CHALLENGES IN DIRECTIONAL ANTENNAS

BY

VINAY KOLAR

B.E., Bangalore University, 2000

THESIS

Submitted in partial fulfillment of the requirements for
the degree of Master of Science in Computer Science
in the Graduate School of
Binghamton University
State University of New York
2004

Accepted in partial fullfillment of the requirements for
the degree of Master of Science in Comptuter Science
in the Graduate School of
Binghamton University
State University of New York
2004

Dr. Nael Abu-Ghazaleh _____

Date _____

Department of Computer Science.

Abstract

Directional antennas have been proposed to improve the performance and capacity of Wireless MAC protocols for use in Mobile Ad hoc Networks (MANETs). Such antennas focus their beams in the direction of their receiver, allowing more of the signal power to be used in the direction of the transmission and simultaneously reducing the interference in other directions. However, several challenges and design issues arise that have no counterpart in omni-directional antennas – as a result, a large number of directional MAC protocols have been proposed.

The thesis has three primary contributions organized into three parts: (1) It analyzes the directional MAC behavior in chain topologies, identifying some interesting interactions with upper layer protocols; (2) It identifies a head of line blocking problem in directional antennas and proposes new queuing policy to address it; and (3) It suggests a mechanism to passively discover directional neighbors and use that information to optimize multi-hop routes in directional antenna systems.

In the first part, we study the interaction between a chain connection and an underlying directional MAC. The interaction of high level protocols with MAC in a MANET environment is often complex and unpredictable, resulting in unexpected behavior: such effects are well documented with Omni-directional protocols. Thus, we seek to understand whether similar interactions occur with directional antenna,

especially when we consider the unique aspects of operation that directional antennas introduce. The problem characterization provides insight into how to design directional MACs to improve the performance of such connections.

The second part identifies the inefficiency caused by FIFO Queuing mechanism while using directional antenna and proposes to use a different queuing policy which could take advantage of the channel utility factor provided by the underlying antenna system. Our results indicate that by using a greedy approach to schedule the packet which has the least wait time increases the overall throughput and end-to-end delay.

The third part targets the problem of the inability of directional MAC protocols to discover directional neighbors (those neighbors reachable by a directional transmission but not by an omni-directional one). More specifically, directional antenna system use omni-directional mode to facilitate broadcasts and to find neighbors whose direction is unknown. The omni-directional mode is used by the routing layers to find the route. Even though the range of directional transmission is much larger than omni-directional, the routing layers are forced to route the packets through omni-directional neighbors. The thesis proposes a new mechanism to update the routing layers with directional neighbors to find better routes. A directional DSR protocol is proposed and evaluated using such a mechanism. Preliminary results are encouraging, but comprehensive route maintenance and route error handling mechanisms needs to be added for the proposed protocol.

ACKNOWLEDGMENTS

It gives me immense pleasure to express my gratitude to all the people who helped me actualize this thesis. I am deeply indebted to my advisor, Dr. Nael Abu-Ghazaleh, for his guidance and support extended to me during the course of my research work. His creative suggestions along with his words of encouragement, patience and time went a long way in writing this thesis. My heart felt thanks to Nael. I consider it my good fortune to have got an opportunity to work with him.

I would like to thank all my friends in the "Lab 311" for providing a thought provoking and pleasurable environment during the course of this thesis. My special thanks to Sameer Tilak for his constructive ideas, which led to tremendous of progress of my thesis. I would also like to thank all my 'maga' friends in Binghamton.

All this would have not been possible without the invaluable encouragement of my parents and Smitha. Their constant moral support have been the foundations of all my accomplishments. Thanks for standing by me in all good and bad times! It would be incomplete without thanking my good-old-days friends of "BMSCE Mountaineering Club", who people taught me to strike a balance between and work and fun!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Mobile Ad hoc Networks, or MANETs, are an emerging class of wireless networks where mobile wireless devices interact with each other and cooperate on forwarding traffic. Such networks are needed in environments where no infrastructure exists or is accessible. Many applications can take advantage of MANETs including search and rescue operations, vehicular networks, disaster recovery operations, and others. Moreover, they are a critical component of emerging applications such as ubiquitous computing and sensor networks. The nature of communication and the absence of infrastructure dictates cooperation among the wireless devices for connectivity.

The wireless medium and the antenna system forms the physical layer in the ad-hoc network node. There are four other important layers above the physical layer that processes the packet received from the antenna. The Medium Access Control(MAC) layer is just above the physical layer and is mainly responsible for sensing the channel. It sends the packet only when the channel is idle, receives the packet from the antenna, checks for packet corruption and propagates the received packet to the upper layer if the node is the intended recipient. The routing layer is responsible for finding the routes, possibly multi-hop routes, to the destination and for directing the packet towards the final destination. The transport layer and the application layer have the functionality similar to the wired networks.

Typically wireless devices communicate with one another by using omni-directional antennas. These antennas radiate the signals in all directions resulting in a circular transmission/reception pattern. The existing MAC protocol used in ad-hoc community is well tuned for omni-directional antennas. There exists a family of specifications called 802.11 for MAC protocol using omni-directional antennas; while there are other MAC protocols for wireless communication, IEEE 802.11 is by far the most commonly used one. We overview IEEE 802.11 in Section 2.4.

The omni-directional nature of transmission propagates the signal in all directions away from the node; the signal is received by all the neighboring nodes (those within transmission range) surrounding the sender. Since a packet is usually intended for a specific receiver, it is not necessary for all the surrounding nodes to receive the signal. Such transmission pattern adds no advantage because the receiver gets only a small part of the energy of the omni-directionally transmitted signal. In fact, the remaining wasted energy also possibly interferes with other ongoing transmissions. If there is an antenna that can focus the beam only towards the receiver, then the nodes that are not in the direction of the receiver are free to go ahead with their communication. Such antennas which have the ability to focus the beam in a particular direction are termed as "Directional Antennas"

Avoiding transmission of signal in all the directions by focusing a beam yields larger free channel space around the sender. This provides *greater spatial reuse*; it allows more transmissions to go on concurrently without collisions. Furthermore, by focusing its beam, the sender sends with higher power in the direction of the receiver. This property can be utilized to get one of the two advantages. To reach the same receiver, the signal can be sent at a lower power, focused towards the receiver rather than sending an omni-directional signal with higher power. This *decreases the energy consumption* while transmission. The second advantage that can be utilized is

increasing the range of transmission. For a given transmission power, a focused beam can reach larger distance than an omni-directional beam; some receivers outside of omni-directional range can be reached in one hop using directional antennas.. This longer range results in a smaller number of hops on end-to-end paths, increasing connection throughput and reducing delay.

The above advantages have sparked interest in directional antennas. Using directional antennas leads to greater bandwidth utilization, lesser interference in undesirable directions, reduced energy consumption and greater range. Although these benefits look lucrative, there are significant challenges that must be addressed before they can be achieved in practice. More specifically, the properties of directional antennas require the design of a unique MAC level protocol that can take advantage of them. For example, a transmitter has to identify the direction of a receiver before it can reach it: a problem that is not present in omni-directional antennas. Another example is how the well-known hidden terminal and exposed terminal problems are different in directional antennas because of the different interference footprint. There are a number of other challenges unique to directional antennas; these are described in detail in Section 3.5. As a result, most directional antenna protocol research todate has focused on developing MAC layer support to take advantage of them. Since there is inadequate existing support from above lying network layers, evaluation of these developed protocols is mostly done using hand-crafted scenarios and hand configured routing.

In this thesis, we first analyze ad-hoc network behavior with directional antennas, and an existing directional MAC protocol. The goals behind this analysis are: (1) characterize the performance of directional antennas under realistic conditions and compare them to omni-directional antennas; (2) Understand the interactions that occur between higher level protocols and directional MAC; and (3) Identify sources

of inefficiency that occur in these scenarios and use this insight to propose solutions to them. The nature of the analysis is difficult because of the complex cross-layer interactions and the sensitivity to the node location geometry. The analysis in this thesis starts with a simple chain topology where nodes are arranged in straight line. Single connection runs from one end to the the other. Simple topologies are studied to isolate the problem. The topology is tweaked by altering the node positions and the effect of geometry of nodes is analyzed. The results from the simulation is analyzed and the problems are highlighted.

The thesis highlights some of the important characteristics of directional antenna and the disadvantages of using vamped up versions of omni-directional MAC protocol. Several new problems that is not existing in the system using omni-directional antennas are seen when the same variation of the protocol is used in conjunction with directional antenna. Major challenges are observed and analyzed in detail. In addition, we study and develop solutions to two of these problems.

The first problem occurs due to the queuing discipline used by existing MAC protocols. More specifically, in ad-hoc networks using omni-directional antennas, the packets are picked up from the queue in a strict priority based FIFO policy. This is not a problem for omni-directional antennas since there is a single state for the channel. If a directional antenna is used, this leads to sub-optimal channel reuse: if the direction in which the first packet needs to be sent is busy, then the other packets need to wait till the first packet is sent. This is true even if the other packets are destined to directions in which the channel is free. This problem is termed as "Head of line blocking". The thesis proposes new solution for this problem in the form of a new queue structure that allows the MAC layer to pick the earliest packet that can be transmitted (whose send direction is free). We show that this solution leads to an improved throughput, especially under high loads (which make head of line blocking

more common).

The second problem addressed by the thesis is routing in directional antennas. Since the range of directional transmission is significantly larger than the range of the omni-directional transmission, some nodes are reachable only through directional transmissions. Routing protocols rely on broadcast operations that are transmitted omni-directionally: these are not able to find directional neighbors. This leads to longer routes being used, which lead to loss of throughput and higher delay. We use a localized solution to address this problem that does not require additional overhead. Depending on geometry, nodes sometimes discover directional neighbors at the MAC level (by overhearing their transmissions). We make this information available to the above layer. This allows routing protocol to compact paths that use the newly discovered neighbors. For source routing protocols such as DSR, this can be done simply by eliminating intermediate hops to the newly discovered neighbor on paths that use it (by simply adding it to a link-based routing cache). Significant improvement in the overall throughput and end-to-end delay of the packet was observed by using the above approach.

The rest of the thesis is organized as follows: Chapter 2 gives an overview of Ad hoc network and the omni-directional antenna systems. A description of the directional antenna, the MAC protocol used, the motivations and the problems that exist in directional antenna systems are described in Chapter 3. The related work in directional antenna is described in Chapter 4. Chapters 5 and 6 analyzes the chain topology. The ineffectiveness of FIFO queuing under directional antennas is described in Chapter 7 and a new queuing policy is proposed to solve the "Head of line blocking" problem. A method to passively discover directional neighbors and use the information for shortening routes is described in Chapter 8. The conclusions are summarized in Chapter 9.

Chapter 2

BACKGROUND

2.1 Overview

This chapter introduces the area of Ad hoc networking. The omni-directional antenna model, which is used by and large in ad hoc networks, is presented. The popular MAC protocol, 802.11, is described in Section 2.4. A brief introduction to the functionality of routing layer is summarized a brief introduction to the DSR routing protocol is provided.

2.2 Ad hoc networking

Wireless mobile networks are envisioned to play an important role in the next generation cyberspace. The current generation has already seen deployed mobile wireless devices like laptops and palmtops being connected to internet. These networks consists of mobile end hosts which are wireless in nature and which communicate to wired internet through a base station. The base stations act as a gateway to the wired network for the mobile end hosts. This kind of networks need significant infrastructure like base stations that can be reached in just a single hop. If the mobile node moves away from the range of all the base stations then the connectivity is lost. Such networks are called as “Infrastructured networks” or “Single hop wireless networks”.

An alternate kind of wireless networks have grasped the interest of the wireless community. This kind of network lack the infrastructure of wired nodes, routers or gateways. The wireless nodes form a network by acting as routers if needed. They determine the routes to other nodes in a dynamic fashion and co-operate in forwarding the packets to the desired destination by multi-hop wireless routes. Such network are called as the "Infrastructureless Networks" or "Mobile Ad hoc Networks" (MANETs). MANETs advantage over last hop aid several applications like sensor networks, rescue operations and military networks. The nature of Ad hoc networks will suit the requirements of such applications. Research in Ad hoc networks have been very active from past decade. There research in this area are also under the name of *packet radio* and *multi-hop networks*. A good introduction is explained by Ramanathan et al. [21] and Royer et al. [22]. The following sections describe the antenna model and the MAC characteristics used in Ad hoc networks.

2.3 Omni-directional antennas

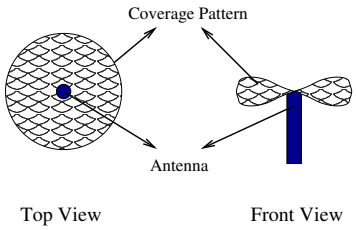


FIG. 2.1. Omni Directional Coverage

Omni-directional antennas were briefly described in section 1. Such kind of antennas are capable of transmitting and receiving 360° around the node (Figure 2.1[1]. The shape of this coverage pattern is also known as torus-shaped or donut-shaped. For a given power, the omni-directional antenna's transmission can be heard

[1]The Figure 2.1 is adapted from the tutorial [23].

by another omni-directional antenna for a specific range of distance. For altering this range, the transmission power is adapted. In Ad hoc networks the range of the antennas will be a few 100's of meters. Specialized networks like sensor networks have more limited range.

It is to be noted that the omni-directional antenna is different from the isotropic antenna. In case of isotropic antenna, the antenna is capable of receiving and transmitting in a spherical shape where as the coverage is torus-shaped in case of an omni-directional antenna as shown in Figure 2.1. It makes sense to use omni-directional antenna in Ad hoc networks because the nodes are not too far apart along the Z-axis.

2.4 MAC Layer: 802.11

There is a family of specifications by IEEE called 802.11 which describes the MAC and the physical (PHY) layer functionality in Wireless LAN. This set of specifications describes the MAC and PHY layer functionalities for different WLANs like infrastructured wireless networks using base stations. The same specification is used in the Ad hoc network also. However certain features which are specific to infrastructured networks are not enabled in Ad hoc networks. This section describes about the MAC layer functionality of the 802.11 specification.

The 802.11 MAC layer specification specifies two kinds of access methodologies as follows.

1. Point Coordination Function(PCF): This is usually used for real time data transmission with priorities in infrastructured networks. This is a contention free access protocol. This is not used in Ad hoc networks. This is not discussed in detail in the document.

2. Distributed Coordination Function(DCF): All Ad hoc networks use DCF as the access methodology. This is a contention based access protocol. This section

describes the features of DCF.

2.4.1 Carrier Sensing

In case of wired networks, the channel access is done by Carrier Sense Multiple Access with collision Detection(CSMA/CD). The node which wants to transmit will first sense the channel. If the channel is busy, then the transmission is withheld, else the packet is transmitted. If collision is detected, then the packet is retransmitted after exponential backoff. This holds good in wired networks where the each node can hear every other node. In Ad hoc wireless networks, this assumption does not hold good. Another factor for which the CSMA/CD has not been used in Ad hoc network is because of the commercial reason. The antennas used in WLANs usually can either transmit or receive but cannot do both simultaneously. For CD to work, the channel must be sensed even while transmitting. Antennas with such capability are expensive. Hence, in wireless networks *Carrier Sense Multiple Access with Collision Avoidance*(CSMA/CA) is used. The carrier sensing works in the same way as in wired internet. The node that wants to transmit a packet will listen to the channel. If the channel is idle for a certain duration of time called *Double Inter-Frame space*(DIFS), then the node assumes that no other node nearby is trying to transmit and goes ahead with its transmission. The recipient of the packet will acknowledge(ACK) the packet if the packet is received without any errors.

2.4.2 Virtual Carrier sensing

Consider the example in Figure 2.2 where node X and node Z wants to send a packet to node Y. Node X will start the transmission of the packet. Node Z is not within the range of node X and hence cannot listen to node X's transmission. If node Z assumes that the channel is free and starts transmitting its packet, then

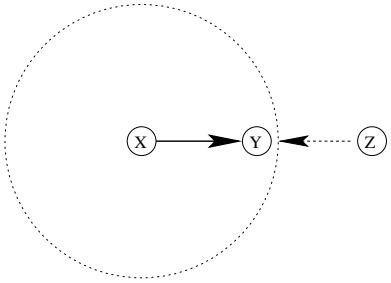


FIG. 2.2. Need for virtual carrier sensing

it can collide with the ongoing X-Y communication. Node Z is *hidden* from node X-Y's communication. Such problem is termed as *Hidden Terminal Problem.* Hence, a basic ACK scheme with CSMA/CA will not work satisfactorily. To overcome this problem, a scheme called as "Virtual Carrier Sensing" was proposed. The motto of this scheme is to let the neighboring nodes of sender and receiver know about the ongoing transmission. It is accomplished by sending small control packets before sending the data.

