC layer. The peak throughput achieved by IEEE 802.11 has now dropped by 1.5 Mbps. TDMA too suffers from the increased number of nodes. This is primarily because now each node has only 0.15 Mbps link bandwidth (6/40). TDMA achieves a peak throughput of less than 1 Mbps, a fall from 1.5 Mbps achieved in 20 nodes, 6 degree case. On the other hand, STDMA throughput remains practically the same at 5.5 Mbps. Again, this is due to the fact that multiple links are scheduled in each slot and therefore, the performance is better than TDMA. However, it is important to note that while the network throughput remains the same, the per-node throughput has decreased (divide network throughput by number of nodes). This is true for all the MAC protocols, and is a consequence of the fact that the average hop count increases in a larger network, requiring more number of intermediate nodes to route packets for other nodes. A fact well illustrated by Gupta and Kumar in [12], where they show that even under optimal circumstances the per-node throughput scales according to the equation 6.1, where $\lambda(n)$ is the per-node throughput and n is the number of nodes in the network.

$$\lambda(n) = \Theta\left(\frac{link\ bandwidth}{\sqrt{n}\ log\ n}\right) \tag{6.1}$$

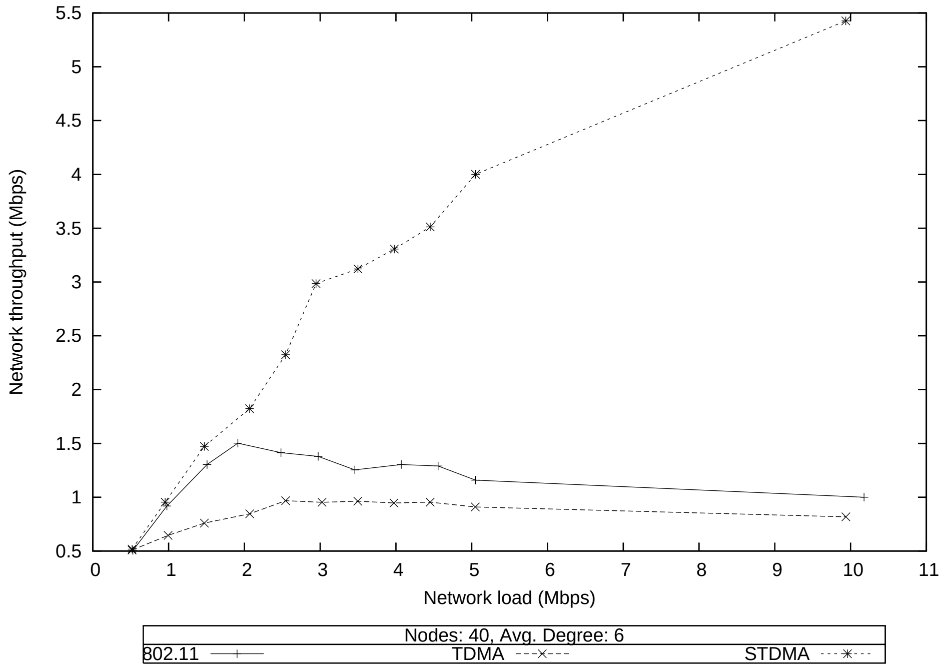


Figure 6.3: Throughput for a network of 40 nodes with an average node degree of 6

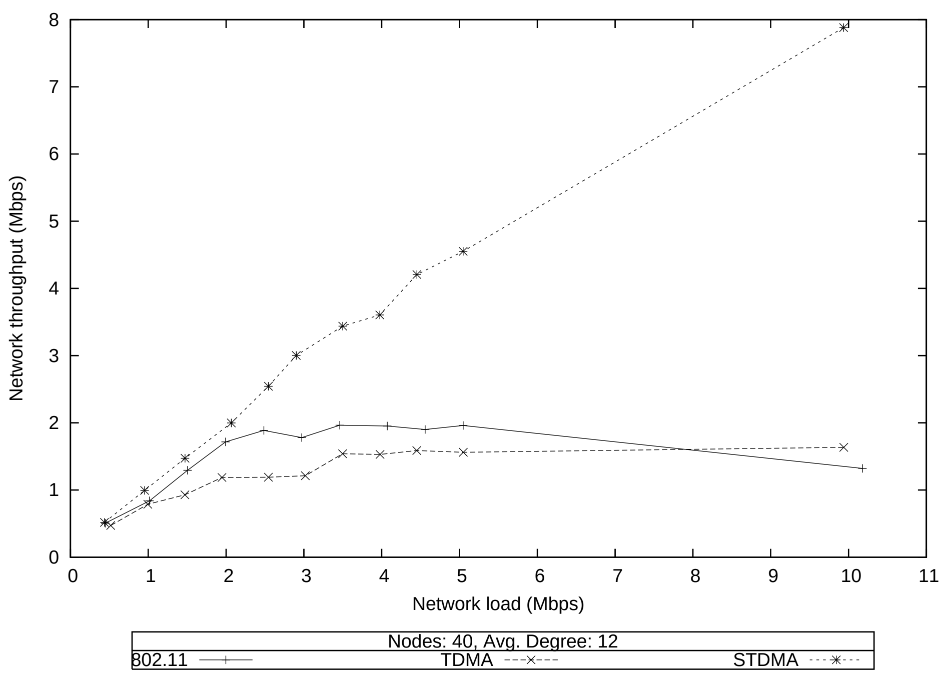


Figure 6.4: Throughput for a network of 40 nodes with an average node degree of 12

In Figure 6.4 we show the result when the average node degree is increased to 12. The increased density, and consequently the smaller average hop count, improves the throughput. Both IEEE 802.11 and TDMA show slight improvement, an increase of about 0.5 Mbps, However, STDMA shows an marked improvement, from 5.5 to 8 Mbps. This can be attributed to the fact that STDMA being contention free benefits directly from the reduction in average hop count, while in IEEE 802.11 the increases interference negates the possible benefits from the reduction in hop count. TDMA shows a slight improvement, the limiting factor still being the fact that each node can only transmit once every 40 slots, unchanged from the 6 degree case.

In Figure 6.5, the results from the 100-node simulations are presented. TDMA shows the worst performance, a result of increased number of nodes in the network, further depriving each node of the chance of transmit, now once every 100 slots. The peak throughput of TDMA is limited to 0.4 Mbps. IEEE 802.11 performs better than TDMA with a peak throughput of 1.3 Mbps. The throughput of STDMA drops to 4.5 Mbps, a slight decrease as increased number of nodes means that the number of slots-per-frame has increases, thus the probability of a link to be active in a slot decreases. However, STDMA outperforms both IEEE 802.11 and TDMA by a margin of more than 3 Mbps.

Another important point to note is that IEEE 802.11 and TDMA performance saturate rapidly, at a load of just 2 Mbps, but the STDMA throughput continues to grow, and there is room for higher throughput. However, due to the limited amount of load simulated, we are unable to determine, at this time, the saturation point for STDMA.