When a sender wants to transmit a packet, it will send a control packet called **Request to Send (RTS)** to the receiver if the channel is sensed as free. Upon receiving the RTS, the receiver will send the **Clear to Send(CTS)** back to the sender if the physical channel is not busy. Then the sender sends the **DATA** packet to the receiver. Upon correct reception of the DATA packet, the receiver will send the **Acknowledgment (ACK)** packet to the sender. This kind of signaling helps to prevent the hidden terminal problem described above. This idea was first proposed in MACA protocol [13] and was later refined by MACAW protocol [3]. Before sending the RTS, the node will sense the channel for DIFS time period. For sending other packets, the node will sense for smaller time called *Short Inter Frame Space(SIFS).*

Each node maintains a table called as the **Network Allocation Vector (NAV)** which aids Virtual Carrier sensing. The RTS and CTS frames contain a *duration* field in their header which specifies the time interval needed to complete the complete RTS-

CTS-DATA-ACK handshake. Any node that listens to these packets will update the NAV as described below. Upon listening to this RTS packet, the neighbors of the sender will know that one of the node near them is about to transmit data and also the duration of handshake. During this duration, if the node sends some packet, then it may interfere with the ongoing communication. Hence it will mark the time in duration field in its NAV. This time is the period for which the node needs to be silent. When any node listens to CTS, it updates its NAV in the same manner.

2.4.3 Exponential backoff

Backoff has been used as a very effective tool in networks to solve the contention of the channel. In Ad hoc network, the node senses the medium for DIFS period before sending the RTS. If the channel is observed to be busy then the node will remain silent for a random number of *slots* and will set its backoff timer accordingly. This random number is chosen from 0 to a maximum value called *Contention Window(CW)*. If the node listens to any other transmission during this backoff time, then the backoff timer is frozen and it will be restarted when the channel becomes idle again. If the node does not get any kind of acknowledgment (CTS for an RTS or ACK for a DATA), then the backoff timer is exponentially increased. It is doubled every time such situation occurs. The backoff timer is capped at a maximum value to avoid very high backoffs.

2.4.4 MAC layer reliability

Every time there is a failure to send the data to the receiver, either because of collision or because of failure to get acknowledgment packets (CTS or ACK), the backoff timer is increased exponentially as described above. The MAC layer tries to retransmit the same data packet for a given number of times given by a constant value called *Retransmit limit*. If the number of retransmissions exceeds the Retransmit

Limit, then the MAC layer gives up sending that data packet (a *Retransmit packet drops*) and reports the failure to the above routing layer. Such a packet drop is also known as *No RouTE error or NRTE*. Receiving such information typically causes the routing layer to initiate route repair mechanisms.

2.5 Routing layer

The *Routing layer* can be drawn parallel with the network layer of wired networks. In fact, it is also known as the Network Layer by some people. The end hosts in a wired network uses network layer to specify the destination. The routers take care of choosing the path to reach the destination. However, since all nodes act as routers in Ad hoc networks, its the responsibility of every node to route the packet. This added functionality makes the name "Routing layer" more suitable than just "Network Layer". This layer is responsible for dynamically finding routes to the other nodes, maintaining the routes and routing the packet towards the destination. The task of maintaining the routes with higher error rates and limited bandwidth. There are many protocols like *Ad hoc On-Demand Distance Vector Routing (AODV)* and *Dynamic Source Routing(DSR)* used in Ad hoc networks at the routing layer. Royer et al. give a brief overview about the variety of routing protocols in the Ad hoc networks in [22].

2.5.1 DSR protocol

This section describes briefly the working of the DSR protocol. DSR protocol is an *On-demand* routing protocol which will try to find the route only when needed. This avoids the periodic messages that are sent in other kind of routing protocols to discover and maintain the route. The packet carries the complete set of nodes through which the packet must flow as a field in its header.

Each node maintains a cache called as *Route cache* which stores the complete routes to different destinations. When a source needs to send a packet to a given destination, this cache is checked to find if the route exists. If the route is found then the packet is sent to the corresponding next hop with the complete path in its header. The next hop node is responsible to forward the packet to its next hop. This continues till the packet reaches the destination.

Route discovery: If the source does not contain the route to the given destination, then it will trigger a route discovery mechanism. The source will broadcast a packet called as the *Route Request(RREQ)*. The nodes that listen to the RREQ packet will first check their route cache if there is already a route available to the destination. If it is present in the route cache then it will reply back to the sender about the complete route. This message is called the *Route Reply(RREP)*.

If there is no route in the route cache then it will re-broadcast the RREQ to its neighbors adding its address as a partial route. Upon the reception of this RREQ, the neighbors of this node will try to find the route in its route cache. This process continues till:

- *Any intermediate node has the route to the destination:* In such a case the node adds its entry to the route and RREP is sent to the node that has broadcasted the RREP. These RREP are unicast messages that will ultimately reach the source.

- *The RREQ reaches the destination:* This happens when none of the intermediate nodes have the entry to destination in their route cache. The destination will now respond with RREP and will unicast it to the sender of the RREQ.

In this manner, the route discovery is invoked. If the destination is unreachable, then the source times out and assumes that the destination cannot be reached and

will discard the packet.

Route maintenance: Consider a scenario in which the route existed to the destination and some part of the link is broken, either because of the mobility of nodes or because of the network failure. In such a case, the intermediate node that could not send the packet to its next hop will send a *Route Error(RERR)* message to the source through the reverse order of the links in which the packet had arrived. Upon reception of RERR message, the source will delete the entry in its route cache that it had used to reach the destination and will try to find an alternative route in the route cache. If no such route exists, then the source will invoke a route discovery mechanism again.

As an optimization, the protocol can invoke a procedure called *Local Repair or Packet salvage*, where the intermediate node that experienced link failure will try to search the route by route discovery to find alternative path from the intermediate node. If this repair procedure is successful, then the intermediate node will inform the source about the new path used to reach the destination. The source and other intermediate nodes will update their route cache upon reception of such a message.

Chapter 3

DIRECTIONAL ANTENNA IN AD HOC NETWORKING

3.1 Overview

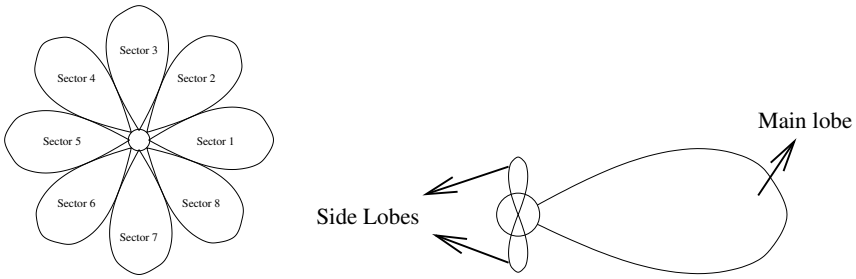
This chapter overviews "directional antenna" basics, the problems and challenges they introduce in ad hoc networks, and existing efforts to address them. Several features of directional systems that are different from the omni-directional systems are summarized. Sections 3.2 and 3.3 explain the antenna model and the MAC protocol that are currently being used. The advantages of directional transmission relative to omni-directional transmission are identified. The chapter concludes by presenting several challenges that needs to be tackled for using the advantages provided by such antennas.

3.2 Directional antennas

Directional antennas have the ability to direct the beam in a particular direction. In this section, we describe the types and operation of directional antennas and then discuss their advantages in a MANET environment.

3.2.1 Types of directional antennas:

There are two major types of directional antennas:



(a) 8 sectors. (b) Single lobe.

FIG. 3.1. Coverage pattern for a switched beam antenna

1. **Switched Beam Antennas (SBAs):** In the case of SBAs, the area around the antenna system is divided into a fixed number of equal-size sectors. Each antenna element transmits a beam such that covers one sector. Hence for an n-sectored SBA, there are n antenna elements covering $(\frac{360}{n})^\circ$ sectors each. An example of the coverage pattern of an 8-sectored SBA is shown in Figure 3.1(a). The sectors shown in the coverage pattern are not ideal circular sectors: the transmission pattern of each antenna is a *lobe*. Figure 3.1(b) shows the typical coverage pattern for a single sector. As we can see from the figure, the coverage pattern consists of a main lobe and two side lobes. There is often a tail lobe as well; however these are more commonly found in steerable antennas. The simulations done in the thesis use a switched beam antenna that does not contain a tail lobe.

   SBAs can propagate a beam in one of the given lobes but cannot alter the angle of the lobe dynamically. The antenna can be visualized as n equally spaced fixed co-ordinated antennas, each of which can transmit a beam in a particular

direction. Switched beam antennas are cheaper and require less complexity than the steerable antennas.

Note that the lobes represent the transmission and reception gains and not just the transmission. In case of switched beam antenna, if a node is listening to the sender, then it will activate its antenna to such a lobe where the directional gain is the best. More about gain is explained in section 3.2.2.

2. Steerable antennas

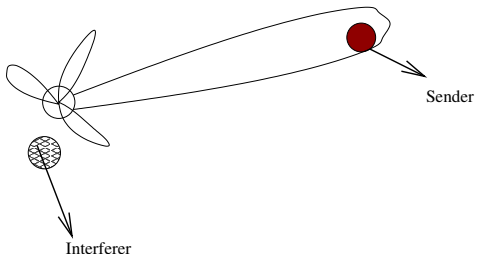


FIG. 3.2. Null steering in a Steerable Antenna

The main disadvantage of the switched beam antenna is the fixed nature of the beams. The beam cannot be focused to the precise angle of the receiver. There are intelligent antennas which are capable of doing the above mentioned task. Such antennas are called as Steerable Antennas. Even steerable antennas are made up of a number of antenna elements. The antenna system logic combines the antenna elements in such a way that the beam is directed towards any given angle. These antennas are also able to minimize the interference from the unwanted nodes. By mixing the antenna elements in such a way that main lobe, side lobes and tail lobe is not directed towards the interferer, the antenna reduces the interference. This is called as *Null Steering*. An introduction to null steering can be found in in the tutorial [23]. There are several algorithms for efficient and effective null steering in adaptive array antennas mentioned

in papers like [7, 9, 14]. Figure 3.2 shows a steerable antenna listening to the sender by mixing its antenna elements such that the main lobe points to the sender and the side lobes and the tail lobes avoid the interferer. The obvious drawback of such an enhanced antenna is the complexity and the higher cost. If the antenna needs to be more precise in direction and null steering, then there should be more number of antenna elements such that the set of combinations of antenna elements from which the antenna system is able to choose is larger. Such antennas are also known as *Adaptive Array antennas*. Details about adaptive antennas can be found in the paper [7].

It is also assumed that both the switched beam and steerable antennas have a fixed frame of reference. The antenna does not lose the frame of reference if the node on which it is mounted is rotated. This means that the sector which points to a geographic direction will continue to pointing to the same direction even if the node is rotated. For more details about the antenna systems and their features, interested readers are referred to the following tutorial [23].

3.2.2 Gain of directional antennas:

We need to define the term "Directivity"(G_d) before explaining the gain of a directional antenna. According to the terminologies in [25] , *"The directivity of a wireless antenna is given by the ratio of the maximum radiation intensity (power per unit solid angle) to the average radiation intensity (averaged over a sphere). The directivity of any source, other than isotropic, is always greater than unity."* This means that even omni-directional antennas have directivity. The only difference is that in case of an omni-directional antenna, the coverage pattern is torus-shaped. This pattern creates uniform gain in the X-Y plane. In case of directional antenna, the directivity is higher than the omni-directional antenna in the direction of transmission

because of its ability to focus the beam.

For a lossless antenna, the gain of the antenna is equal to the directivity of the antenna. For a real antenna with losses like reflection losses due to impedance mismatch and side lobe power dissipation, the Gain(g_d) can be defined as the product of directivity and efficiency(η). The IEEE standard 145-1993 provides a more comprehensive definition of the terms [11].

$$g_d = \eta G_d \tag{3.1}$$

The transmission gain of the directional antenna is denoted as g_t, the reception gain as g_r and the omni-directional gain as g_0. From the property of directional antennas, we can state that

$$g_t \geq g_0 \tag{3.2}$$

$$g_r \geq g_0 \tag{3.3}$$

Because of a greater gain in directional antennas, the signal transmitted with directional antenna with some power will be able to reach a larger distance than a signal that is transmitted with an omni-directional antenna. Note that the gain is does not have any unit because it represents a power ratio. However, gain is often expressed in decibels (dB). If g is any of the above gains discussed, then the equivalent gain in decibels(g_{db}) is given by:

$$g_{db} = 10 \log_{10}\ g \tag{3.4}$$

Detailed discussion about the gain of directional antenna can be found in the technical paper [8].

3.3 Directional MAC (DMAC)

There is no standardized MAC layer specification which describes directional antenna operation. However, papers like [1, 2, 6, 10, 15, 18] have come up with MAC protocols for directional antenna. These are reviewed in detail later in this thesis. The implementation we have worked with in this analysis is very closely related to the work done by Choudhury et al. in [6]. This protocol, called Directional MAC (DMAC), is provided in Qualnet simulator [20] that was used for the experiments in this thesis.

The DMAC works with RTS-CTS handshake similar to the 802.11 protocol. Since the transmission is focused in a particular direction, the protocol needs a mechanism to store the angle in which the beam should be focused to reach an intended receiver. This is enabled by maintaining a table called *Angle of Arrival(AoA) cache*. To enable virtual carrier sensing, there is a need to keep track of the channel state for each of the sectors separately because the channel can be split into several sectors. This makes *Directional Virtual Carrier Sensing(DVCS)* different than the omni-directional Virtual carrier sensing. DVCS is discussed in detail by Takai et al. in [24].

In order to support broadcasts, and for transmissions to nodes whose direction is unknown, the directional antenna is also capable of transmitting in the omni-directional mode. When the antenna transmits in omni-directional mode, then the advantages of directional gain is absent. Thus, there is a difference in reachability when the directional antenna transmits in directional and omni-directional mode. This limits the ability of DMAC in discovering directional neighbors. When the node is idle, it receives in the omni-directional mode. This is to enable the node to receive from all the directions. The gain of the receiver in omni-directional mode is lesser than gain of a receiver which is focused directionally. The remainder of this section describe the new mechanisms that are in place to enable directional

communication.

3.3.1 Angle of Arrival (AoA) Cache

To enable directional communication, DMAC needs to know the direction in which the receiver is located. To find out the angle of the next hop, the DMAC uses caching of the $\langle node, angle \rangle$ pair. This cache is called as the "Angle of Arrival Cache" (AoA cache). For every transmission heard by a node, an appropriate tuple is either added or updated into the cache. If the node does not listen to any signal from another node for the given amount of time, the entry in the cache is marked as stale and will be purged.

Before sending the packet, the AoA cache is queried to get the angle recorded for the next hop. If the cache does not have an entry for the next hop node, then the packet is transmitted in an omni-directional mode. Otherwise, the packet is transmitted in the angle fetched from the cache. The node tries to transmit in the directional mode for a fixed number of times. If the number of consecutive failures to transmit the packet directionally exceeds this threshold, then the cache entry is purged and the packet is transmitted omni-directionally.

3.3.2 Virtual carrier sensing

In 802.11, Virtual Carrier sensing is done by maintaining a "Network Allocation Vector"(NAV). In case of DMAC, virtual carrier sensing needs to be altered to take advantage of the spatial reuse provided by the directional antenna. If a node listens to an ongoing transmission in a particular angle then an appropriate space of channel around that angle should be marked as busy. This is done by maintaining a "Directional NAV"(DNAV) table. The angles around the node that is marked busy is given by the "DNAV delta angle"(δ_{dnav}). If the angle of arrival is aoa, then the space

marked as busy will be $(aoa - \delta_{dnav})$ to $(aoa + \delta_{dnav})$. This segment of the channel is marked as busy for the given duration of time. The angle $(aoa - \delta_{dnav})$ is called as the *lower bound angle*(lb) and the angle $(aoa + \delta_{dnav})$ is called as the *upper bound angle*(ub).

The entries in the DNAV store the lb, ub and time at which the wait expires. For each signal overheard a new DNAV entry is created in the DNAV table. The wait time for the entry is marked as the duration for the transaction. When a packet needs to be transmitted directionally, the angle at which the packet needs to be sent is first retrieved from the AoA cache. The DNAV table entries are queried to get the maximum wait time for that particular angle.

Let a_t be the angle in which the packet needs to be transmitted. Let E_{sel} be the entries selected in DNAV for a given a_t. Let $DNAV_j$, lb_j and ub_j be the j^{th} DNAV entry, lower bound and the upper bound angle for the j^{th} entry respectively. Then,

$$E_{sel} = \{DNAV_j\} \ \ such\ that\ ((lb_j <= a_t) \wedge (ub_j >= a_t)) \tag{3.5}$$

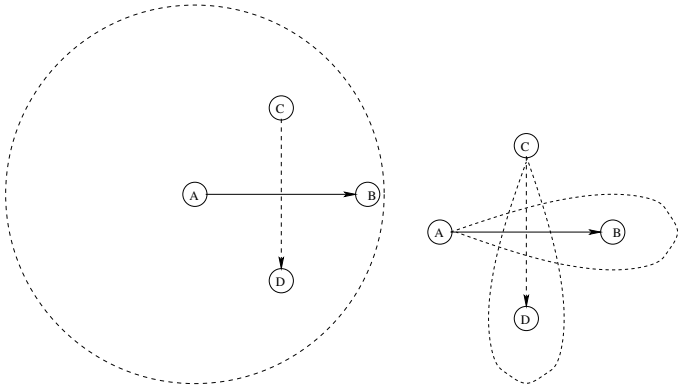
Let w_i be the wait time for the i^{th} entry in the set E_{sel} and n be the number of entries in the E_{sel} as given Equation 3.5. Let W_{max} be the maximum wait time for a given a_t. It is given by the Equation 3.6.

$$W_{max} = max(w_i)\ where\ i \in 1..n \tag{3.6}$$

More details about directional virtual carrier sensing is explained by Takai et al. in [24].

3.4 Motivation to use directional antennas in Ad hoc networking

In this section, we describe the key advantages of directional antennas. To motivate the use of directional antennas in Ad hoc networking let us consider two simple examples as explained in the subsections below which explains few of the main advantages of directional antenna.



(a) Omni directional mode. (b) Directional mode.

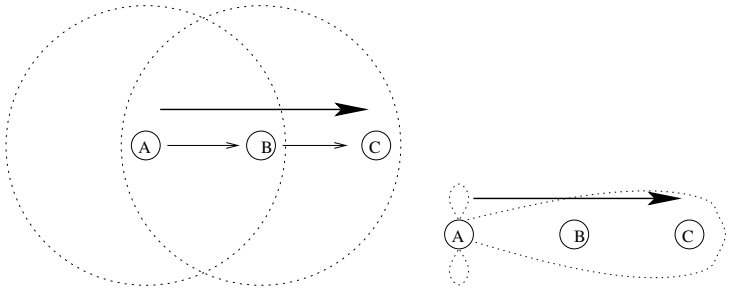
FIG. 3.3. Spatial reuse in directional antenna

3.4.1 Spatial reuse factor

Consider a simple scenario as shown in Figure 3.3 where node A wants to communicate with node B and node C with node D. If omni-directional antennas are used, then node C cannot communicate with node D when node A is sending packets to node B. This is because of the fact that node C's packet may interfere with A-B's communication. This is shown in Figure 3.3(a). If the nodes use directional antennas, then the sender will focus the beam towards the receiver. This makes it possible for the transmission between A-B and C-D to go on concurrently as shown in Figure 3.3(b). We can infer from the above example, that if the nodes use directional

antenna then neighboring nodes that are not in the direction of signal can go ahead with their transmissions. Multiple transmissions can now be initiated by different nodes instead of a single transmission in omni mode if they do not interfere with one another and thereby increasing *spatial reuse factor*.

3.4.2 Extended Range and Energy Savings



(a) Omnidirectional mode. (b) Directional mode.

FIG. 3.4. Extended range in directional antenna

Consider the Figure 3.4 in which node A wants to communicate with node C. If omni-directional communication was used then A cannot reach C in single hop. Hence, A has to first transmit the packet to B and B will then transmit the packet to C. Larger directional gain ,given in Equation 3.2, helps node A is able to reach node C in a single hop as shown in Figure 3.4(b). The advantage of higher directional gain can be made used of in two ways. Firstly, because of the fact that focused beam can travel a larger distance than the unfocused omni-directional signal, the sender can now reach a receiver which farther away. This increases the *transmission range*. The greater reception gain helps the nodes to listen to a weaker signal if the signal is arriving at a direction in which the antenna is turned towards. Secondly, the power required to reach a maximum distance d is lesser in directional antenna than

in omni-directional antenna. Hence, by using directional antenna and by regulating the power, transmission and reception cost can be cut down drastically. This *reduces the energy spent by the nodes for transmission and reception.*

These advantages have attracted a significant amount of directional antennas research in Ad hoc environments. However, several challenges are still unsolved; protocol design for directional antennas remains at relatively early stages. These challenges are the topic of the next section.

3.5 Challenges in Directional Antennas

This section highlights the major challenges that needs to be addressed before the potential of directional antennas can be fully realized. This section is organized as two parts, one explaining the problems due to the characteristics of directional antennas and another due to the implementation of DMAC.

3.5.1 Drawbacks in directional antennas

This section describes the problems that are inherent to the nature of the directional antennas. The problems discussed in this section does not appear in omni-directional antenna systems.

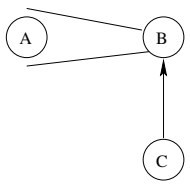


FIG. 3.5. Deafness

Deafness: "Deafness" is one of the problems that occur when directional antenna is used. When an omni-directional antenna is used, all neighbors are capable of

listening to an ongoing transmission. However, when directional antennas are used, there is a possibility that a node is turned in a particular sector while it is receiving. This node is said to have *locked* in a particular sector. When the node is locked in a sector, all the signals that arrive in other sectors cannot be received by the node. The node is said to be **Deaf** in all the other sectors.

Figure 3.5 shows one scenario where the node can be deaf. Node B is communicating with node A and node C wants to send a packet to node B. Node B is "deaf" towards node C since it is turned away from node C and cannot hear to C's transmission. Choudhury et al. [4] present a solution to this problem that utilizes an additional low bandwidth channel. However, without such specialized hardware, the problem remains challenging. In particular, deafness can cause destructive interactions with upper layers: for example, due to deafness, multiple retransmissions may fail causing a node to think that the connection is lost due to mobility and triggering a route discovery search.

Interference caused by higher gain The higher gain of directional antennas results in larger range of the signal. At a given distance, the strength of the focused beam is much higher than the strength of the omni-directional beam. If the signal is able to reach longer, there may adverse effects of interference caused to other ongoing communications in that direction. Details are given in section 5.3.2

3.5.2 Drawbacks Specific to DMAC

Though the physical layer has changed significantly, the above layer does not seem to harness the features of the antenna model. By reusing the same approaches as that of omni-directional MAC, the DMAC creates new problems that were not present in the omni-directional MAC. Most of these drawbacks are explained in the Section 5.3 while trying to analyze the DMAC over chain scenario. Some of the

important drawbacks are mentioned below.

Heightened hidden terminal: Hidden terminal problem arises when a node transmits a signal that may affect an ongoing transmission. This problem has been solved for the 802.11 MAC, but it persists in DMAC. This is explained in more detail in Section 5.3.4

Head of Line blocking: Because of the FIFO queuing mechanism, the node with directional antenna will pick the first packet in its queue to transmit. If the channel is not idle in the direction in which the packet needs to be transmitted, then the node has to wait till the channel becomes idle. There may be other packets in the queue which needs to be transmitted in the direction where channel is not idle. The first packet blocks all the other packets which could be transmitted. This problem is referred as the *Head of line blocking(HoL blocking)*. An analysis of severity of this problem is done in Section 5.3.5. An improvement to the DMAC was proposed to avoid such a problem and is explained in detail in Chapter 7

Imperfect virtual carrier sensing: Virtual carrier sensing was described in Section 3.3.2. Nodes often do not listen to all the signals around them because of deafness. This causes an incomplete DNAV table which does not consistently store the state of the channel in a different directions, leading to imperfect virtual carrier sensing. Section subsec:virtualSensingDNAV the scenario in detail.

Using omni-directional routes: The routing layer uses broadcasts to find the routes. The broadcast, being an omni-directional transmission, does not reach all the directional neighbors. Hence the packet forwarding done by the routing layer will use omni-directional neighbors. This restricts the use of *higher gain* of the directional

antenna. This problem is analyzed in detail in Chapter 8 and new protocol is proposed to passively discover the directional neighbors.

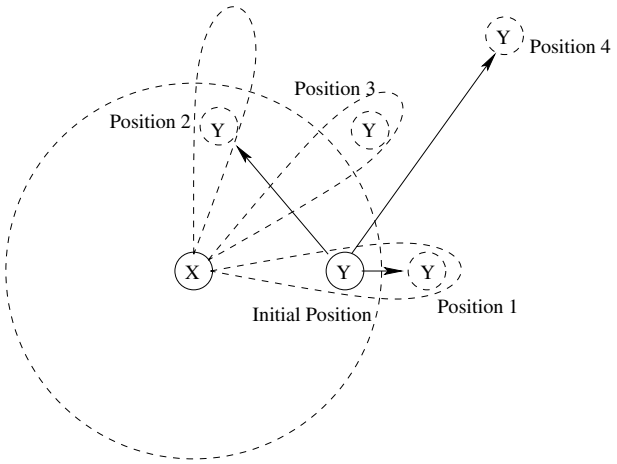


FIG. 3.6. Effect of Mobility

Effect of mobility: Consider the Figure 3.6 where node X is communicating with node Y. To illustrate the effect of mobility, let node Y move to three different positions as indicated in the figure. The omni-directional range of node X is shown with the circle with dotted lines and the directional range of node X in different sectors is shown by the dotted lobes. There are four possible situations under which we can contrast the effect of mobility while using directional antenna with the one using omni-directional antenna.

- *Reachability due to higher range:* In case of an omni-directional antenna as shown in Figure 3.6, if node Y moves to Position 1 then node Y will be out of range of node X and hence cannot receive the packet from node X. Since the directional gain is higher than the omni-directional gain, node Y may be still with in the directional range of node X and hence node Y will be able to receive the packet from node X as shown in Figure 3.6. This shows the case where

advantage of higher range in directional antenna is helping in case of mobility. However, if node Y moves out of the directional range, then the node X will be unable to reach node Y.

- *Reachability in different sector:* Consider the scenario when node Y moves to a new location as indicated by Position 2 in Figure 3.6. In case of omni-directional antenna, node X can still reach node Y as shown in Figure 3.6. While using directional antenna with DMAC, node X will fails to reach node Y using the same sector. Node X fails to communicate with node Y since node Y has gone out of range of signal that is transmitted in that sector. Node X will try to send RTS to node Y for a specified number of times as given by a constant "Directional retransmit limit". If it fails, it will purge the entry to node Y from its AoA cache and will try to send a omni-directional signal to locate node Y. In this case, node Y will receive when an omni-directional RTS is sent and will respond to node X with a CTS. Upon receiving the CTS, node X will update the AoA with the new angle of node Y and start communicating with Y by directional signal. We can observe that node X will now communicate with node Y in a different sector recovering from the short term disconnection. Hence mobility which leads to such a scenario will involve RTS drops to figure out the new sector of node Y.

- *Unreachability due to omni-discovery:* Consider the scenario when node Y moves to position 3 as shown in Figure 3.6. Node X tries to transmit to node Y in its old direction. After failing to reach it after "Directional retransmit limit", node Y tries to send an omni-directional signal. Since node Y is unreachable by directional signal, node X cannot discover node Y and assumes that node Y is unreachable. Even though node Y is reachable by transmitting directional signal in different sector, node X never tries to reach Y using different sector.

In such cases, node X reports an error to the above routing layer and drops the packet due to *no route*. This error can be overcome by searching the node Y directionally in different sectors. Node X, instead of giving up to reach Y, can have a low overhead scanning in sectors adjacent to the original node Y's sector. Scanning of adjacent sectors can be done in a heuristic manner by recording the mobility pattern of node Y.

- *Totally unreachable:* If node Y moves to a new position as shown by Position 4 in Figure 3.6, node X will not be able to contact node Y both omni-directionally and directionally. In such cases, node Y will be out of omni-directional and directional range of node X. Such cases of mobility lead to disconnection which cannot be recovered in the above mentioned ways.

Chapter 4

RELATED WORK IN DIRECTIONAL ANTENNAS

The MAC protocol for Directional Antennas in ad hoc network has received significant interest in recent years. There are many variations of MAC protocols that have been proposed for these antennas. The design of these protocols are influenced by the 802.11 MAC protocol. Many proposed protocols use the RTS-CTS mechanism used in the 802.11. The difference between them can be found in at least one of the following categories:

1. *Directionality of frames:* All the protocols transmit the DATA frame directionally to use the advantages of directional antenna. The initial protocols considered using a mix of omni-directional and directional transmission for RTS and/or CTS.

2. *Directional virtual carrier sensing:* Some protocols had the same virtual sensing mechanism as present in the 802.11. The more advanced protocols accounted for the channel state in each direction and proposed/used a new directional carrier sensing.

3. *Directional range:* Initial protocols assumed that the range of the directional antenna is same as that of the directional antenna.

4. *Number of channels:* Some of the protocols use more than a single channel

The protocols can be further divided into 3 different sections based on the kind of problem being solved:

- Proposing a different MAC
- Mechanisms to overcome drawbacks in MAC
- Routing with directional MAC

One of the initial directional MAC protocol was proposed by Ko et al. [15]. In the protocol, they propose that by sending CTS in omni-directional mode, the ACK collision can be reduced. They advocate the use of omni-RTS when the complete channel is free and use directional-RTS otherwise. While using the directional-RTS in this case, there may be chances of deafness and hidden terminal problem which may result in packet collision. Overall this is a probabilistic model to reduce the collisions. The virtual carrier sensing is not used in the MAC proposed by Ko et al. The DMAC that has been used in this study always uses D-RTS and D-CTS. Hence, it is vulnerable to collisions but because of virtual carrier sensing, this is less likely. The problem of routing layer discovering directional routes has not been addressed in this study. It assumes that each node knows its location and its neighbors' location.

Nasipuri et al. [18] proposes a directional MAC protocol which uses omni-RTS and omni-CTS with a directional DATA and ACK packets. The ability to measure the angle of arrival(AoA) of a packet can be recorded. Hence, after the omni-RTS and CTS, the source and receive will always know the direction of their counterpart. If the packet needs to be sent in omni mode, then the channel should be idle in all sectors around the node. This leads to a scenario which is similar to 802.11 where the packet transmission cannot be initiated even if the channel is idle in the direction of the receiver. If such a scheme was followed for the Head of line blocking described in

Chapter 7, then there would no benefit. This is one of the drawbacks of the protocol. It also assumes that the range of the directional transmission is same as that of the omni-directional transmission. The higher gain of directional antenna is hence not used.

Huang et al. [10] proposed another directional MAC which uses multiple channels. They assume three channels, one for data transfer and two more to send the busy tones. Sender sends the busy tone in sender-channel and the receiver in receiver-channel. The sender senses the receiver-channel before transmitting RTS and receiver senses the sender-channel before responding with CTS. This reduces the hidden terminal problem present. The use of multiple channels not only reduces increases the complexity in deployment but also reduces the bandwidth of the data channel. The DMAC that is used in this thesis is a single channel DMAC. The paper has not dealt with discovering the neighbors. It assumes that the direction to reach the neighbor is known by the node.

The closest directional MAC protocol that was used for the study is the directional MAC protocol suggested by Roy Choudhury et al. in [6]. The virtual carrier sensing is done in a way similar to [24]. The RTS is always sent in directionally. The receiver will receive the RTS in omni-directional mode. The receiver sends the directional CTS to the sender. The reception of the DATA and the ACK packet are directional. This protocol does a good job to identify the transmission and reception modes can be both omni and directional. They also propose a method to shorten the hops by sending multi-hop RTS and send the DATA to hop which can be reachable by directional transmission but not by omni mode. The drawback of this protocol is that they assume that there is a existing neighbor discovery layer which knows the angles in which the beam has to be focused to reach the receiver. The overhead and errors due to routing layer during route discovery will hence be totally eliminated.

The protocol that was used for study does not assume that the sending and receiving directions to neighboring nodes are known. The AoA cache explained in section 3.3.1 maintains the table with the node ID and angle if it has already heard a transmission from that node. If the direction is unknown then omni-RTS will be sent to the intended receiver.

The higher gain of directional antenna was tried to be used by Korakis et al. in their directional MAC protocol [16]. Since the omni transmissions cover lesser range, they proposed to use the "Circular RTS". Instead of sending a single omni-RTS, the protocol suggests to send a directional beam in all the sectors thus covering 360° around the node. The drawback of such a protocol is the wait time to transmit omni beam. If a omni beam needs to be sent then the channel should be idle in all directions around the node. This eliminates the hidden terminal problem to some extent but will suppress the channel reuse factor. The protocol does not study the omni-directional transmission of broadcast packets. Hence, it does not resolve the issue of finding routes even though it proposes the method to know the direction of the neighbor after listening to its transmission.

The virtual carrier sensing in directional antenna that was used by the study was introduced by Takai et al. in [24]. They cache the Angle of arrival of the signal into the AoA cache as explained in Section 3.3.2. Based on the state of the DNAV, transmissions are scheduled. If the sector in which the beam is to be transmitted is busy, then it will be marked in the DNAV. This information is made use before initiating the conversation. However the signal may be listened by side lobes too as explained later in Section 7.3.2. This does not address the effect of side lobes while the antenna is locked in some other direction which may lead to incorrect updates of the AoA cache.

One of the main problems of directional antennas is the "Deafness" problem described in Section 3.5.1. To solve this problem Roy Choudhury et al. describe a novel method in [4]. Sinusoids are sent in a separate channel after the data transmission is over. If the sender or receiver was deaf to any RTS that could have arrived during that time, then the sender of the RTS will update its state after hearing to the sinusoid and will realize that the other node was deaf. It may attempt to re-contact the node after listening to the sinusoid. However, the use of multiple channel to solve deafness is the deployment barrier. The DMAC studied in this thesis does not employ this method to detect deafness.

The routing layer using omni-directional routes cannot reach directional neighbors barring them to used while constructing the routes. Hence the routes discovered do not use the higher range of the directional antenna. Some protocols has been proposed to overcome this deficiency. Roy Choudhury et al. attempt to make use of directional range by "sweeping" the beam across all sectors instead of sending a single omni-directional beam for route request broadcast packets. The idea is similar to the "Circular RTS" proposed by Korakis in [16]. By doing so, the routing layer will have the knowledge of the directional neighbors too. Hence the directional routes can be found reducing the hop count of the path. Sweeping in all the sectors instead of single broadcast is more expensive. They propose certain schemes to optimize the sweeps but it is still costlier than the single broadcast. The directional routing described in this thesis 8 tries to reduce the number of hops by interaction with MAC layer instead of sweeping. Our study does tries to eliminate the overhead by reducing the number of hops based on the AoA cache which stores the directional and omni-neighbors present in the MAC layer.