6.3.3 Throughput for Network of 100 Nodes

Next, we show the results for the 12-degree case in Figure 6.6. As with the previous cases, the throughput shows an improvement attributed to the reduction in average hop count. TDMA derives the minimum benefit with the throughput rising by 0.1 Mbps to 0.4 Mbps. IEEE 802.11 increased to 2 Mbps from 1.3 Mbps. The TDMA performance is limited by the increased number of slots-per-frame (100 slots-per-frame). Any benefit derived from the reduced number of hops is negated by the minuscule transmission window allocated to each node (1 in 100 slots). IEEE 802.11 benefits slightly, again a trade-off between the reduction in average hop count and increase in interference due

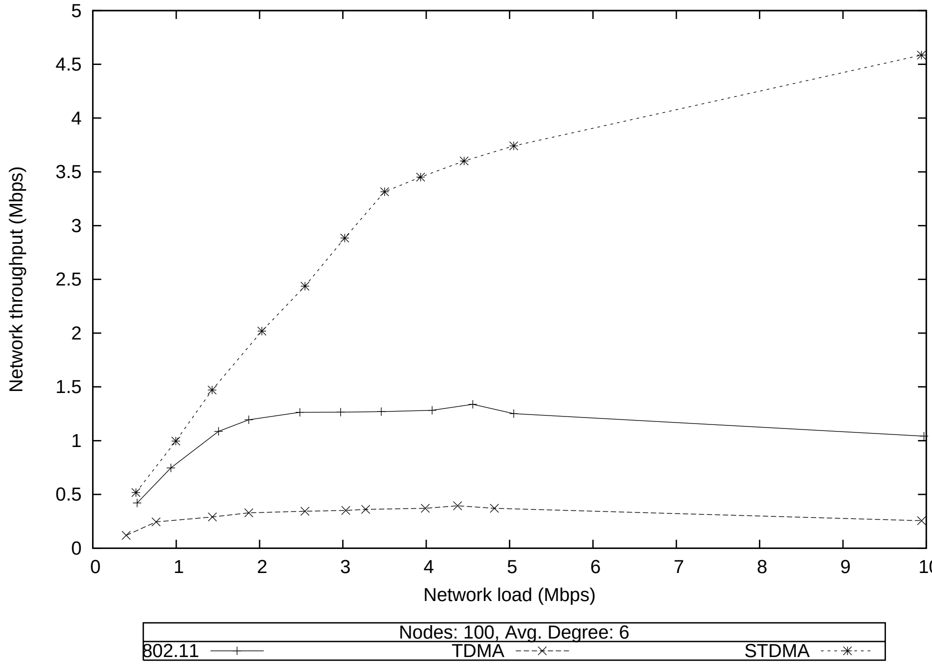


Figure 6.5: Throughput for a network of 100 nodes with an average node degree of 6

to high node density. STDMA derives the maximum benefit from the reduction in hop count and shows almost a doubling of the peak performance to 8 Mbps from 4.5 Mbps in degree 6 case.

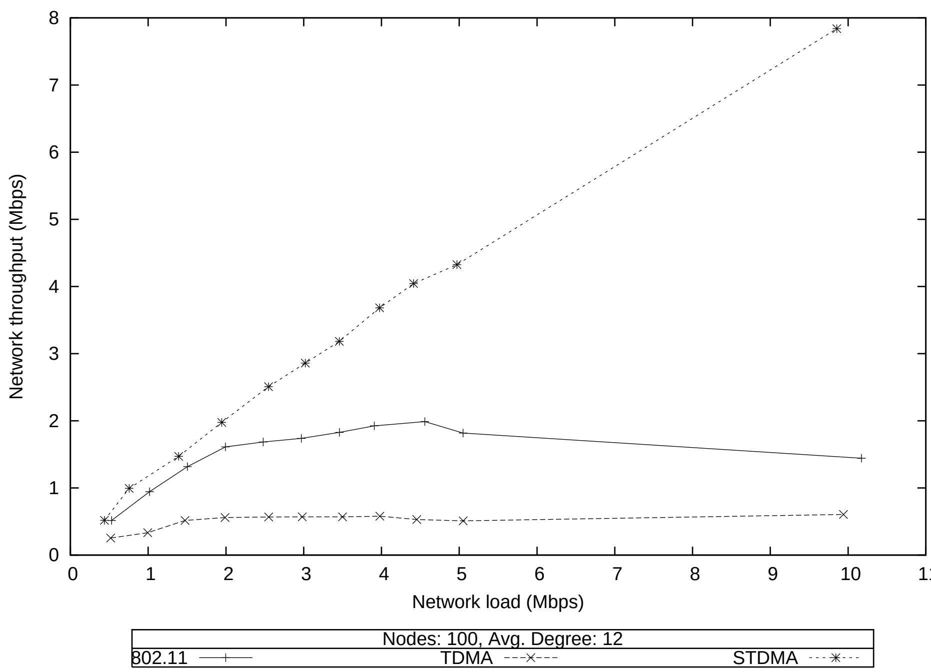


Figure 6.6: Throughput for a network of 100 nodes with an average node degree of 12

6.3.4 Summary of Throughput Results

Table 6.4 tabulates the peak throughput results from the simulations. It is clear that in all the cases studied, STDMA delivers the highest peak throughput with a gain of two to three times over IEEE 802.11 and TDMA. The increased capacity is a result of the fact that STDMA is contention-free, unlike IEEE 802.11 and is able to schedule more than one transmission per slot, unlike TDMA. These two facts enable STDMA to deliver a much higher throughput. STDMA uses the radio resources, in space, and time, more effectively than the alternatives. While STDMA is unable to, as with any other MAC scheme, alleviate the reduction in per-node throughput with increasing network size, it is able to sustain a high throughput even when the network size grows. IEEE 802.11 and TDMA suffer from increased interference and reduced transmission opportunity, respectively, and show a reduction in network throughput with increase in network size.

Nodes	Avg. Degree	IEEE 802.11 Peak Tput(Mbps)	TDMA Peak Tput(Mbps)	STDMA Peak Tput (Mbps)
20	6	2	1.5	5
20	12	1.5	1	5.5
40	6	1.5	1	5.5
40	12	2	1.5	8
100	6	1.3	0.45	4.5
100	12	2.0	0.57	7.9

Table 6.4: Summary of Throughput vs. Load performance

6.4 Delay

In this section, we study the delay performance of the three MAC schemes. As mentioned before, the delay metric is the average time taken for the packet to reach the destination node. The delay can be attributed to two components: i) queueing delay at source and each intermediary node and ii) propagation delay. The propagation delay is negligible when compared to the queueing delay. The queueing delay is the result of packets waiting in the node's queue for a chance to be transmitted. In general, for IEEE 802.11, the more the contention in the network, the larger is the queueing delay. For contention free MAC such as TDMA and STDMA, the queuing delay is the result of node waiting for its transmission slot. Another factor for queueing delay is the number of other packets already in the queue (queue size). The more the number of queued packets, the larger is the delay experienced by any one packet. An increase in traffic load increases the queue size of the nodes, thus increasing the delay in the network.

In a contention based MAC, such as IEEE 802.11, the queueing delay continues to increase with increase in contention. This is because a packet queued at the MAC layer waits for the radio to transmit; if there is a lot of contention in the network, the opportunity to transmit might take a long time. On the other hand, in contention-free MAC such as TDMA and STDMA, each node is assured a transmission opportunity after a fixed period. This has the benefit that the delay is upper bounded. Thus, contention-free MAC show an increase in queueing delay with increased load, and then finally settling down at an upper bound (when the queue of all the nodes in the network is full, resulting in packet drops).

Next, we will look at the results from the simulations carried out to study the delay performance.

6.4.1 Average Delay for Network of 20 Nodes

For a network of 20 nodes and an average degree of 6, the delay performance is shown in Figure 6.7. It is clear that for IEEE 802.11 and TDMA, the delay grows rapidly. While the large delay values appear surprising on first look, they are consistent with the slope of throughput vs. load curve (c.f. Figure 6.1) beyond 2 Mbps load for IEEE 802.11 and 0.5 Mbps load for TDMA. The network has reached the saturation point in the throughput-load curve and thus there is a rapid increase in the delay due to queueing at nodes. For IEEE 802.11, being contention based, delay continues to grow as the load is increased, the major component of the delay being the queue time at MAC layer, a result of back-offs. On the other hand, for TDMA, where the nodes are assured transmission at regular intervals, the delay grows rapidly and then settles down to a value of 25 seconds. Once all the network layer queues are full, further load results in packet drops at the IP queue and do not contribute to additional delay.