Asis et al. [1] proposes to optimize the route re-construction by restricting flooding in a particular direction instead of re-flooding in all directions after a route is

broken.

The HoL problem which is solved in this thesis is present in all the MAC protocols described above because of the FIFO queuing policy. This thesis analyzes and proposes a solution to the HoL blocking caused in directional antenna.

Chapter 5

ANALYSIS OF DMAC

5.1 Overview

Omni-directional 802.11 MAC has proved to be ineffective for multi-hop connections because of unintentional and destructive interactions with upper protocols. There is motivation in the Ad hoc community to come up with a more powerful transmission strategy for using the channel more effectively. "Directional Antennas" provide some of the solutions to the shortcomings of omni-directional antennas.

Our attempt in this study is to characterize the behavior of directional antenna MAC protocols and their interactions with upper layer protocols. More specifically, as was discussed in Chapter 3, directional antennas pose several unique challenges such as deafness, different forms of hidden and exposed terminal problems and gain mismatch. In the presence of the unique challenges posed by directional antennas we believe that such analysis is needed to identify the most important challenges to solve and to identify interactions with upper layers. Furthermore, directional MAC protocols employ techniques similar to those developed in the omni-directional context. Thus, we seek to understand whether such techniques are effective or whether new solutions are needed for directional medium access.

The complexity of the interactions that occur in a directional antenna setting have made the analysis task quite formidable. We carry out the analysis using a

Parameter	Value
Number of nodes	8
Omni-directional range	250m
Simulation time	50sec
Mobility	none
Propagation Channel Frequency	$9.14 * 10^8$ Hz
Path loss Model	Two Ray
Transmission power	24.5 dBm
Receiver sensitivity	-68.1 dBm
Directional gain	10.0 dB
Antenna Model	Switched Beam
Directional NAV Delta Angle	22.5 degrees
Routing Protocol	AODV
Transport Protocol	TCP Reno

Table 5.1. Simulation Parameters

simple chain topology to make identifying problems easier. Further, while we have learned several important lessons, we have not been able to explain all of the observed behavior; in some cases our understanding remains limited and further analysis work needs to be carried out.

5.2 Simulation details

The simulations are run on Qualnet simulator [20] with DMAC support. The analyzed topology consists of a chain of 8 nodes placed at different angles. Each node is 200 m apart from its adjacent node. The omni-directional range is adjusted to 250m. Table 5.1 gives some of the relevant simulation parameters. Most of the simulations involve a single TCP connection across 4 hops. The the behavior of the network is studied with emphasis on the MAC layer transmissions.

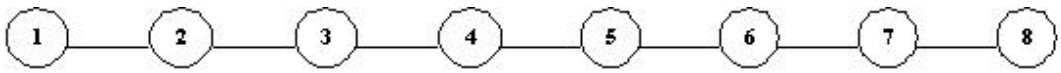


FIG. 5.1. Chain Topology

5.3 Analysis of a chain topology

Consider the 8 nodes arranged in a chain topology as shown in the Figure 5.1. There is a single TCP connection from node 2 to node 6. Similar analysis has been done for 802.11 by Xu et al. in [26] and a few surprising results were brought up. This study tries to analyze the DMAC protocol in simple chain scenarios.

For a simple chain scenario as described above, there is around 41% increase in throughput vs. omni-directional MAC as shown in Table 5.2. We also observe that the number of packets dropped due to route failures (No RouTe Error or NRTE) are higher in omni-directional than in directional.

NRTEs are generated when a packet transmission fails for a number of consecutive retries. Wireless MAC protocols typically use the consecutive failures as an indicator that a node has moved and is no longer reachable. However, Xu and Saadawi show that persistent interference can cause NRTEs to occur [26].

Overall these results indicate that the directional MAC is able to achieve higher reuse than its omni-directional counterpart. As delve deeper, it was found that the basic handshake in DMAC is not serving the purpose and the advantages are not being used effectively. In the following sections we try to analyze the effect a number of factors like handshake, nature of packet drops, deafness and queuing policy. Overall, we conclude that despite the many inefficiencies that remain to be solved in DMAC and its interactions with higher layers, already it is more effective than omni-directional MAC. Moreover, as these issues start to be addressed, the potential exists for significantly higher capacity and re usability using directional antennas.

Description	Omni	Directional
Number of NRTEs	26	14
Number of RTS drops	3626	5493
Throughput	221065 bps	312354 bps

Table 5.2. Comparison of Directional and omni-directional MAC protocols

5.3.1 Is the RTS-CTS handshake effective in DMAC?

The RTS-CTS handshake was devised to reduce the effect of the hidden terminal and exposed terminal problems in omni-directional antennas. With the directional propagation, neither directional or omni-directional RTS-CTS block all possibly interfering nodes: omni-directional RTS/CTS do not reach interfering directional neighbors. Moreover, directional RTS/CTS only block interferes in the sector covered by the RTS/CTS. Observing the above throughputs and NRTEs, the DMAC seems to be working better than the omni-directional. However, surprisingly, it can be seen that the basic handshaking protocol hardly serves its purpose in directional mode. For the regulated single TCP connection, DMAC results are positive. But when the same chain topology is subjected to high rate CBR connection, the DMAC should perform on-par or better than the omni-directional MAC. A chain topology depicted in Figure 5.1 is set up and a CBR connection running from node 2 to node 6 was simulated. Not only the throughput was lesser than the omni-directional MAC, but also the number of NRTE's in directional was very high as depicted in Table 5.3.

In directional antennas, we conjecture that additional NRTEs occur due to deafness in two ways: (1) deafness causes RTS sends to a deaf destination to fail; and (2) deafness causes interfering nodes to possibly miss RTS/CTS packets. Under different geometries, its also possibly that directional RTS/CTS transmissions are not sufficient to block interferers, but this would not be the case in a straight-line chain.

In case of omni-directional communication, the RTS and CTS is designed with

Description	Omni	Directional
Number of NRTEs	35	84
Number of RTS drops	4578	4633
Throughput	344691 bps	309457 bps

Table 5.3. Comparison of Directional and omni-directional MAC protocols for a CBR connection from 2-6

two objectives:

1. To let the receiver know that the sender wants to transmit the data.
2. To let the sender know that the receiver is free to receive the data.
3. To notify the neighbors to remain silent for the time the data is being sent. This is accomplished by the nodes setting the NAV table when it listens to the RTS.

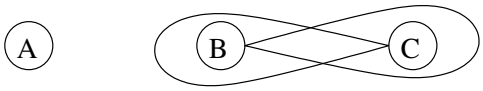


FIG. 5.2. A case where the handshake is not functional

Consider the scenario of RTS-CTS handshake in DMAC as described in Figure 5.2. If node B wants to send a packet to node C, then B will send an RTS in the direction of node C. Assuming that node A is directionally out of range of the CTS that is sent by B, it will be unaware of this transmission going on. This assumption is fair because node A will stay in omni-directional mode for receiving packets when it is not engaged in any conversation. The range of the signal for a *Directional Send - Omni Receive* is not as high as *Directional Send - Directional Receive*. Hence, there is node A will not listen to the node C. When the communication between node B-C is going on, node C would have turned its direction towards node B. Any signal that

is coming in this direction will be received with higher directional gain by node C. If node A wants to send a packet to node B and sends an RTS in this direction, there is a possibility that it will interfere with the data being sent from node B to node C because of the direction of node A's RTS packet. A proper handshake should prevent node A from sending the RTS. This simple example shows the ineffectiveness of RTS-CTS in DMAC. It serves only two of the three RTS-CTS functionalities. Informing the neighbors about the silence duration is an important factor to avoid random backoffs which will be explained in detail in Section 5.3.4. It is to be observed that deafness is also seen in this case. Node B will be deaf towards node A. The point to be emphasized is that the handshake is not effective. Even if deafness is ignored, the RTS/CTS does not function as effectively as it is in 802.11. Deafness will be explained in more detail in the following sections.

5.3.2 Effect of Higher Gain

In this section we study if the higher gain present in the directional antennas are really used effectively and the ill-effects of higher gain.

Interference in DMAC Consider node 2 transmitting a packet to node 3 in Figure 5.1. The range of the transmission is much higher than that of the omni-directional MAC because of the higher directional gain. Though node 4 is not in the omni-directional range of node 2, it is in the directional range of node 2 when both the sender and receiver are locked towards each other. Hence, the interference effect is more pronounced at node 4 while using the directional MAC. In the chain topology described above, interference because of directional gain should lead to higher interference for the nodes that are reachable in directional gain.

Directional routes The node uses omni-directional transmission if:

- The node does not know the direction of the intended next hop
- The packet needs to be broadcasted.

All the routing algorithms use broadcast to search their neighbors. Because of this, the initial routes found out are omni in nature and hence does not make gain of the directional reachability. Hence, if a node is at a distance that is directionally reachable and out of omni-directional range then it will not be considered as the next hop.

This effect is solved by using altered MAC protocols by Choudhury et al in [6]. In [6], Choudhury et. al. try to reduce the number of hops. This helps to achieve higher throughput. So, directional range is not used in normal directional MAC but its being penalized because of the range because of the interference range of directional MAC is much higher than the omni-directional MAC.

5.3.3 Scenarios for RTS packet drops

It can be observed in Table 5.2 that the number of RTS drops in DMAC is around 50% more than that of omni-directional 802.11 MAC. This makes the study of the RTS drops an important part of the DMAC analysis. This section tries to give the typical scenarios in DMAC which may lead to packet losses. Primarily, the packet losses can occur due to the 3 kinds of scenarios in DMAC:

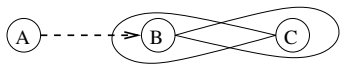


FIG. 5.3. Standard Deafness

Scenario 1: Standard Deafness: Consider the Figure 5.3 in which node B is communicating with node C. If node A tries to send an RTS to node B, the RTS

packet will be dropped because of node B being turned away from node A. Node B will wait till for a timeout period and will backoff after the timeout assuming that the channel is congested. Let us call such a deafness as *Standard Deafness*.

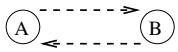


FIG. 5.4. Simultaneous RTS

Scenario 2: Simultaneous RTS: Figure 5.4 shows a case where node A and node B try to contact each concurrently (within propagation delay time). Such scenarios will lead to RTS packet drops at both the nodes. Although the occurrence of this scenario seems unlikely, it was observed in the simulations.

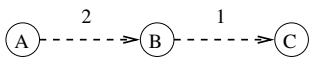


FIG. 5.5. Back-to-back RTS

Scenario 3: Back-to-back RTS: In the Figure 5.5, node B is trying to reach node C and node A is trying to reach node B at approximately the same time. If node B first sends an RTS towards node C and node A then transmits an RTS to node B, the RTS of node A will collide with the reception of RTS at node C because of the interference caused due to higher gain as explained in Section 5.3.2. This leads to the drop of RTS packet sent by node B to node C. Since node B was turned towards node C, it will be deaf towards node A. Hence the RTS packet sent by node A is also not received by node B. This scenario leads to two RTS drops. The precondition for such a scenario to occur is node B starting the RTS transmission to node C first and node A starting its RTS transmission to node B before the complete RTS packet has been received at node C.

The above 3 scenarios can be considered as the basic scenarios under which RTS drops may occur. There were several instances in the simulation where a combination of such scenarios occurred resulting in RTS packet drops. Few of the interesting ones are explained below:

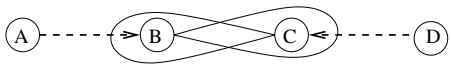


FIG. 5.6. Two-way standard deafness

Two-way standard deafness: In such a cases standard deafness happens at both the ends involved in communication. Figure 5.6 shows node B communicating with node C. Node A tries to reach node B and node D tries to reach node C. Two RTS losses will occur in this scenario.

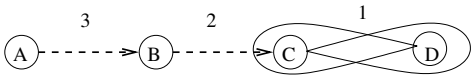


FIG. 5.7. Standard deafness followed by back to back RTS

Standard deafness followed by back to back RTS: Figure 5.7 shows a scenario where packets are lost because of standard deafness and back to back RTS. Node C is involved in transaction with node D when node B tries to reach node C. This is the *Standard deafness*. Node A tries to contact node B at this time losing the RTS packet. Even though the RTS drop at node C does not happen because of collision as in case of *Back-to-back RTS*, the timing of RTS packets from node A and node B follows the rule stated in *Back-to-back RTS*. This scenario leads to 2 RTS losses.

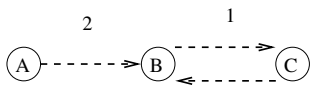


FIG. 5.8. Standard deafness over Simultaneous RTS

Standard deafness over Simultaneous RTS: Even though very rare, this kind of RTS loss was also observed. This kind of scenario leads to 3 RTS losses. Figure 5.8 shows such a scenario. Node B and C will lose 2 RTS by *Simultaneous RTS*. Node A also transmits an RTS to contact node B which is deaf towards node A because it is busy in sending RTS to node C. Hence, node A loses its RTS packet too.

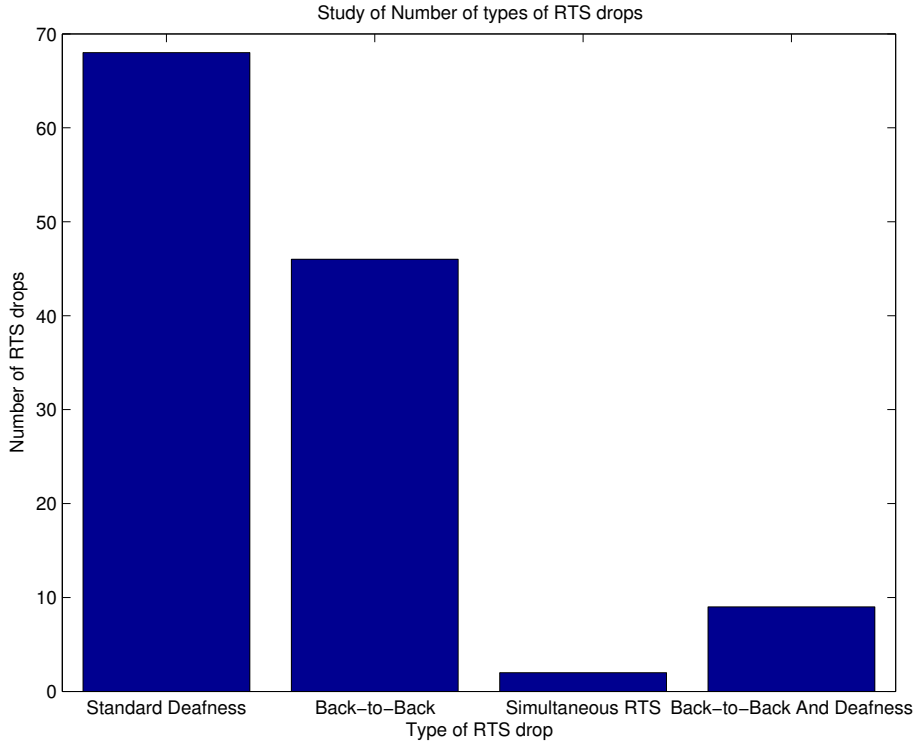


FIG. 5.9. The number of RTS drops of each kind

The graph 5.9 shows the number of RTS drops in the above kind. The graph is not constructed by considering all the RTS drops. Around 125 RTS drops were

scanned to find out the cause and the graph depicts the number of RTS drops under each case. We can see that Standard Deafness and Back-to-Back RTS are the main causes of the packet drops.

5.3.4 Consequences of Deafness

One of the critical things to be resolved while using directional antennas is deafness. This happens when a node is turned away from the direction in which the signal is arriving. Choudhury et al. [4] tries to solve the problem of deafness by transmitting sinusoidal waves in a separate channel. The current DMAC does not incorporate the solution proposed by Choudhury et al. [4]. The need for a separate channel for control sinusoids makes the solution harder than a single channel DMAC for deployment.

In case of the chain topology with TCP connection described above, deafness should be a dominant effect because of the nature of packet forwarding. Consider the chain in Figure 5.1 where node 2 originates the packets destined for node 6. Node 2 will pass the packet to node 3 which in turn turns towards node 4 for forwarding the packet. If node 2 has more packets to send to node 3, then it will send the RTS to node 3. There is a possibility that node 3 is now deaf towards node 2 because it is transmitting the packet to node 4. Hence, each node will be deaf towards its predecessor node when it is forwarding the packet to its next hop.

Deafness results in significant problems in DMAC. The following subsections discuss some of the important consequences of deafness:

RTS drops: The Section 5.3.3 shows the possible scenarios under which the RTS packets may be timed out. By observing the main scenarios, it can be found that two of the three scenarios are because of deafness.

Hidden terminal: Hidden terminal problem is solved in omni-directional system but it still persists in DMAC. A part of the problem is explained in section 5.3.1. It was seen that the RTS and CTS packets will not reach all neighboring nodes, making them unaware of the ongoing transmission. Figure 5.3 shows the hidden terminal problem in DMAC. Node A being unaware of the transmission between B and C will try to send an RTS to node B. Since node B is deaf towards node A, A fails to get a CTS. This is a typical hidden terminal problem that is unsolved in the case of DMAC.

Virtual carrier sensing: Consider the Figure 7.2 where node W is communicating with node X and node Y with node Z. Let nodes X and Y be within transmission range of node W. When the node W is communicating with node X, node Y should ideally mark its DNAV appropriately indicating the wait time in the direction of node W. If node Y is busy communicating with node Z, then node Y will be "deaf" to node W. This inhibits the accurate DNAV update at node Y. The state of DNAV does not reflect the channel state in the direction of node W. This shows that the deafness prevents the DNAV updates.

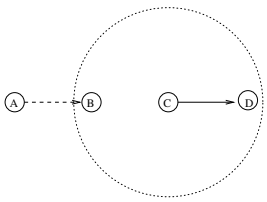


FIG. 5.10. Exposed Terminal in Omni-directional MAC

Deafness causing Backoffs and NRTEs: Consider a scenario in the chain topology on TCP connection from node 2 to 6 as shown in Figure 5.1. We observed in table 5.2 the RTS drops in DMAC is around 50% more than the omni-directional MAC, but the NRTE's are much lower than that of omni-directional MAC.

In case of omni-directional transmission, NRTE's will be found because of the "Exposed terminal problem". Consider the nodes A,B,C and D arranged in a chain in Figure 5.10. When node C is sending a packet to node D, node B will not initiate any conversations. Node A being unaware of the data transmission between C and D, is free to send an RTS to node B. Even though node B gets the RTS of node A, it does not reply with a CTS because the transmission from node B may interfere with the communication going on between node C and node D. Node A is the exposed terminal that is unaware of ongoing transmission. This problem is called as *Exposed Terminal problem*. Since node A does not get a CTS back, it assumes that the network is congested and will backoff and transmit the RTS packet after backoff interval. *This backoff truly reflects the state of the channel*. Since the node C-D's communication can go on for a longer time, node A will usually experience 3 to 4 backoffs before C-D has finished the transmission. The number of backoffs depend on the amount of time remaining between C-D's conversation when node A tries to send the RTS. Node A would have backed off for a longer time due to the exponential nature of the back off. It is generally observed that by the time node A tries with another RTS, node C would start transmitting another packet to node D if there is a queue of data packets in node C which needs to be transmitted to node D. So, when node A tries again after the long backoff period, it is possible that it does not get any CTS back from node B. After *RTS Threshold* attempts (usually, 7 attempts), node A will drop the packet because of NRTE. The advantage in omni-directional mode is the correct virtual carrier sensing. Since node B knows the state of communication between nodes C and D, it will try to transmit a packet after the communication and does not backoff. This silence period is exactly marked in its NAV table. Hence, if node B has packets to send to node C, it will do so without backing off. When node B sends the RTS to node C, node A will also listen to the node B's RTS. This makes

node A to **freeze** the backoff counter for the duration node B is transmitting to node C.

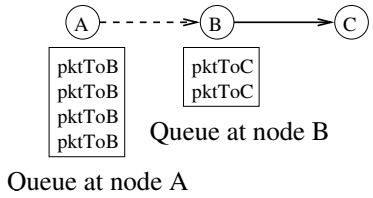


FIG. 5.11. Backoffs getting triggered in DMAC

But in case of directional transmission, NRTE's is mainly caused because of the deafness causing *Hidden terminal problem* and improper updates of the DNAV as explained in sections 5.3.4 and 5.3.4. Figure 5.11 shows 3 nodes arranged in chain topology running a part of TCP connection with TCP data flow the direction A to C. Let us assume that the node B has enough packets in its queue to send it to node C. The validity of this assumption is later explained in Section 5.3.6. Node A being nearer to the source will have more packets to send to node B. After a packet is sent from node A to node B, either node A will send another packet to node B or node B will send a packet from its queue to node C. If node B wins the channel and sends an RTS towards node C, then node B will be deaf towards node A. Node A being unaware of B-C communication, will initiate an RTS for node B. This will result in a RTS drop because of *Standard deafness*. Hence, node A backs off and will try to send an RTS after the backoff time. Since the time required to send a TCP data packet is much larger than the initial backoff time, node A will try to send the RTS again to B when B is still communicating with C. In a similar fashion, A will experience around 4 backoffs before the communication between B and C. This results in an increased backoff timer because of exponential backing off. Hence, when A will send an RTS again after relatively longer time. After B is finished its communication with node C, it will wait for DIFS period and will send an RTS to node C again to send the packet.

While this communication is going on, A's backoff period will end and A will again try to send a packet to B. B being deaf again, A is going to backoff further. Node A is going to find B in a free state to communicate will be only when one of the below given conditions is satisfied:

1. Node A tries to send RTS during the DIFS wait period between the transmissions of node B.

2. Node B tries to communicate with node C, but node C is busy and hence node B has backed off and is free to communicate with A.

3. Node B does not have any other packets to send.

Condition 1 is rare because of the exponential backoff that node A experiences and the small size of the DIFS period. The backoff timer of node A has to end at a time such that after node A waits for DIFS period and tries to send an RTS then node B is waiting in its DIFS period. Condition 2 can happen often if node C also has enough packets to send to its next hop. Here node B's RTS to node C will time out because of one of the reasons given in Section 5.3.3. This analysis will be similar to the analysis done here.

Condition 3 is seen to occur more frequently than the other two points. Once node B has exhausted all its packets, it will be free and node A can send the RTS to node B. If there are many packets at node B, then there is chance that node A will experience larger RTS timeouts resulting in NRTEs. The effect of regulating the queue of packets down the chain is explained in section 5.3.6

Route error messages: Consider the case where node 3 experiences an NRTE while trying to send a packet to node 4 in the Figure 5.1. Many of the routing layers like AODV and DSR will first invoke a *Route repair* mechanism to fix the route locally

at node 3. If this repair fails, then the source is reported of the broken route. The source will initiate the *Route Discovery* to the destination again. After node 3 reports the broken route to node 2, node 2 will start the route discovery process.

The route repair and route discovery phases operate by sending broadcast packets and no data packets can be further sent before the route is built. This leads to large wait time for the packets that are generated by the source and is waiting in the queue of the nodes. As we see in section like 5.3.6, this may result in many retransmissions in the source too. Hence, experiencing an NRTE when the actual route is still available is a costly affair which degrades the throughput of the channel by longer idle times.

5.3.5 Queuing in DMAC

Queuing policy at the MAC layers was explained in the Section 7.2. In this section, we try to describe the problems in DMAC which are related to queuing policy. As explained in the Section 5.3.4, the queue length at each node plays a critical factor which controls the throughput of the network. The subsection 5.3.5 explains the effect of the queue length and the subsection 5.3.5 describes another important effect of queuing policy in DMAC.

Effect of queue length: For the channel to be used efficiently, there should be as many parallel transmissions as possible. To enable this, the nodes should have sufficient packets to transmit to their next hops. In the chain topology described above, where there is a single source, which keeps injecting the packet into the network, the source is responsible for loading the queue of the subsequent nodes. If a greedy attempt is followed by the source to as many packets as possible, then the nodes will have enough packets for parallel transmission which should increase the throughput of the channel. A very large queue length in the nodes will result in the degradation of throughput The number of packets in the queue which are waiting to

be sent to the next hop determines the throughput of the connection.

Consider the chain scenario as shown in Figure 5.1 with a TCP connection running from node 2 to node 6. If node 3 has a large number of packets in its queue, then it will try to transmit them to its next hop node 4. If node 2 wants to send the packet to node 3, then the chances that node 3 is free are lesser. This creates RTS drops because of deafness at node 2 which may lead to NRTEs resulting in large delay times in the network. This causes the throughput to decrease.

A source with TCP with window size 32 packets can push more packets to the network than a TCP with window size 2 packets. The queue lengths of the nodes will be higher in TCP with window size 32 packets. Section 5.3.6 analyzes the effect of queue length in more detail by studying the TCP with window size 32 and 2 packets.

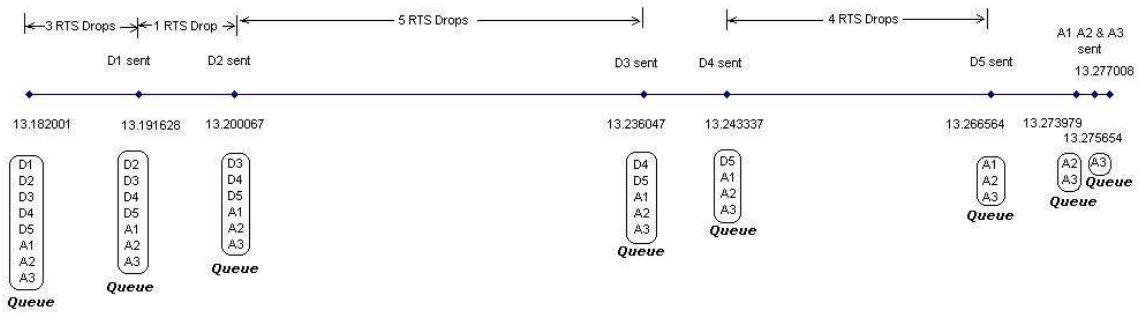


FIG. 5.12. Head of line blocking at node 3 in TCP traffic

Head of line blocking: Consider the TCP-ACK traffic in the Figure 5.1. The TCP-ACKs is sent from node 6 to node 2. In the simulations it was found that the TCP-ACK packets which are blocked by the DATA packets get delayed even if the channel towards which the TCP-ACK packet needs to be sent is free. Lets assume node 3 has data packets to be sent to node 4 and TCP-ACK packets to be sent to node 2. If the data packets are ahead in the queue than the TCP-ACK packets and

if node 4 is busy with node 5, then the TCP-ACK packets should wait till the node 5 is free and the data packets are sent to node 4 even though node 2 is free and the transmission of TCP-ACK packet does not interfere with node 4. This *Head of line blocking* problem was briefly explained in introduction. Consider the Figure 5.1.

To visualize the importance of the problem consider the state of node 3 from time 13.182 sec to 13.275 sec. The time line in Figure 5.12 shows the events that occur at node 3 during this time period. The data packets are named as D1,D2,D3,D4 and the A1,A2,A3. The queue of node 3 is shown at the bottom of the figure. We can see that the 3 TCP-ACK packets are blocked by 5 DATA packets. Throughout this time line, node 4 is also trying to transmit its packet to node 5. Hence, *standard deafness* will be experienced by node 3 when it tries to transmit to node 4. There will be RTS drops and backoff by node 3 when it tries to send RTS to node 4 if node 4 is communicating with node 5. Node 2 is free most of the times to receive the TCP-ACK packets during this time period. We can see that sending TCP-ACK packets to node 2 and sending data packets to node 4 happens in different sectors of node 3. Hence instead of waiting to send data packet when node 4 is busy, node 3 can send TCP-ACK packet back to node 2. This gives better channel utilization.

Consider the waiting time for the TCP-ACK packets. Node 3 was able to transmit all the three TCP-ACK packets from time 13.273979 sec to 13.277008 sec. This is because node 2 is free and there will not be any RTS losses while transmitting these TCP-ACKs. It had to wait from time 13.182001 sec to 13.273979 sec. This long wait period is because of the 13 RTS drops that are experienced while sending the data packet to node 4 since node 4 was busy. Hence the policy to select the first packet from the queue is not efficient in the DMAC. DMAC can have a better queuing policy to pick the right packet from the queue instead of selecting packets in FIFO order. The chapter 7 gives a detailed analysis of this problem and proposes a solution for

such a problem.

5.3.6 TCP Analysis

TCP is sensitive to the packet drops in the network. It adjusts is sending rate based on the the state of the receiver and its interpretation of possible congestion in the network. It is well observed that TCP does not work perfectly over 802.11. In this section, we try to analyze the the working of a simple TCP chain over DMAC.

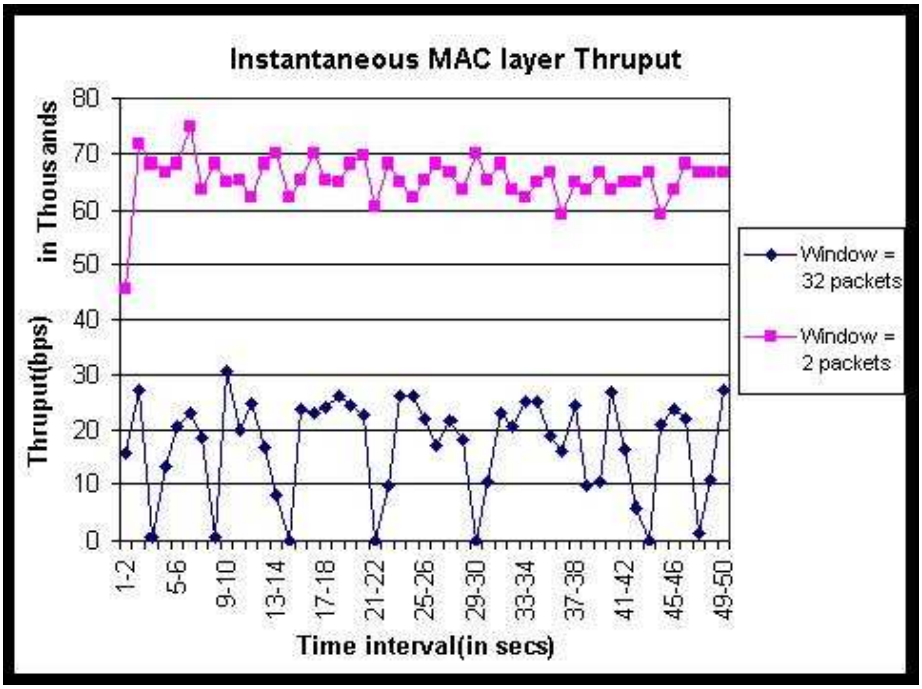


FIG. 5.13. Instantaneous Throughput

Source dictation: In the single connection chain topology explained above, there is only one source. The flow of packets between the nodes and their effect on throughput is primarily based on the source controlling the packet injection into the network. The throughput observed at other hops is a reflection of the throughput available at the first hop.

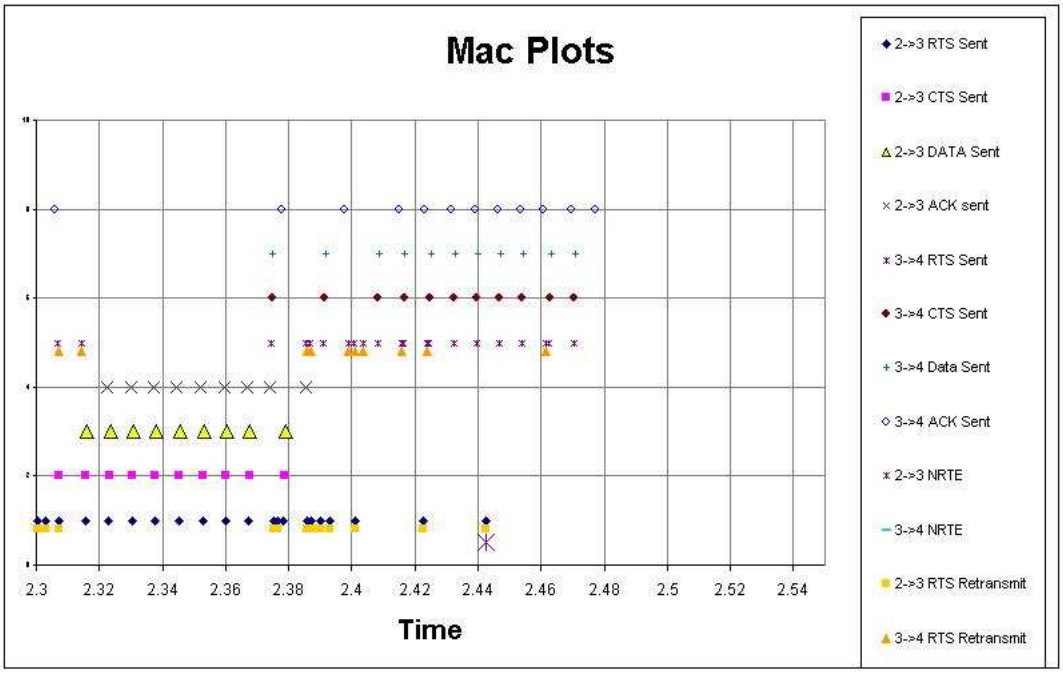


FIG. 5.14. Source pushing more packets

Description	Win 32	Win 2
Number of NRTEs	14	0
Number of RTS drops	5493	924
Number of DATA drops	26	4
Number of ACK drops	6	2
Throughput	312354 bps	496133 bps

Table 5.4. Comparison of TCP Window Size 32 v/s Window Size 2

If source is well behaved, then the packet flow in the network smooth resulting in higher throughput. If source tries to pump in more packets greedily, then the instantaneous throughput drops to zero. From this we can infer that the queue length at 3 is a major indication of how the throughput in next few seconds is going to be.