STDMA performs much better than IEEE 802.11 and TDMA. The delay remains below 0.5 seconds when the network load is 2.5 Mbps. Beyond that, there is a gradual increase in the delay, settling down around 5 seconds. Again, as with TDMA, nodes using STDMA are assured transmission at regular intervals and therefore delay is upper bounded. The upper bound for STDMA is lower than TDMA as more than one node can transmit in each slot, resulting in the interval between transmissions being shorter than TDMA.

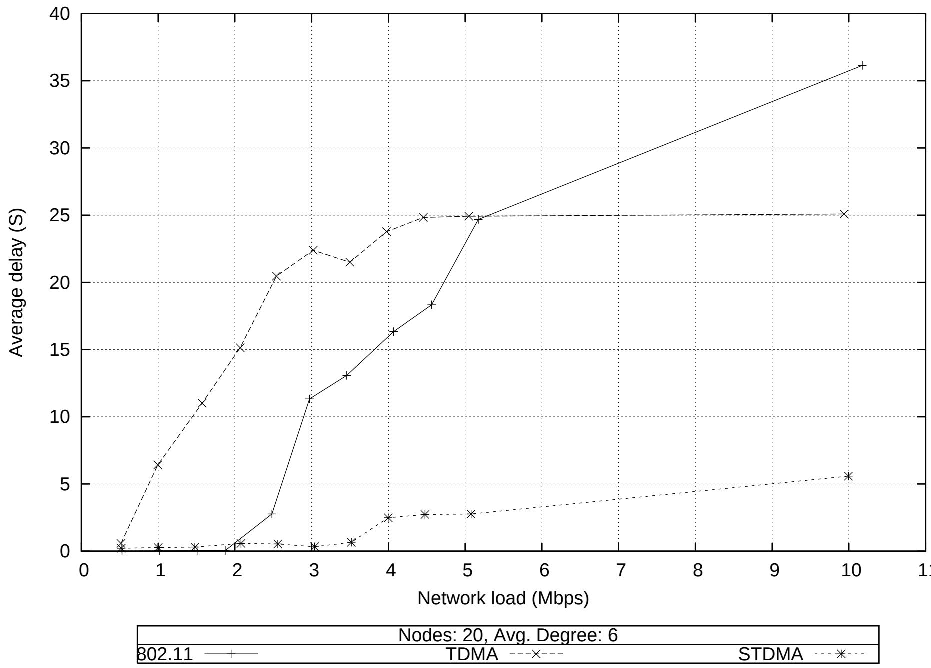


Figure 6.7: Delay for a network of 20 nodes with an average node degree of 6

In the case of 20 nodes with degree 12, there is a decrease in the delay due to reduced number of hops, as shown in Figure 6.8. For TDMA the delay starts to increase rapidly after 1 Mbps load, compared to 0.5 Mbps in degree 6 case; for IEEE 802.11, the turning point has increased to 2.5 Mbps from 2 Mbps, both consistent with the increase in network capacity as shown in Figure 6.2. An important observation is the increase in delay for STDMA beyond the 2.5 Mbps point, compared to 3.5 Mbps point in degree 6 case. The reason for this increase is the fact that increased node density results in longer schedules as less number of nodes are able to transmit in the same slot [refer to table showing the schedule for 20, 12 case].

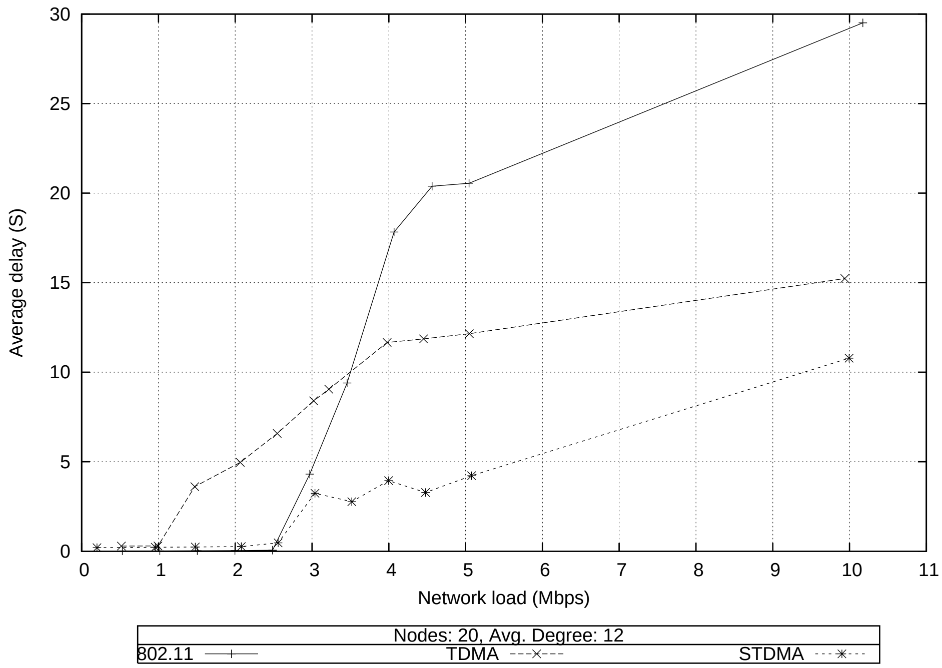


Figure 6.8: Delay for a network of 20 nodes with an average node degree of 12

6.4.2 Average Delay for Network of 40 Nodes

Figure 6.9 shows that there is a general increase in the delay for all three MAC schemes, compared to 20 nodes case. An increase in the average hop count is the primary reason. For TDMA, we can see that the average delay, even in lightly loaded condition, has increase to around eight seconds as nodes have an opportunity to transmit only once every 40 slots.

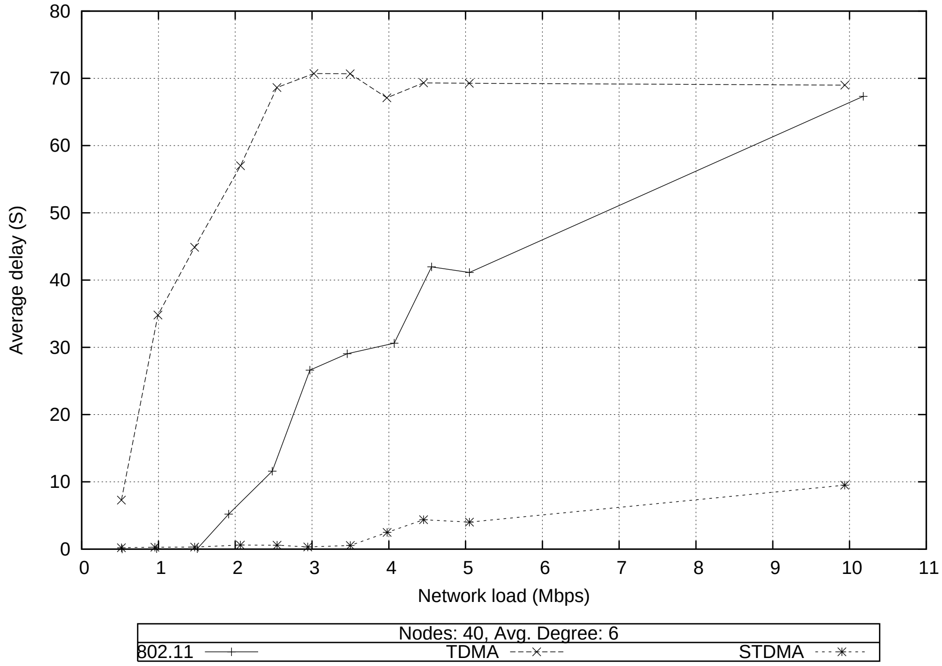


Figure 6.9: Delay for a network of 40 nodes with an average node degree of 6

Figure 6.10 shows a drop in delay compared to Figure 6.9. The increase in density, and the consequent reduction in average hop count, leads to reduction in delay.