When the TCP window size is set to 32, the source has more room to push the packets into the network and hence causes the queue length at nodes to increase and hence a choppy instantaneous throughput as shown in the Figure 5.13. This can be

clearly seen in Figure 5.14. The graph is plotted as follows. The value of 1 on Y axis indicates an RTS Send from node 2 to node 3. Similarly, CTS, DATA send and ACK sent from node 2 to node 3 are represented from value of 2 through 4. Values 5 to 8 correspond to RTS,CTS, DATA and ACK sent from node 3 to node 4. The RTS timed out from node 2 to node 3 are shown right below the 2-3 RTS send. The NRTEs between node 2 and node 3 are shown as asterisks below 2-3 RTS send.

We see that node 2 is pumping packets to node 3 and node 3 does not get enough chance to push the packets to the node 4. This can be seen from time 2.31 to 2.38 seconds. Node 2 keeps winning the channel and keeps sending packets to node 3. Node 3 does not have time to send those packets to node 4 because it keeps receiving RTS from node 2 before it tries to contend for the channel to send the packets to node 4. Finally, node 3 wins the channel at 2.38 seconds. When this happens, node 3 will be locked towards node 4 for sending the packets. Hence it will be *deaf* towards the node 2's RTS packets. We can see that are lot of RTS retransmits from 2 to 3 during the time 2.39 to 2.44 seconds. This makes the node 2 to back off and finally cause an NRTE at 2.44 seconds. The packet drop will be considered as the network congestion by the TCP, and node 2 will be idle till the time 2.82 seconds. There will be no more packets generated by node 2 to send across the network. Hence the whole network stays idle till 2.82 seconds. There is another NRTE generated at 3.27 seconds. After the second NRTE, the channel will go completely idle from 3.27 seconds to 4.55 seconds. This example shows that the pumping of packets by source leads to channel idle times leading to lesser throughput.

Now consider a well behaved source. We can simulate this by setting the TCP window size to 2 so that the source cannot transmit more packets into the network. The overall application level throughput is 496k bits per second where as in case of the window size 32 the throughput was 319k bits per second. This is a 55% improvement

in the throughput. There are no NRTEs present in this case and the number of RTS retransmits are also low as shown in 5.4. The instantaneous throughput never drops down to zero. The graph for instantaneous throughput is very smooth when compared with that of the window size 32 as shown in Figure 5.13. The goodput also stays high. This effect is also observed in omni-directional transmission as cited by Xu et al. [26].

TCP delays: This section will try to identify the segments of the chain where the delays may happen. It tries to point out the bottleneck links in the chain topology. The same effect can be observed for UDP traffic too.

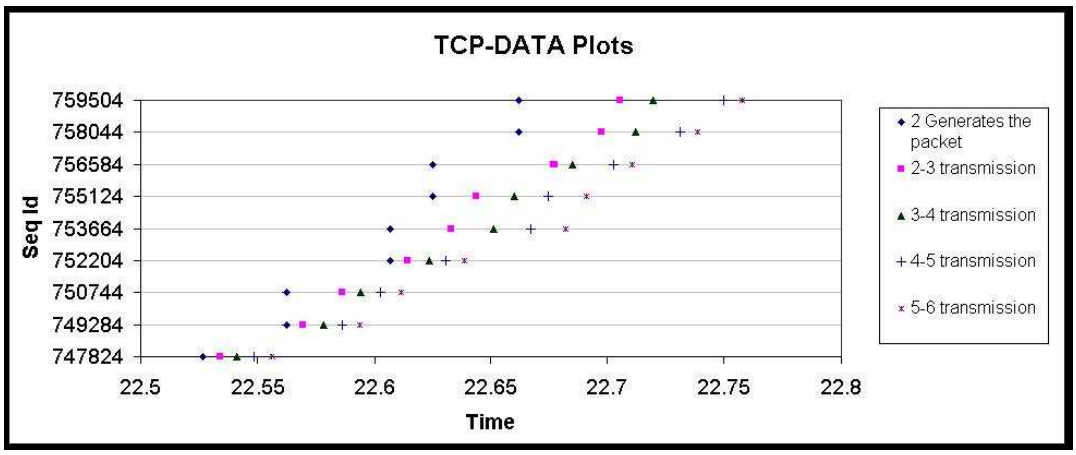


FIG. 5.15. TCP Send: Delays building up

The Figure 5.15 shows the typical scenario in which the packets are forwarded through the nodes. The points describe the time at which the packet was generated at the source and the time at which the packets are transmitted through the chain. This graph is taken for a TCP of window size 32 packets with each packet 1460 bytes and the connection running from node 2 to node 6. The graph plots the points when the node 2 starts producing packets after a long channel idle time. Hence, there is not much traffic when first packet with sequence number 747824 is sent. First two

packets make it to the destination without any delay in any of the hops. As the packet queue builds up at node 2, we can see a larger delay near the source to push the packets. We had seen the effect of queue length earlier in section 5.3.5. In this section, the delays are seen from the TCP perspective from the time the packet has been generated at the transport layer. As the sequence number increases, the trend of larger wait time at source node can be seen. This is because node 3 already has packets to send to node 4 while node 2 is trying to send a packet to node 3.

Consider the packets with sequence numbers 752204 and 753664 (renamed A and B, respectively). Both packets are generated at the same time in the transport layer. Node 2 successfully transmits packet B to node 3 at the time when packet A is being transmitted from 4 to 5; concurrent transmissions can proceed. Node 2 could not have have transmitted the packet B if it was using omni-directional antenna because the CTS from node 3 would have disturbed the 4-5 transmission of packet A. *Thus, a primary advantage of directional antennas, even when using only omni-directional neighbors is higher reuse ratio.*

The amount of time taken by packet B to reach node 6 is more than twice the time taken by packet A. This indicates that the connection is not able to take advantage of concurrent transmissions between the nodes. In a perfectly pipelined scenario, packet B should be transmitted from 3 to 4 when packet A is going from 5 to 6. However, this was not observed; from MAC traces we noticed that node 3 tries to send the packet to node 4 when node 4 is transmitting to node 5.

We can see that as the the sequence IDs increases the delays increase. A stage is reached when congestion causes RTS drops because of deafness and lead to NRTE in the TCP-DATA. This NRTE not only causes route errors as described Section 5.3.4 but also causes a dropped TCP-DATA packet which eventually leads to a *hole* in the TCP window. This hole will lead to sending of duplicate ACKs.

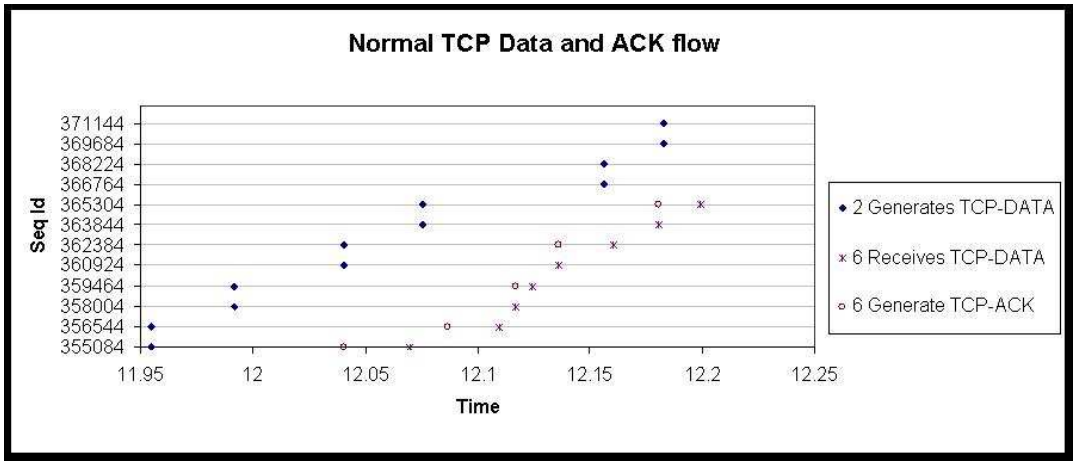


FIG. 5.16. TCP Sending Cumulative ACKs

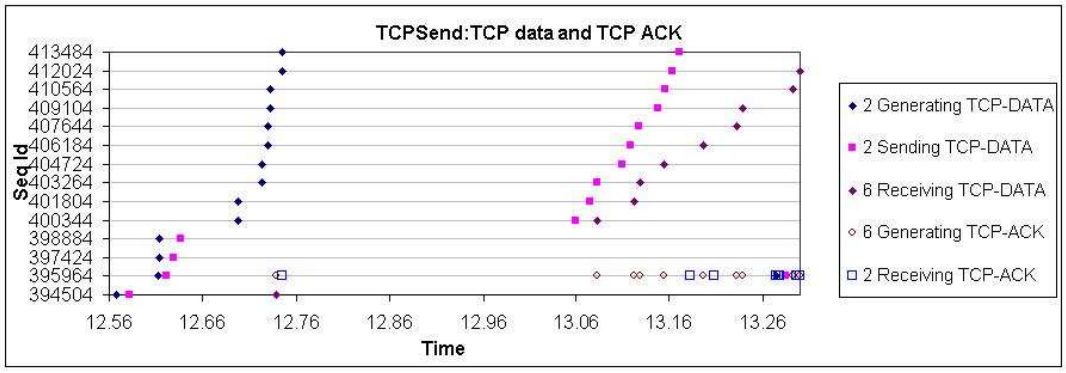


FIG. 5.17. TCP Sending Duplicate ACKs

Usually, TCP sends *delayed ACKs*; it does not ACK every packet it receives. Instead it waits for certain time and sends a cumulative ACK as shown in Figure 5.16. It can be seen that the node 6 is generating a TCP-ACK for every 2 data packets. When there is a hole in the window, a duplicate ACK will be sent for every packet it receives. This doubles TCP-ACK traffic in turn increases the delays caused because of increased network load. This is aggravated by *Head of line blocking* causing longer delays for both TCP-DATA and TCP-ACKS. Consider Figure 5.17. It can be seen that node 6 generates an ACK for sequence ID 395964 as soon as it gets the TCP-DATA packet with sequence ID 394504 to indicate that the next sequence ID

it expects is 395964. This expected TCP-DATA packet which was already in flight and was dropped at link 3-4 because of an NRTE which resulted in route errors (Not shown in the figure for clarity).

The effect can be seen in the graph by observing that node 6 never receives the packet with sequence IDs 395964, 397424 and 398884. The next two packet with sequence IDs 397424 and 398884 are also dropped in the network because of no route. After the route is again formed, node 6 receives packets whose sequence IDs are greater and equal to 400344. For each TCP-DATA packet received, node 6 sends an TCP-ACK. We can see the duplicate TCP-ACK with sequence ID 395964 being generated every time 6 receives any TCP-DATA packet. By observing the graph more closely, we can find out the enormous increase in the delays. Especially the ACK flows. Consider the duplicate TCP-ACKS that are being generated with sequence ID 395964. The first ACK takes a total of 6.37601 millisecond to reach from node 6 to node 2. This is because the route from node 2 to node 6 was being built and there was not much TCP-DATA traffic fighting against the TCP-ACK. The next TCP-ACK that is being generated, at around 13.08 seconds, takes 99.7257 milliseconds, 16 times larger than the previous TCP-ACK, to reach node 6. This is because of the TCP-DATA traffic against which it has to compete. The following are the consequences of NRTE and route error that caused the delay:

- There are more Duplicate TCP-ACKs being generated which are delayed in the network. This is because of the hole in the window.
- The TCP retransmit for the lost packets also begins late because of late arrival of TCP-ACK at node 2.
- More head of line blocking for both TCP-DATA and TCP-ACK packets causing larger idle times.

This shows the need to avoid "Head of line blocking problem" and stresses on the hazards of NRTEs and route errors.

5.3.7 DMAC Remains Effective

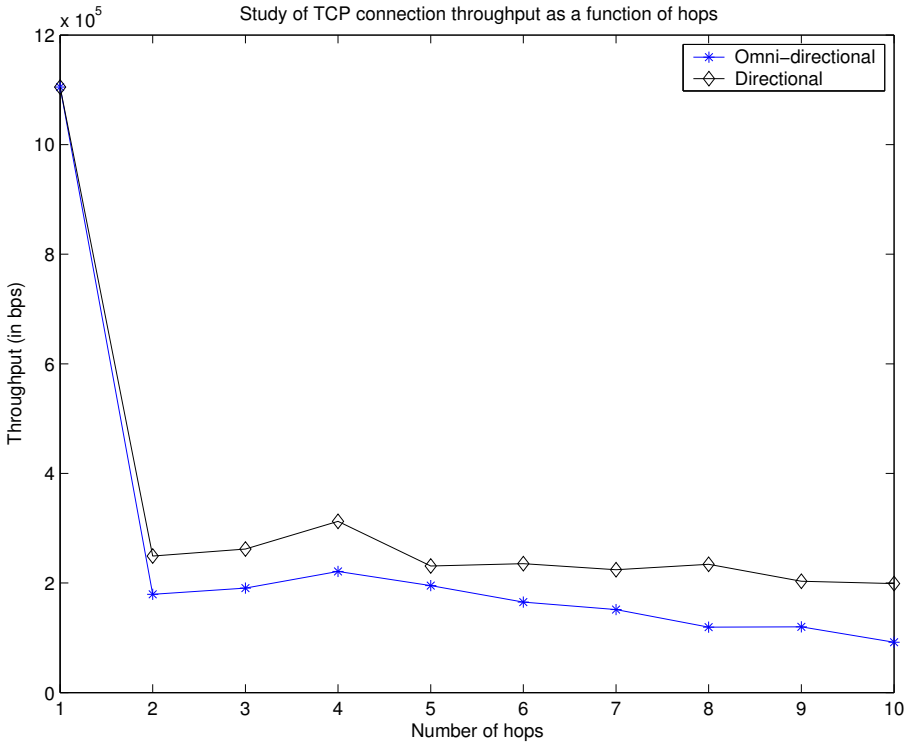


FIG. 5.18. Throughputs for directional and omni-directional MAC as a function of hops

If we observe the throughput for TCP over chain topology in Table 5.2, the throughput is around 51% higher than that of omni-directional MAC. Graph 5.18 shows the improvement of throughput obtained while varying the number of hops for a TCP connection with window size 32. These increase in the throughput can he explained by *Higher channel reuse*.

In the chain topology, consider a node placement scenario where each adjacent node is placed at enough distance so that the distance between the nodes is equal

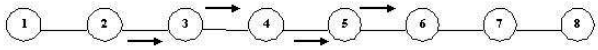


FIG. 5.19. More channel reuse

to the omni-directional range as shown in Figure 5.19. In case of omni-directional transmission, two nodes can simultaneously transmit only if they are at least 3 hops away from each other. In case of directional transmission, two nodes can go on transmitting in parallel if they are just 2 hops away from each other as shown in the Figure 5.19. The arrows marked above the line connecting the nodes can go together and the ones marked below can go on simultaneously.

The obtained results did not follow intuition because the availability of packets is less than that needed to achieve the theoretical limit. In the above example, node 4 will transmit to node 5 only when it has a packet to send to 5. The source dictates the flow as explained in Section 5.3.6. Hence, we see that most of the time there are not enough packets at 4 to send it to 5 when 2 is transmitting to 3. If the source pauses its transmission, then there are not many packets in transition to see the channel reuse. Thus the throughput is not realized to its fullest extent. The theoretical limit cannot be reached by increasing the sending rate. As the sending rate increases, the source becomes greedy and pushes too many packets towards the destination. This results in increased packet drops and NRTEs. This effect is explained in detail in Section 5.3.6.

Transmission Concurrency Analysis: For a omni-directional system, if node 2 and node 3 are communicating, then nodes 3-4 or nodes 4-5 cannot communicate. Transmissions in chain nodes can go on only if they are at least 2 hops away from each other. In case of a directional antenna system, node 4-5 can communicate when 2-3 is going on. Such parallel transmissions yield a greater throughput.