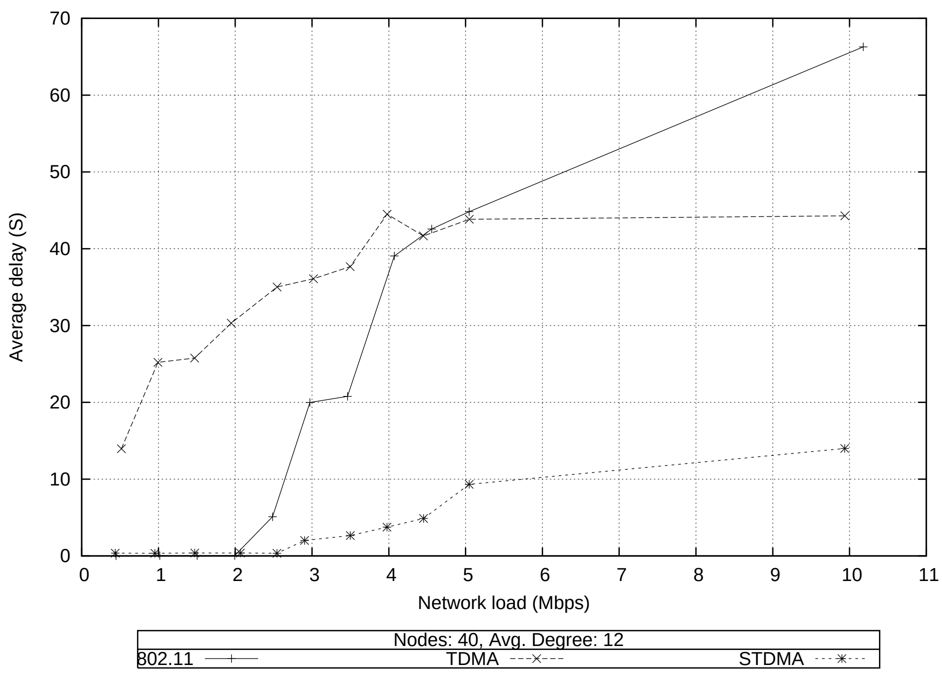


Figure 6.10: Delay for a network of 40 nodes with an average node degree of 12

6.4.3 Average Delay for Network of 100 Nodes

On increasing the node numbers to 100, there is a general increase in the delay in both TDMA and STDMA, by about two times as shown in Figure 6.11. This is expected as the TDMA schedules become longer and the average hop count increases with number of nodes. The large delay values (more than 120s) observed for TDMA are a consequence of the increase in the TDMA frame size to 100, and also, the fact that the capacity of the network is very low and there is extensive queueing at nodes.

An interesting point to note is that the delay performance of IEEE 802.11 remains practically the same as in the case of 40 nodes. A possible explanation of this phenomenon is that with increasing terrain size (note, since we keep the average density same, an increase in node numbers means larger terrain), there is better spatial reuse in IEEE 802.11. The impact of interference from distant nodes is reduced and more number of nodes can transmit simultaneously. This hypothesis is further strengthened by the observation that the delay increases as soon as we increase the density, as shown in Figure 6.12.

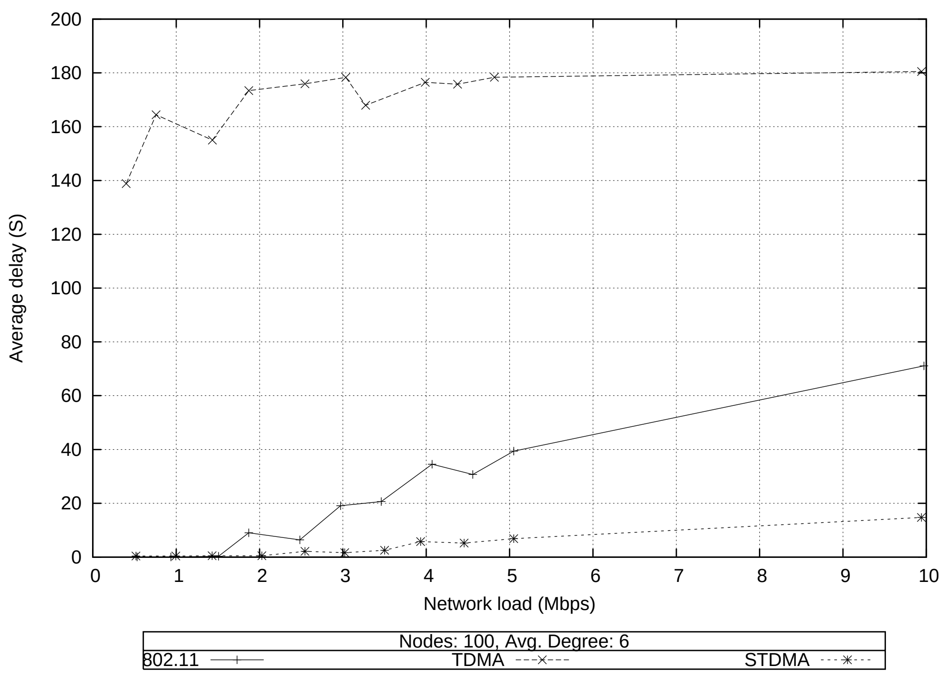


Figure 6.11: Delay for a network of 100 nodes with an average node degree of 6

Finally, Figure 6.12 presents the results for the 100 nodes, degree 12 case. Again, STDMA and TDMA benefit from the reduced hop count. IEEE 802.11 on the other hand shows a worsening of performance, as explained before.

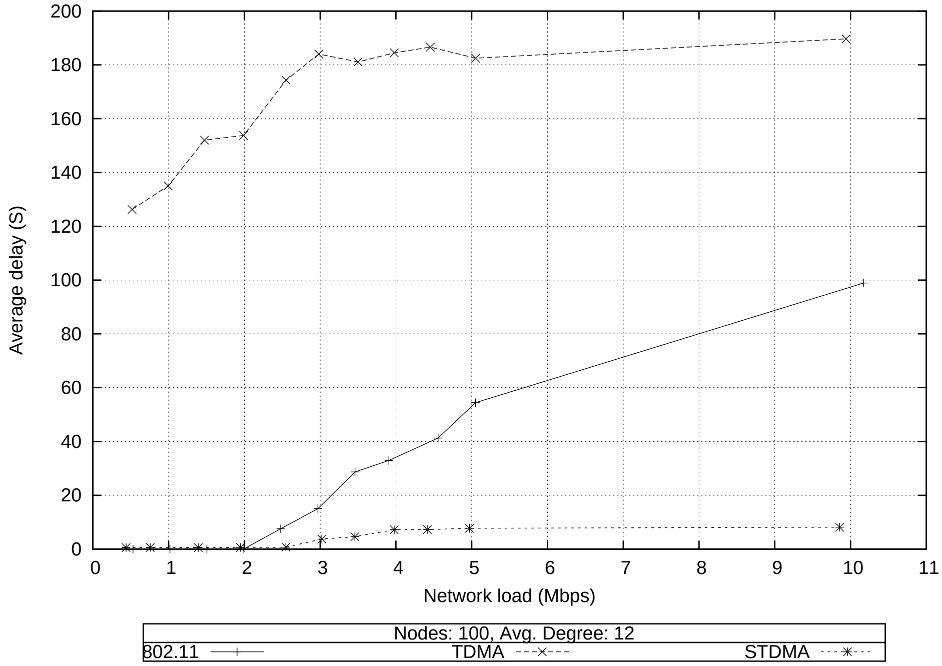


Figure 6.12: Delay for a network of 100 nodes with an average node degree of 12

6.4.4 Summary of Delay Results

From the measurements of average delay, as summarised in Table 6.5, it is clear that TDMA has the worst delay performance, with the only benefit being that the delay is upper bounded. IEEE 802.11 has low average delay at low traffic loads, however, due to limited capacity of the network, the delay starts growing rapidly after about 2 Mbps load. STDMA, on the other hand, has excellent delay performance showing very low delay values for loads up to 3 Mbps and acceptable delays up to 4 Mbps. As with TDMA, the delay for STDMA is upper bounded and the bound is much lower than that of TDMA. TDMA is unable to scale with growing number of nodes as the TDMA frame becomes longer and the delays get larger, resulting in large delays even when the traffic load is low. On the other hand, at low traffic loads, IEEE 802.11 is unaffected by the increase in number of nodes and delays start growing as the contention in the network increases due to more nodes contending (increase in traffic load) or increased interference (due to increased node density). STDMA scales very well with increase in network size as it takes advantage of the space diversity to schedule multiple transmissions in each time slot, thus allowing nodes to transmit more often.

Nodes, Degree	Average Delay in seconds								
	1 Mbps			5 Mbps			10 Mbps		
	A	B	C	A	B	C	A	B	C
20, 6	0.03	0.57	0.15	23.83	17.84	0.37	32.97	17.76	0.52
20, 12	0.02	0.29	0.22	20.54	12.15	4.22	29.5	12.22	4.78
40, 6	0.05	34.79	0.26	41.14	69.27	4.0	67.32	69.98	9.49
40, 12	0.03	25.20	0.35	44.82	43.83	9.31	66.29	44.28	13.98
100, 6	0.04	164.45	0.38	39.37	178.39	6.83	71.07	180.52	14.71
100, 12	0.02	134.99	0.57	54.39	182.51	7.71	98.89	189.67	8.14

Table 6.5: Summary of Delay Results for 1, 5, and 10 Mbps traffic load. The columns A, B, and C represent IEEE 802.11, TDMA, and STDMA, respectively. All delay values are in seconds.

6.5 Packet Delivery Ratio (PDR)

In this section, we study the PDR performance of IEEE 802.11, TDMA and STDMA. As mentioned before, PDR is calculated as the ratio of the packets received at the destination node over the packets transmitted at the source node. In general, PDR decreases with increasing load on the network. For low load conditions, the PDR is close to one.

Packets are lost in the network due to i) packet collisions and ii) packet drops due to queue being full. Packet collisions only affect IEEE 802.11, whereas packet drops due to full queues affect all three MACs, and is the major contributor to packet loss.

6.5.1 PDR for 20 Nodes

As shown in Figure 6.13, the PDR for IEEE 802.11 drops rapidly after the 2 Mbps load point (0.9) and this is consistent with the fact that the throughput of IEEE 802.11 reaches network capacity at that point (c.f Figure 6.1). This is the knee-point in the throughput-load graph for IEEE 802.11. At 10 Mbps load, the PDR for IEEE 802.11 is only 0.18. For TDMA, the PDR starts dropping rapidly after the 1 Mbps load point (0.85), reaching a minimum of 0.15 at 10 Mbps load. STDMA on the other hand continues to have good PDR (greater than 0.75) as the load increases until about 4 Mbps , reaching a minimum of slightly above 0.5 at a load of 10 Mbps.

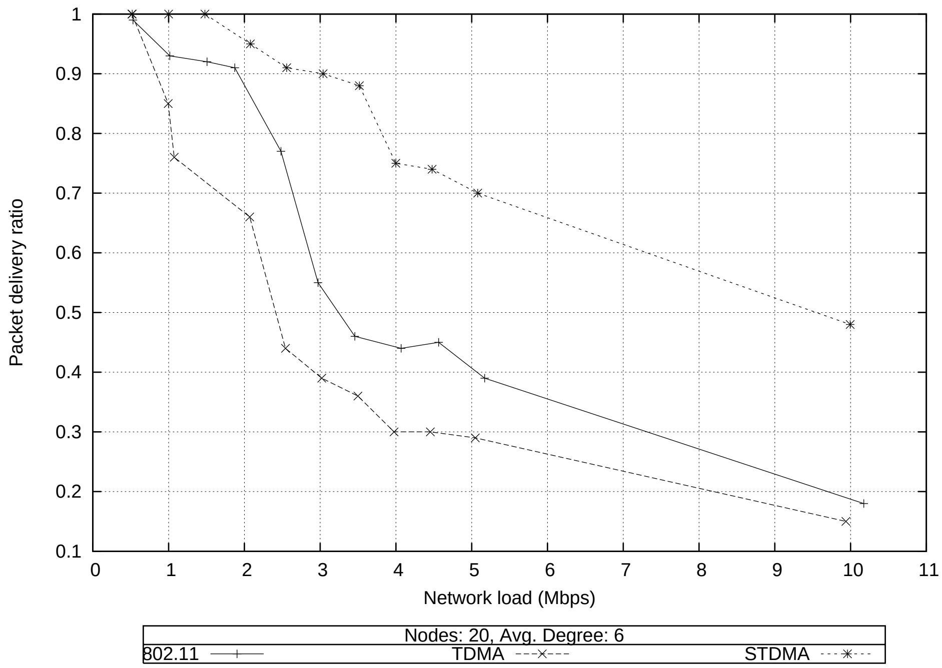


Figure 6.13: Packet Delivery Ratio for a network of 20 nodes with an average node degree of 6

With an increase in node density, we see an increase in PDR (cf. Figure 6.14. Again, this is attributed to a reduced number of hops, and therefore the chance of packet drops at intermediate nodes. All three MACs deliver packets with PDR greater than 0.8 for loads up to 3 Mbps. STDMA performs much better than the rest maintaining a PDR greather than 0.7 for loads up to 9 Mbps. This tallies well with the high throughput shown by STDMA.

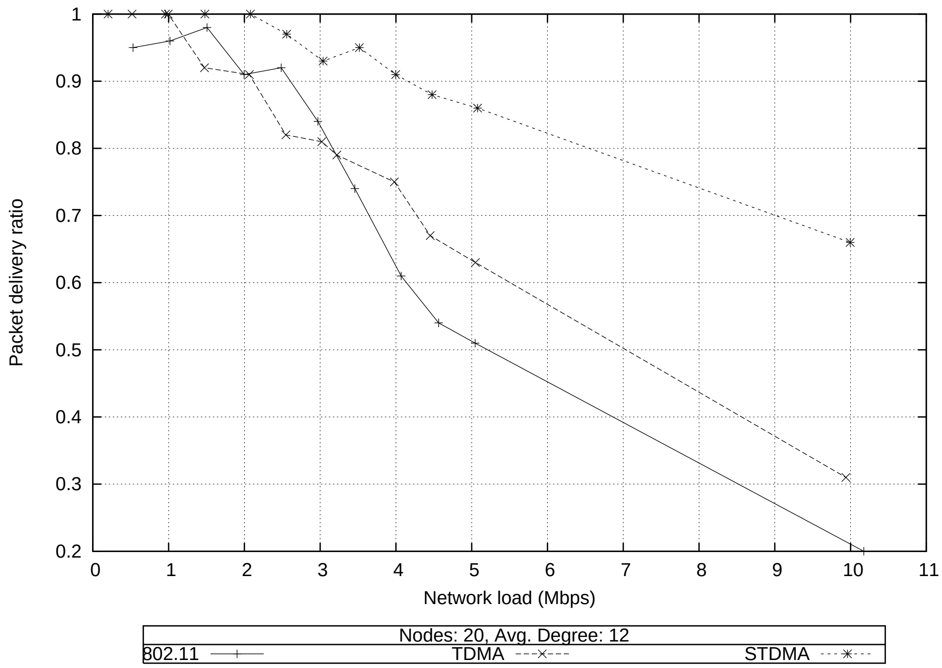


Figure 6.14: Packet Delivery Ratio for a network of 20 nodes with an average node degree of 12

6.5.2 PDR for 40 nodes

An increase in network size to 40 nodes has a detrimental effect on both IEEE 802.11 and TDMA (cf. Figure 6.15. IEEE 802.11 suffers from increased interference, and TDMA from the increased size of TDMA frame, resulting in queue build-up and eventual packet drops due to full queues. Acceptable performance for TDMA ($PDR > 0.7$) now drops to a load of 1 Mbps from 1.5 Mbps in 20 nodes, 6 degree case. IEEE 802.11 shows acceptable PDR till a load of 2 Mbps, again reduced from 2.5 Mbps in the 20 nodes, 6 degree case. STDMA still continues to outperform with an acceptable PDR of 0.7 at loads up to 7 Mbps.