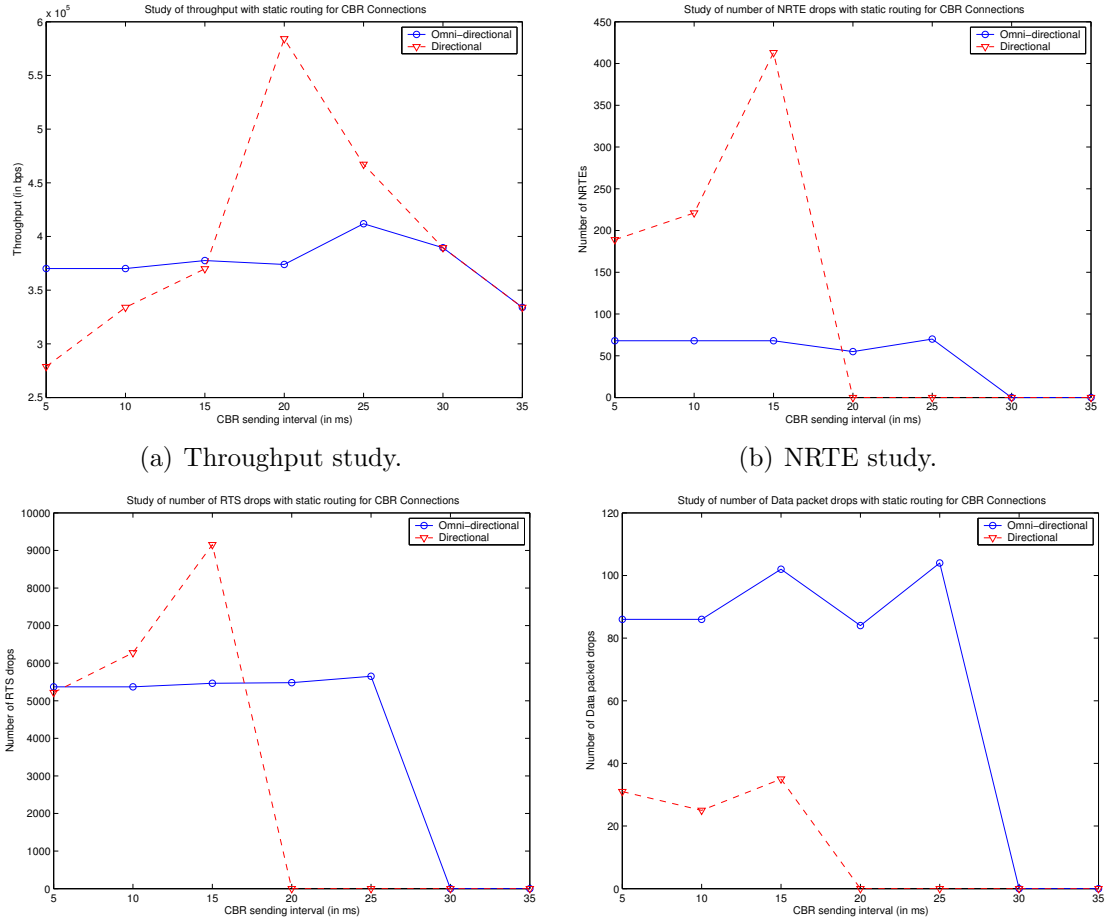
Relaxed Exposed Terminal problem: Consider the chain topology in Figure 5.1. If node 4 is sending a data packet to node 5, and node 2 wants to send a packet to node 3. In case of omni-directional MAC, even if node 2 sends an RTS to node 3, node 3 will not respond to node 2 because of the ongoing transmission between 4-5. This is called as the *Exposed terminal problem*. However, in DMAC because of the directional nature of the signal, node 2 and 3 can go on when nodes 4-5 are communicating. This relaxed exposed terminal problem yields higher throughput.

5.3.8 CBR analysis using static routes

The effectiveness of directional MAC is overshadowed by the routing protocol and transport. The NRTE's that are generated will trigger route repair or route discovery mechanisms. Even though the nodes are not mobile in the above scenario, the routes are assumed to be lost. This increases the channel idle time and decreases the throughput. The TCP protocol will be suspect to congestion in case of a packet drop and will trigger its congestion control mechanisms which leads to lower throughput. To evaluate the effectiveness of directional MAC without these adverse effects, a chain scenario was subjected to CBR traffic. Nodes were configured with static routes to avoid the routing issues. The Graph 5.20 shows the comparison of 802.11 and DMAC under statically routed CBR traffic. The connection is 4 hops, from node 2 to node 6.

It can be seen that for very high traffic(e.g: 5 ms sending interval), the throughput of DMAC will be lower than that of the 802.11. After a certain threshold, DMAC outperforms the 802.11. For higher sending interval, the performance of 802.11 and the DMAC are similar. The explanation of the behavior under different rates is as explained below:

- *Low sending interval(Higher Traffic):* Consider the cases where the CBR send-



(a) Throughput study.

(b) NRTE study.

(c) RTS packet drops.

(d) Data packet drops.

FIG. 5.20. Comparison of omni and directional with CBR using static routes

ing interval is 5 ms to 15 ms. Graph 5.20(a) shows that the throughput of the DMAC is worse than the 802.11. Even though the number of RTS drops are almost the same for 802.11 and DMAC as shown in Graph 5.20(c), it can be seen that the number of NRTEs for DMAC is far higher than that of the 802.11 as shown in Graph 5.20(b). In case of 802.11, the RTS packet drops will be because of the exposed terminal problem. We conjecture that the ineffectiveness of handshake leads to RTS packet drops because of deafness in

DMAC. The deafness of the node to listen to its immediate neighbor when it is in conversation with the the other adjacent neighbor leads to inconsistent DNAV updates. These factors lead to higher NRTEs, resulting in packet drops and lower throughput.

- *Medium sending interval (Typical Traffic):* It can be seen that DMAC outperforms 802.11 in throughput at certain range of sending intervals(e.g: 20 ms to 30 ms) as shown in the Graph 5.20(a). This can be explained by the ability of the DMAC to initiate parallel transmissions as explained in Section 5.3.7. At such sending intervals, the packets move at a rate where parallel transmissions can be initiated at 2 hops away nodes. In case of 802.11, nodes which are 2 hop away cannot initiate the parallel transmission because of the hidden terminal effect. However, it was seen in Section 5.3.7, that DMAC can have parallel transmissions between the nodes that are two hops away. This increases the throughput of the channel. The RTS drops and NRTEs is zero in case of DMAC and is quite significant in the 802.11. The higher drops are because of the exposed terminal effect in 802.11.

- *Higher sending interval (Low traffic):* It can be seen in Graph 5.20 that after certain sending interval (e.g: after 30 ms), the 802.11 and the DMAC performs identical. This is because of the slower rate of packets. At such sending rates, the DMAC cannot make use of the advantage of parallel transmissions because of large gap between the two consecutive packets.

It can also be seen that the throughput, RTS drops and the NRTEs in omnidirectional is almost constant for different sending rates for higher and normal traffic whereas the DMAC fluctuates.

Chapter 6

NON-LINEAR CHAIN: EFFECT OF GEOMETRY

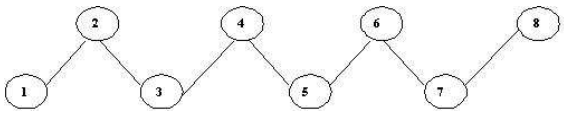


FIG. 6.1. Angular Placement

In case of a straight line placement of the nodes, there cannot be much channel re-use because of interference. This is because of the fact that the interference caused when one node is transmitting may affect the nodes that are in line with the transmission direction. Hence, the nodes that can listen to the transmission will set their DNAV table, to defer till the ongoing transmission is complete and then contend for channel. Also, there is significant interference between the transmissions since they are aligned in the same direction. If node 5 is receiving a packet from node 4, then the transmission from node 2 to node 3 may affect its reception because of the higher directional gain. So, if we place the nodes such that two parallel transmissions can go on together, then there must be higher reuse of channel and hence increasing the throughput. At the same time, directional RTS/CTS are more effective in a straight line topology because they "cover" all potentially interfering nodes. If the geometry

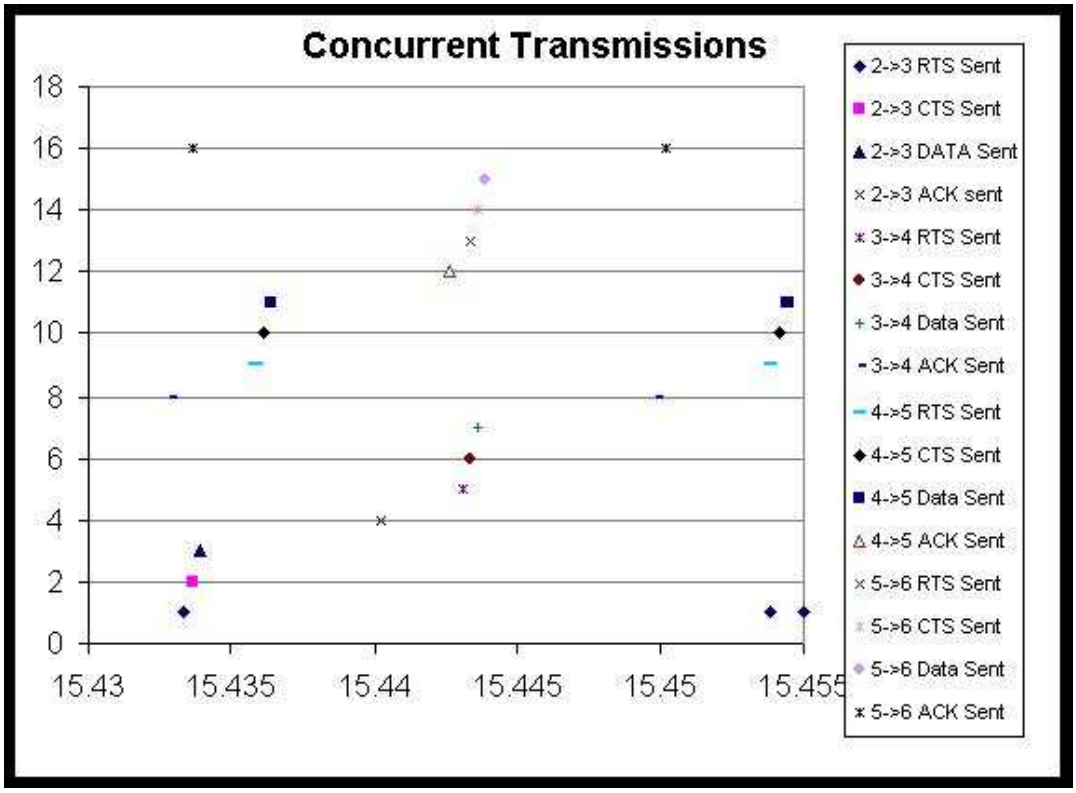


FIG. 6.2. Concurrent Transmissions

of the connection is different, the RTS and CTS may not cover possibly interfering nodes. Thus, it is not clear how such connection would behave.

The analysis in this chapter uses a chain where the angle between the edges connecting the nodes is controlled. We conclude this section by showing that the throughput observed is a intricate function of the the placement of the nodes.

6.1 90 degrees between node edges

Consider the zig-zag pattern of the node placement with 90 degrees between the node edges as shown in Figure 6.1. The application is set up in the same manner as the previous scenario. There is a TCP connection running from node 2 to node 6 with window size of 32. The distance between each node is 200m apart. We expect parallel

transmissions between any parallel legs of the connection since they do not interfere with each other. The result is observed is goes very much against our intuition. We see that the overall throughput in zigzag case is lesser than that we observed in omni. Its around 3/4th the throughput we observed in chain topology.This means that the channel reuse we expected is either not happening or there are adverse effects of the channel reuse. Observing the MAC layer plots, we see that there is parallel transmission going on between the parallel legs of the connection. A sample plot is as shown in Figure 6.2 shows MAC interaction from 15.43 sec to 15.45 sec where 2-3 and 4-5 are going in parallel. We can also see the transactions between 3-4 and 5-6 going concurrently. So, the throughput is not lower because of lower channel re-use factor.

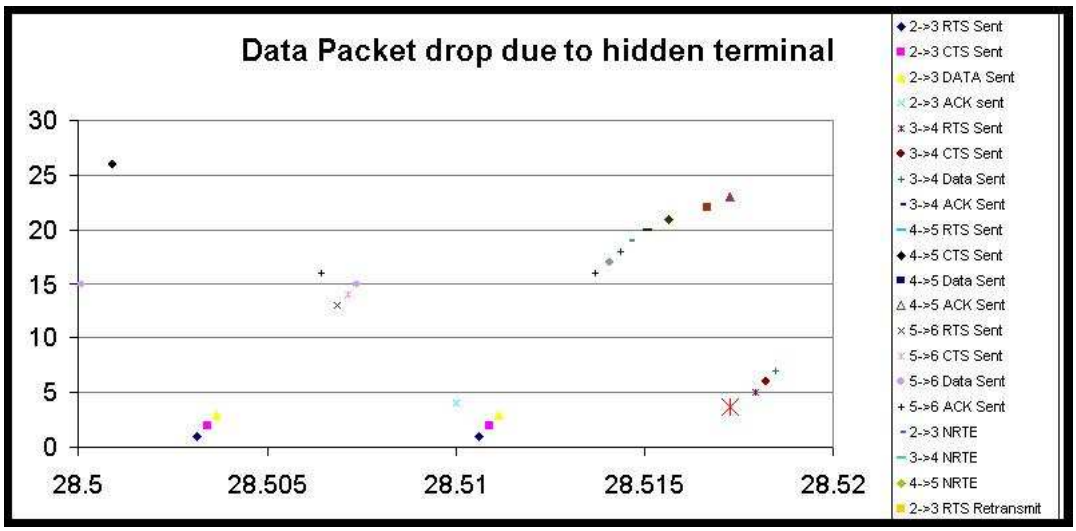


FIG. 6.3. Hidden terminal causing DATA packet drop

Heightened Hidden terminal problem:

The striking feature of this scenario is the number of MAC layer DATA packet losses observed. In case of a chain, there were 26 DATA packet lost. In case of zig-zag the number of DATA packet lost is 155 . This is an increase in a factor of 6. Consider

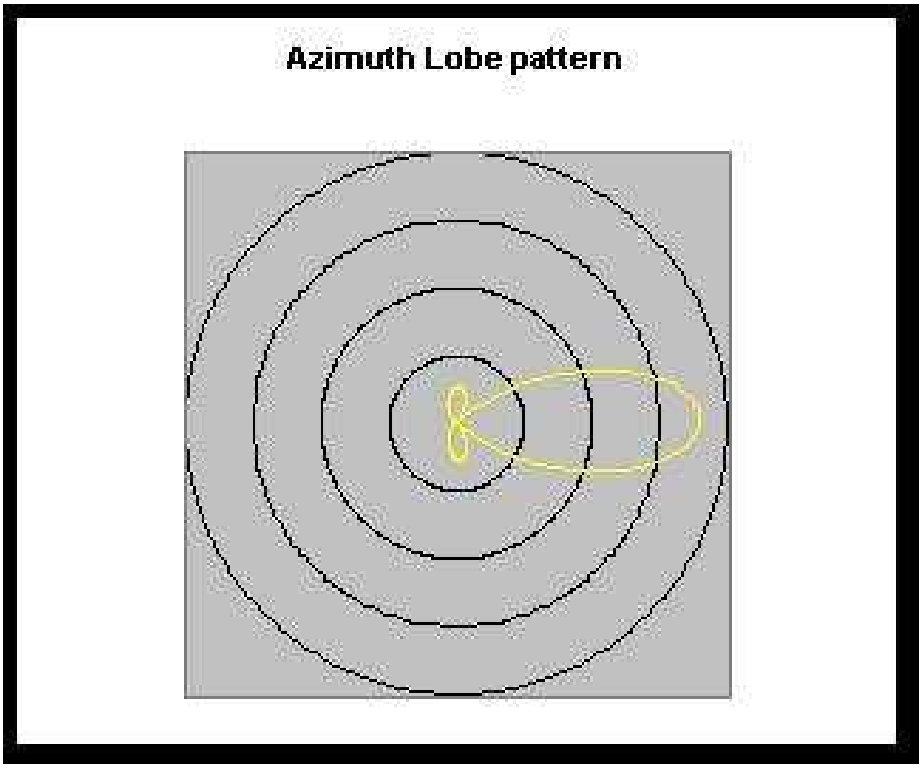


FIG. 6.4. Single Lobe

a sample MAC plot of a MAC layer DATA packet loss observed in the zig-zag pattern.

Consider the graph in Figure 6.3. Node 2 sends a DATA packet to node 3 at time 28.511126 after a successful RTS/CTS handshake. At the same time, 6 sends an ACK to 5. this packet is transmitted from 5 to 4. Then node 4 tries to send this ACK to 3. Since node 3 is receiving a much larger TCP-DATA packet, it is in still receiving state engaged with node 2. With directional transmission we expect the antenna to be totally deaf towards other sectors.

But, we see that it is not so. To explain the above effect, consider the Azimuth pattern for reception. Figure 6.4 shows a single lobe, the power in mW is plotted against the angle. We see that this pattern is aligned for 0 degrees and has maximum power at this angle. We can also observe the side lobes. There can be number of such patterns aligned for different directions. In the simulation used for these scenarios,

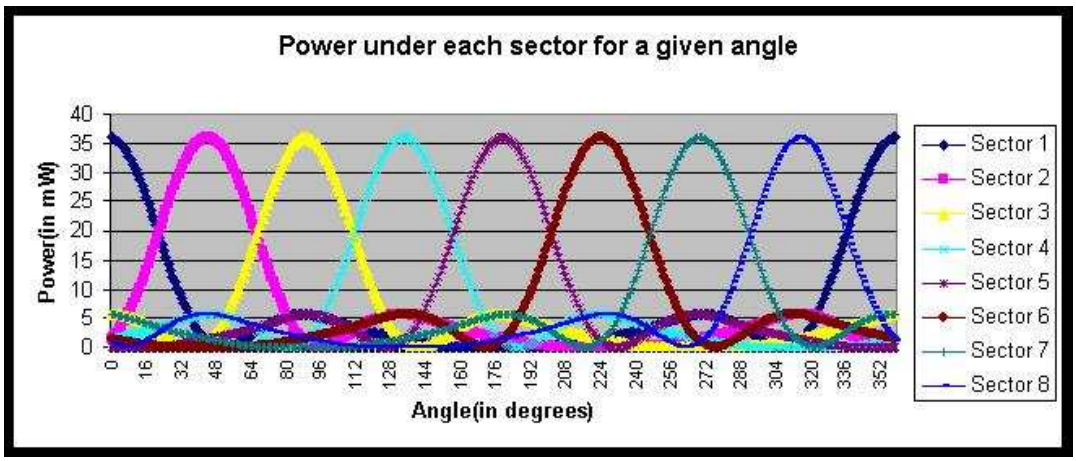


FIG. 6.5. Power at incident angle

there are 8 patterns for azimuth. They are numbered from pattern 1 to pattern 8 and there will be 8 such lobes each one aligned at 45 degrees from the previous lobe. The power v/s angle for each of the lobes are as shown in Figure 6.5. When a packet arrives at a node that is directionally aligned to a sector, the gain with which the node is going to receive the packet will be receiver gain as specified by the angle of arrival for the corresponding pattern. We see that when the TCP-ACK packet arrives at node 3, the directional receiver gain is not very low because of the side lobes in pattern 4. This gain is high enough to interfere with the current ongoing reception and hence causes the DATA packet being sent by node 2 to be dropped. Hence the numerous DATA packets lost is because of the reverse traffic causing the interference with the TCP-DATA packets. This is the prominent hidden terminal problem caused due to the directional nature of the RTS-CTS.