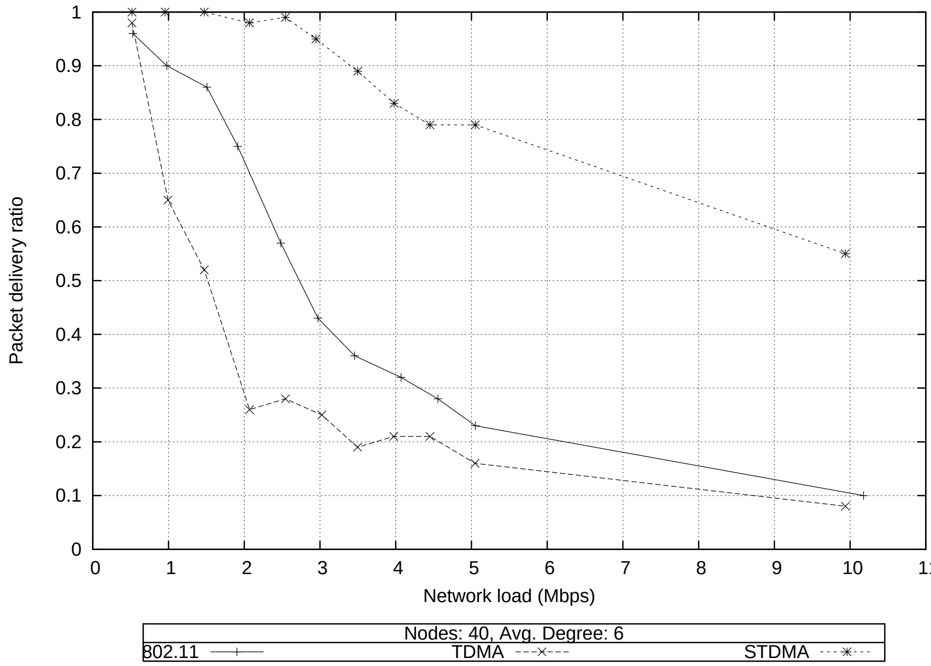


Figure 6.15: Packet Delivery Ratio for a network of 40 nodes with an average node degree of 6

While all three MACs benefit from the reduced hop count due to increased density (cf. Figure 6.16), STDMA derives the maximum benefit, maintaining a PDR greater than 0.8 up to the peak load of 10 Mbps.

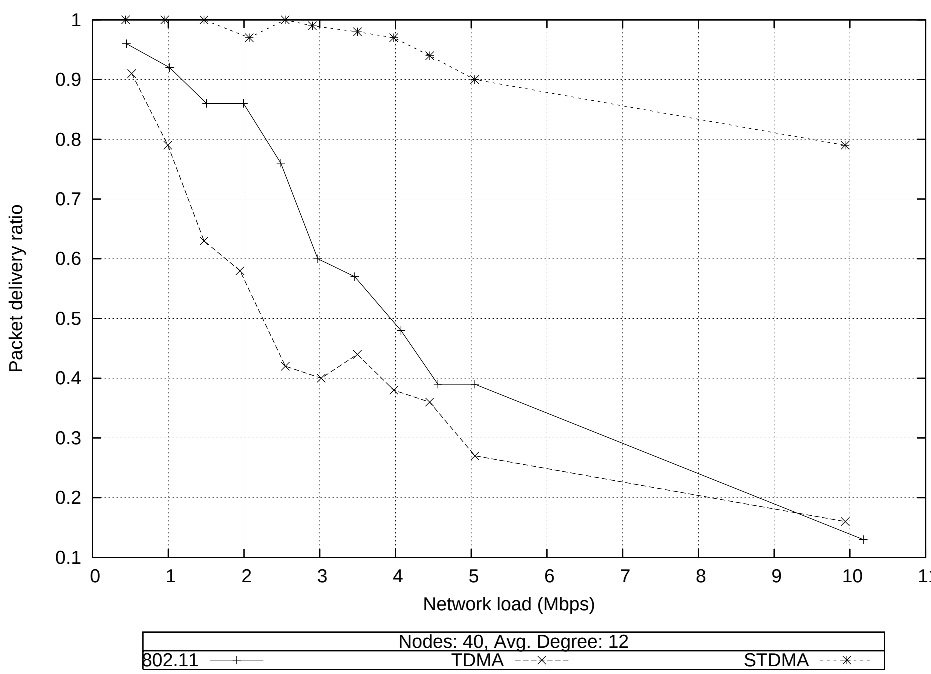


Figure 6.16: Packet Delivery Ratio for a network of 40 nodes with an average node degree of 12

6.5.3 PDR for 100 nodes

With an increase in node numbers to 100, the PDR of TDMA falls drastically as shown in Figure 6.17. This is consistent with similar drop in throughput and delay performance. TDMA with a long schedule of 100 slots, is unable to perform well due to large queue build-up at nodes, leading to large number of packet drops. IEEE 802.11 continues to work well for loads less than 1.5 Mbps, after which there is a sharp drop in the PDR as the network becomes congested, resulting in high contention and collisions. STDMA takes advantage of the spatial gain and performs well up to loads of 5 Mbps. There is a drop in performance from the 20 and 40 nodes case, which can be attributed to the fact that with 100 nodes the STDMA schedule, as with TDMA schedule, becomes larger, and therefore, network queues build up resulting in packet drops. Another point to observe is that for STDMA there is a sudden and large drop in PDR from the 3.5 Mbps load to 4 Mbps load point, after which it stabilises. We conjecture that this is point where the queue in some critical nodes (many routes pass through the node) in the network. At present we don't have a convincing explanation for this phenomenon.

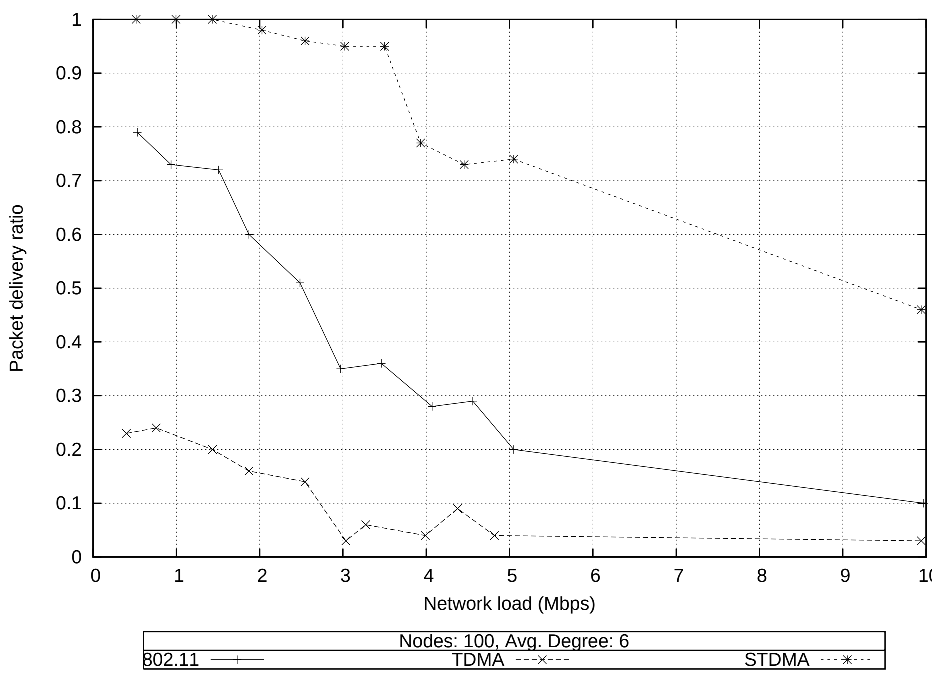


Figure 6.17: Packet Delivery Ratio for a network of 100 nodes with an average node degree of 6

Finally, Figure 6.18 shows that with increased density all three MAC protocols show an improvement in PDR. Again, STDMA delivers a PDR greater than 0.8 for peak load of 10 Mbps (an improvement of 0.3 from the degree 6 case).

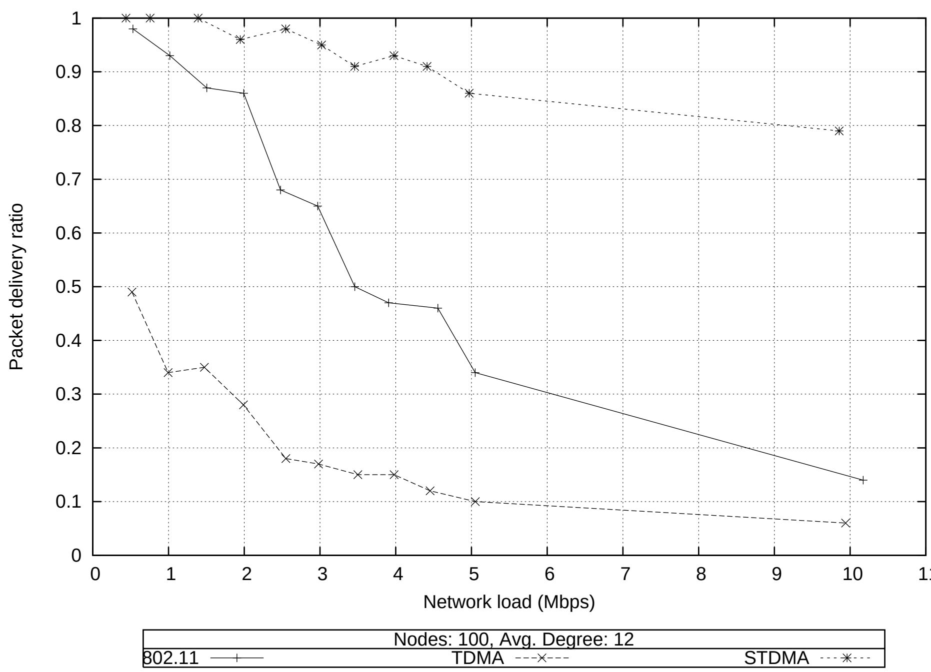


Figure 6.18: Packet Delivery Ratio for a network of 100 nodes with an average node degree of 12

6.5.4 Summary of Packet Delivery Ratio Results

The results from the PDR study is summarised in Table 6.6. It is clear that for less than 1 Mbps load, all three MACs perform acceptably ($PDR > 0.7$) under the simulated conditions. However, for increased loads beyond 3 Mbps, only STDMA delivers acceptable performance. The PDR of STDMA is high under all simulated conditions and it is able to take advantage of spatial diversity, as well as reduced hop counts in dense networks. TDMA's performance is the worst of the three, with IEEE 802.11 doing well when the traffic load is low, with rapid degradation in performance at increased traffic loads. The contention free nature of STDMA and the increased network capacity due to multiple transmissions per slot are in favour of STDMA and enable STDMA to show good packet delivery capability.

Nodes, Degree	Packet Delivery Ratio								
	1 Mbps			5 Mbps			10 Mbps		
	802.11	TDMA	STDMA	802.11	TDMA	STDMA	802.11	TDMA	STDMA
20, 6	0.93	0.84	1	0.39	0.28	0.7	0.18	0.14	0.48
20, 12	0.96	1	1	0.51	0.63	0.85	0.20	0.31	0.65
40, 6	0.90	0.64	1	0.23	0.15	0.79	0.1	0.08	0.54
40, 12	0.92	0.79	1	0.39	0.27	0.90	0.12	0.16	0.79
100, 6	0.73	0.24	1	0.19	0.04	0.74	0.1	0.02	0.74
100, 12	0.93	0.33	1	0.34	0.1	0.85	0.14	0.06	0.79

Table 6.6: Summary of Packet Delivery Ratio Results for 1, 5, and 10 Mbps traffic load.

6.6 Discussion

The simulation results prove without doubt that STDMA has superior performance over IEEE 802.11 and TDMA. In all three metrics, STDMA performs significantly better than the others. STDMA is able to deliver a peak throughput of almost 8 Mbps, compared to 2 Mbps by IEEE 802.11 and 1.5 Mbps by TDMA. A discerning reader might point out that it is possible to have up to 6 Mbps (link bandwidth) throughput by IEEE 802.11 if there is just one connection in the network; while this is true, our simulated scenarios ensure that there are more than one connection in the network (the minimum being 13). Multiple connections are a more practical scenario, and the results from multiple connection scenarios are more applicable to real world networks. Multiple connections lead to contention in the network and bring forth the contention related effects on throughput, delay and PDR.

To summarise the results from the simulation, we list down the major results:

- STDMA delivers the peak throughput with a maximum network throughput of almost 8 Mbps, compared to 2 Mbps for IEEE 802.11 and 1 Mbps for TDMA.

- STDMA's delay performance is better than IEEE 802.11 and TDMA, with delay less than 1 second in most low to medium load conditions, whereas IEEE 802.11 has low delay only at low network loads (1 Mbps) and much higher delay (greater than 20 seconds) at higher loads. TDMA performs poorly with delays above 10 seconds for practically all but the 20 nodes, light load condition.

- STDMA has very high PDR for most of the simulated conditions, staying close to one for low to medium loads. IEEE 802.11 on the other hand, shows high PDR for low load conditions, but suffers a great deal with increased load. TDMA's performance is the worst of the three.

Chapter 7

Conclusions and Future Work

In this thesis, we presented the design of Achilles, a high capacity wireless network utilising fixed-beam directional antenna and commodity radio hardware (IEEE 802.11a). We set out with the intention of creating a mesh network capable of covering a large terrain with minimal number of nodes. The use of directional antenna was necessary to achieve the goal. In addition, we designed the system so that it would benefit from the desirable properties of directional antennas, vis-a-vis the ability to transmit and receive in the intended direction. We picked Spatial TDMA (STDMA) as the MAC protocol for the system as it has the benefit of TDMA, being contention-free, and at the same time uses the radio resources efficiently by scheduling multiple transmission in the same time slot by effectively using the available space diversity.

7.1 Achievements

The main challenges we faced and the solutions we proposed are:

- **Antenna Selection:** Since we use multiple fixed-beam antennas to provide a 360^o coverage, we need a method to switch between the antennas. This was achieved using a RF switch controlled from the MAC layer. Using this switch, the node is able to select the antenna as required.

- **Neighbourhood Discovery:** With directional antennas, it is not assured that all the nodes within the radio range can discover the transmission from a node. It depends on the direction the node is transmitting in, and the direction the neighbouring node is receiving in. While a random antenna switching scheme can lead to neighbourhood discovery, being random in nature, there is no guarantee that all the neighbours would be discovered. This could lead to loss of network capacity. We solved the problem by proposing a deterministic discovery algorithm that ensures that all the nodes in the neighbourhood are discovered. This scheme depends on our deployment scenario, where mesh nodes are deployed in the target terrain and configured with three necessary parameters, i) the number of nodes in the mesh network, ii) the unique ID of the node, and iii) a network bootstrap time.

- **Topology Broadcast:** With directional antennas, transmission to all the neighbours at the same time (broadcast), is not feasible with a single transceiver and RF switch. In order to solve the problem of network-wide broadcast of information such as topology, we devised a scheme which ensured that a broadcast packet reaches all nodes in the network within a fixed duration. This scheme is used to broadcast the neighbour information from each node to all other nodes in the network. The neighbour information is used to calculate the STDMA schedule as well as routes. The scheme works by specifying an antenna switching algorithm at each node which ensures that a packet broadcasted by a node reaches all its neighbouring nodes within a fixed number of time slots.

- **Link Scheduling:** Next, we solved the problem of link scheduling. Link scheduling is required to determine which node transmits when and to whom. In order to use the spatial diversity (nodes which are far away can transmit at the same time without interfering) in an effective manner, we used the Spatial Time Division Multiple Access (STDMA) scheme. This scheme is able to schedule multiple transmissions in the same slot as long as they do not interfere with each other. In order to determine which links can be active at the same time, we devised a method based on link tests. This method is very practical as it does not require prior knowledge of terrain and propagation conditions (both of which are very difficult to obtain and are often not accurate, leading to over-engineering). In our proposed scheme, links are tested on the field after deployment to determine which links can be active at the same time. Our algorithm specifies the procedure to orchestrate the tests and to use the results to create a STDMA schedule.

We also specified a queue mechanism fo MAC, with multiple packet queues, and antenna switching algorithm that is able to use the STDMA schedule.

Having developed the schemes we performed extensive simulations to compare the performance of STDMA MAC used by Achilles, with IEEE 802.11 and simple TDMA. Our simulation results showed that STDMA outperforms both in every aspect (throughput, delay, and packet delivery ratio). This is expected as STDMA inherits the desirable qualities of TDMA, being contention-free, and at the same time ensuring short TDMA schedules by scheduling multiple transmissions in each slot. STDMA shows an improvement of two to three times over both IEEE 802.11 and TDMA.

In conclusion the design of Achilles has met the goal of creating a high capacity network using directional antennas for long distance communication. Our target deployment scenario of a maritime mesh network with mesh routers placed on buoys is well served by Achilles.

7.2 Future Work

We believe that Achilles provides a robust platform to develop a high capacity mesh network. In order to gain further improvements in the performance of Achilles, the following work can be undertaken:

- **Power Control:** The present design of Achilles does not take advantage of the capability of the hardware to control the transmit power. By using power control so that transmission power is limited to the minimum required to reach the destination node, we can pack in more transmission per time slot. This will further improve the capacity of the network by enhancing spatial reuse. A way to achieve this would be to use the *linktest* and *linkresult* messages to record the received power in the first run. In the second run, the transmitter should reduce the transmit power so that the received power is sufficiently above the sensitivity threshold. By performing the test in two runs, and recording the transmit power (along with antenna direction) in the neighbour table, unintended and unnecessary interference can be reduced, allowing more transmissions to be packed in each slot.

- **Traffic Adaptive Scheduling and Routing:** The current routing scheme used by Achilles uses Dijkstra's algorithm to calculate the shortest paths. The edges are weighted based on the the number of times they are scheduled in each STDMA frame. While this scheme performs sufficiently well when the traffic is uniform, it is not optimal when there are specific patterns in the traffic (some source-destination pairs are more probable than others). In order to optimise the performance, the traffic matrix (if already available) can be used as an input in the STDMA schedule generation phase, as suggested by Gronkvist, et. al. [11]. In addition, the routing protocol could assign weights to links based on the expected traffic to prevent congestion at specific links. Another enhancement would to be to use multiple routes for each source-destination pair.

- **TDMA Slot Duration:** In the current design, the STDMA slot duration is 8 ms. The limitation of firmware, kernel timers, and time synchronisation using the available GPS hardware place a restriction on the available timing resolution. By using a real-time kernel and more advanced GPS hardware (build specifically for timing applications), we can reduce the slot duration. A reduction in the slot duration will improve the delay performance of the network, critical for applications such as VoIP.

- **Mobility Support:** The current design of Achilles is for a static mesh network. However, the design can be extended to support limited mobility. The main challenge here is to detect mobility and to perform a limited link test in the affected zone, without requiring network-wide action.

Overall, the design of Achilles has proven to be a very valuable experience and we expect that it will provide a good base for both research in, and deployment of, wireless mesh network.

Bibliography

[1] H. R. Anderson. *Fixed Broadband Wireless System Design*. Wiley, 2003.

[2] L. Bao and J. J. Garcia-Luna-Aceves. Transmission scheduling in ad hoc networks with directional antennas. In *Proceedings of ACM MOBICOM*, Sept. 2002.

[3] V. Bhargavan, A. Demers, S. Shenker, and L. Zhang. Macaw - a medium access control protocol for wireless lanx. In *Proceedings of IEEE ICUPC'97*, pages 868–872, Sept. 1994.

[4] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *Proceedings of ACM MOBICOM, August 28 - September 2*, 2005.

[5] P. Bjorklund, P. Varbrand, and D. Yuan. Resource optimization of spatial TDMA in ad hoc radio networks: a column generation approach. In *Proceedings of IEEE INFOCOM*, volume 2, pages 818–824, 2003.

[6] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya. Using directional antennas for medium access control in ad hoc networks. In *Proceedings of ACM MOBICOM*, Sept. 2002.

[7] CM9. Cm-9 802.11a/b/g mini-pci card. `http://metrix.net/datasheets/CM9.pdf`.

[8] K. Dyberg, L. Farman, F. Eklof, J. Gronkvist, U. Sterner, and J. Rantakokko. On the performance of antenna arrays in spatial reuse TDMA ad hoc networks. In *Proceedings of IEEE MILCOM*, pages 270–275, Oct. 2002.

[9] J. Gronkvist. Traffic controlled spatial reuse TDMA in multi-hop radio networks. In *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 3, pages 1203–1207, 1998.

[10] J. Gronkvist. Assignment methods for spatial reuse TDMA. In *Proceedings of IEEE MOBIHOC*, pages 119–124, Aug. 2000.

[11] J. Gronkvist, J. Nilsson, and D. Yuan. Throughput of optimal spatial reuse tdma for wireless ad-hoc networks. In *Proceedings of IEEE Vehicular Technology Conference, VTC 2004-Spring.*, volume 4, pages 2156–2160, 2004.

[12] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[13] P. Karn. Maca - a new channel access method for packet radio. In *Proceedings of ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, Sept. 1990.

[14] Y. B. Ko and N. H. Vaidya. Medium access control protocols using directional antennas in ad hoc networks. In *Proceedings of IEEE INFOCOM*, pages 13–21, Mar. 2000.

[15] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. *Mobile Computing and Networking*, pages 61–69, 2001.

[16] Motorola. Motorola mesh network. `http://www.motorola.com/mesh/`.

[17] R. Nelson and L. Kleinrock. Spatial TDMA: A collision-free multihop channel access protocol. *IEEE Transactions on Communications*, 33(9):934–944, 1985.

[18] S. Networks. Qualnet home page. `http://www.scalable-networks.com`.

[19] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit. Ad hoc networking with directional antennas: A complete system solution. In *Proceedings of IEEE Journal on selected areas in communication*, volume 23, pages 496–506, 2005.

[20] T. S. Rappaport. *Wireless communications: principles and practice*. Wiley, 1996.

[21] M. Sanchez, J. Zander, and T. Giles. Combined routing & scheduling for spatial TDMA in multihop ad hoc networks. In *Proceedings of IEEE Wireless Personal Multimedia Communications*, volume 2, pages 781–785, Oct. 2002.

[22] M. Takai, J. Martin, and R. Bagrodia. Directional virtual carrier sensing for directional antennas in mobile ad hoc networks. In *Proceedings of ACM MOBIHOC*, June 2002.

[23] Tropos. Tropos networks. `http://www.tropos.com/`.

[24] S. Vasudevan, J. Kurose, and D. Towsley. On neighbor discovery in wireless networks with directional antennas. In *Proceedings of IEEE INFOCOM*, volume 4, pages 2502–2512, 2005.