The RTS sent for sending TCP-DATA packets do not cause huge drops in TCP-ACK packets because of the asymmetric connection. The TCP-ACK packets are much smaller than the TCP-DATA packet. So the time for sending it is faster which reduces the chances of the RTS for TCP-DATA causing drops. However, this effect may be observed when the nodes at both end of the connection are trying to send

data packets or when two or more connections are headed through a common node in opposite direction.

The effect is not seen if CBR was chosen as the transport protocol instead of TCP for the same scenario because of the absence of the reverse traffic. In case of CBR, the throughputs for the chain topology and for the zig zag topology are almost same. Hence, the placement of nodes and the traffic flow plays a dominant role when the traffic is bi-directional either because of the the type of higher layer protocols used or because of multiple connections.

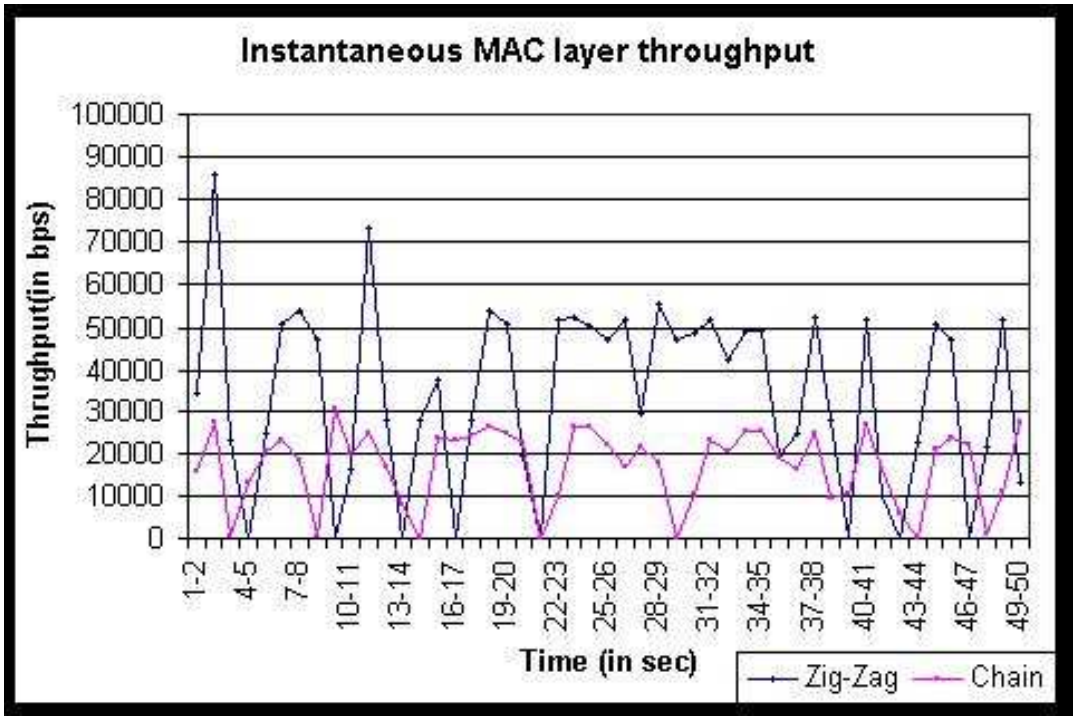


FIG. 6.6. Instantaneous throughput in case of zig-zag pattern

NRTEs causing large channel idle time:

The flow in zig-zag is also more little more choppy than the chain flow as seen in the Figure 6.6. It can be noted that the zig-zag pattern is less stable than the chain topology. Except for the middle part of the graph, the throughput is bound

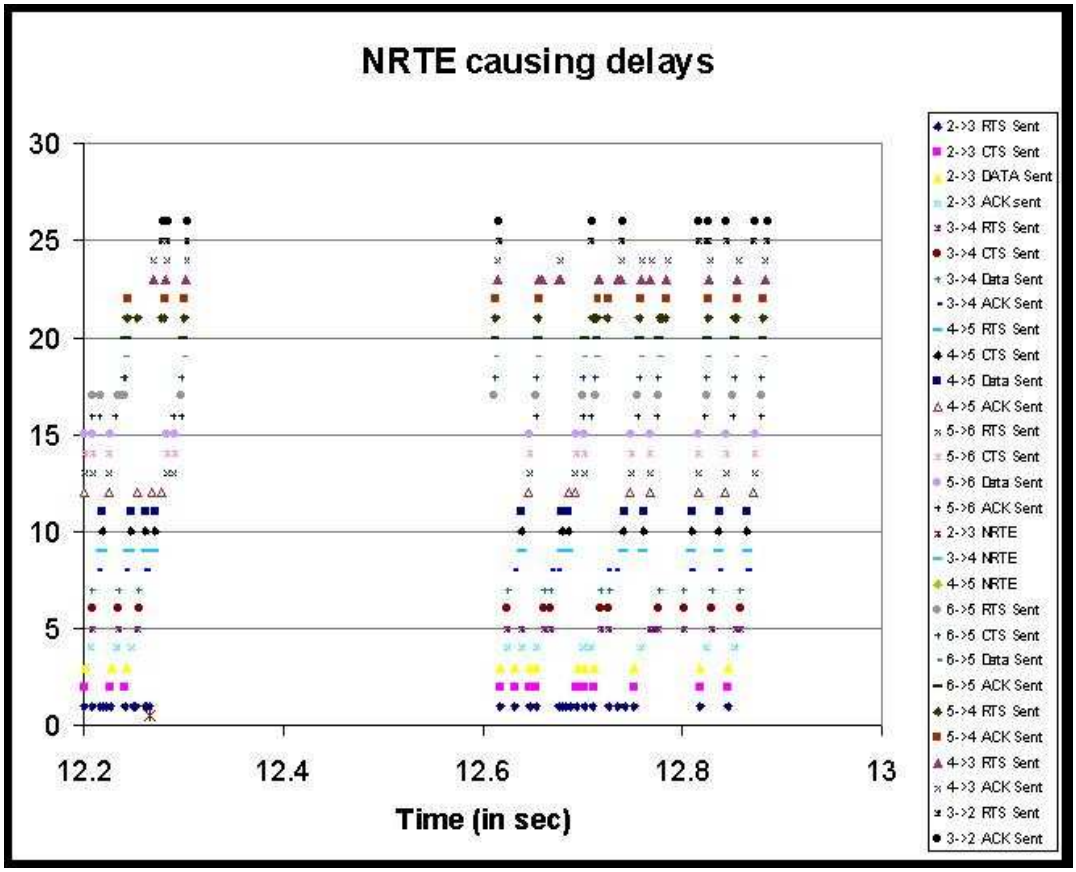


FIG. 6.7. NRTEs causing large channel idle time

to droop to zero if if there is steep climb in throughput. Even though the MAC layer throughput is higher than that of the chain topology, the realized throughput at application layer is much lesser than the throughput observed in chain topology. This is because of the umpteen DATA packets that are dropped. Most of these gaps are due to the reason of out-of-order delivery of data.

Consider the MAC plot graph as shown in Figure 6.7. There is an NRTE being generated at time 12.265972 that is marked by an asterisk near the Y value of 0. This NRTE does not trigger a route discovery process immediately. Node 6 waits to send a delayed ACK. After the timeout, it triggers an TCP-ACK transmission at time 12.611317. When this packet arrives at node 2, then it starts transmitting the

next packet. The route has already built for node 2 and it does not need to trigger a broadcast for route discovery. This NRTEs long term effect is even interesting than the short term delay explained above. The after effects are seen after around 0.6 seconds later. The consecutive packets after the short delay causes a hole in the TCP window at node 6 because of the TCP-DATA packet being dropped due to NRTE. This results in sending duplicate TCP-ACKs. Node 2 receives duplicate TCP-ACKs and then backs off. There is a retransmission of the packet again at time 14.5175 seconds. This large gap of around 1.6 seconds is because of the NRTE that caused out of order delivery.

6.2 Analysis based on Azimuth patterns

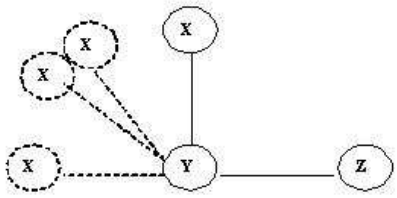


FIG. 6.8. Nodes placed at various angles

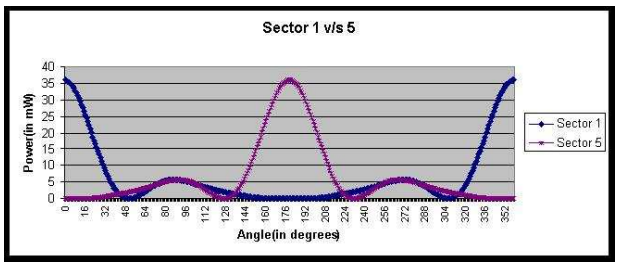


FIG. 6.9. Power gains in sector 1 and 5

In the chain topology that we have described above, we see that the sectors are perfectly aligned to the adjacent hosts. Hence the receiving nodes get the maximum

directional gain. In case of the zig-zag scenario, the angle between the adjacent nodes is 90 degrees. We expect the throughput to be higher as the angle between the nodes increase from 90 to 180 because of lesser interference between the two hop nodes. The throughput in case of 90 degrees is 229953 bps, whereas if we increase the angle to 120 degrees, then it 212656 bps. This can be explained because of the receiver gain as given by azimuth configuration. In the directional antenna we use, there are 8 patterns. Hence there is a lobe every 45 degrees. In case of zig-zag pattern with 90 degree separation between the nodes, the receiver is perfectly aligned to get the maximum gain in the lobe. Whereas in the 120 degree separation, the gain is reduced because of the relative interference of the lobes. At this placement, the maximum gain is not got for the receiving direction of arrival. To add to the problem, there is also an interference created if the other adjacent node tries to send a packet to the node that is already receiving the DATA packet.

The graph in Figure 6.5 shows the interfering patterns of the lobes. Consider a node Y which is perfectly placed to talk to another node Z with maximum gain in sector 1 as shown in the figure 6.8.

If the angle between the node edges is 180 degrees(chain topology), then the other adjacent node X's transmission will be received in sector 5 since the maximum gain will be in this sector among all the sectors. The interference caused will be minimum when nodes X, Y and Z are placed such that if the power of the sector at the which Y is listening to Z is significantly more than the power caused by interference of the other sector's power when node X transmits a packet to Y. The difference between these powers should be a significant indicator of the interference power. For example, consider the 180 degree separation scenario. In this case, node Y listens to node Z at sector 1 and node Y listens to node X at sector 5. We see that the best angle required angle in the case when the the ebb of one wave will be right below the flow of the other

wave. Hence the difference is always maximum benefiting the transmission going on in that sector and preventing the interference caused by other sector. As we can see in the Figure 6.9 which is a brief version of the Figure 6.5, this is the best possible configuration to prevent the neighboring nodes to interfere. Hence, lower interference the throughput obtained is higher. The difference if power when node Y is listening to node X and node Z tries to transmit to node Y is 35.92mW at node Y.

Now in case of 120 degree separation between the node edges, node Y will talk with node Z in sector 1 and with node X in sector 5 because sector 4 gives the maximum gain for 120 degrees. We also observe here that the gain got while receiving packet from node X at sector 4 is not the maximum gain for sector 4. If we observe the powers for sector 1 and sector 4 in the Figure 6.5, that the ebb and flow do not lie in a fashion that reduces the interference. This causes significant interference which may lead to packet drops causing lower throughput which is exactly what we observed.

In case of 90 degree separation between the node edges in the zig-zag pattern, node Y will talk to node Z in sector 1 and with node X in sector 3 as shown in the Figure 6.5. We note that here that there will be an interference caused when node Y is talking with node Z and node X tries to send a packet to node Y. However, we also observe that node Y gets the highest possible gain when talking with node X which was not the case when the nodes were arranged in 120 degrees apart. Hence the difference observed between the the waves is 30.43mW where as the difference observed when 120 degrees node separation was 26.17mW. This power can be considered as relative gain for the transmission. The higher this factor is the lower will be the interference. Hence the throughput observed in 90 degrees is better than the throughput observed in the case of 120 degree separation.

By looking at Figure 6.5 we can also guess the throughput got at various other

Angle	Throughput	Power difference
90 deg	229953 bps	30.43mW
120 deg	212656 bps	26.08mW
135 deg	245568 bps	34.61mW
180 deg	312354 bps	35.92mW

Table 6.1. Throughput at various angles

angles. We can see that angles 45 degrees and 135 degrees, there should be lesser interference and hence better throughput. Note that we cannot choose very small angles between the node edges. This is because of the fact that if the angle between nodes X-Y and Y-Z is reduced, then X and Z will come close enough that they can directly communicate instead of communicating with node X. This plays the limiting factor. For the 200m separation between the nodes, this minimum angle should be 90 degrees separation. Hence creating scenarios with 45 degree separation is not practical. With 135 degree separation we see node Y will communicate with node Z in sector 1 and with node X in sector 4. The difference between the waves is 34.61mW when node Y is listening to node X and node Z transmits a packet to node Y.

The application level throughputs for various angles are as shown in the Table 6.1.

Sorting the difference in the powers we expect the interference to be in the order 180 degrees, 135 degrees, 90 degrees and 120 degrees. Hence the best possible throughput should be obtained in the same order with 180 degrees getting best possible throughput and 120 degrees getting least possible throughput. This is confirmed by the values in the Table 6.1.

6.3 Conclusions

The analysis brings up several interesting aspects of Directional system that are different from omni-directional system. Overall, the the DMAC running on directional

antenna performs better than the omni-directional antenna.

There was a significant improvement in throughput though the number of packet drops depends on the topology and connection pattern. We observe the factors like deafness, backoff algorithm and the positioning of the nodes plays a very critical role for a good connection throughput. We conclude that the current RTS-CTS mode of handshake is ineffective and a more comprehensive handshake protocol is needed at MAC layer for effective use of directional antenna. An interesting scenario was also seen where deafness helped to gain throughput. The azimuth pattern and the geometry of topology also plays the role in deciding the channel utilization and packet drops. It can be concluded that the effectiveness of DMAC is a function of many intricate characteristics of the directional antenna system.

The DMAC can be improved in many directions. With the present version of the DMAC which do not use the features of the directional system effectively, there was a good improvement seen when compared to the omni-directional system. We believe that the with an appropriate DMAC protocol, the improvement seen can be much larger. The problems faced while using directional antenna are much different and harder than the omni-directional counterpart. This study helps to highlight certain characteristics of directional system which, we hope, will help to design a better DMAC.

6.4 Future Work

There is lot more to explore in directional antennas. This part of the thesis tries to capture the overall effects observed while using DMAC. Extending the study to delve deeper would be interesting and can unearth more intriguing effects of directional antennas. The analysis was complicated because of several factors of directional antenna that were interwoven. An effort was made to isolate the characteristics and to

study the effect of those characteristics. The analysis can still be further improved by studying the CBR traffic in detail over directional antenna. Switched beam antenna was analyzed because of its simplicity. It would be no surprise if steerable antennas has a lot different characteristics than switched beam ones. Studying steerable antennas can be one more direction of continuing the analysis.

Multiple connections and their effect on each other can come up with interesting features in directional antennas. It would be interesting to study scenarios with multiple connections in different geometrical topologies. After the careful understanding of chain topology, it can be extended to study more real world topologies like grid and random. At the present, we believe that there should be more in depth understanding of the directional antennas and DMAC to continue with this study. The effect of the number and pattern of connection can be altered and analyzed.

Several improvements can be made to existing DMAC by isolating the undesirable characteristics from this analysis. Infact, the thesis tries to propose a new DMAC for solving two such problems.

Chapter 7

AVOIDING HEAD OF LINE BLOCKING IN DIRECTIONAL ANTENNA

Several features of directional antennas are not completely harnessed in the current DMAC. One such powerful feature which is suppressed is the greater channel utilization. The *Head of line blocking* was described in Section 3.5.2 as one of the drawbacks of DMAC which discounts the free channel space. This chapter proposes a modification to DMAC to solve the *Head of line blocking* problem.

7.1 Overview

Among the queue of packets to be transmitted by a node, the existing directional MAC layer chooses the head of the queue as the packet to be transmitted. The other packets must wait till the current transmission is complete even though the channel may not be busy in their respective directions. The severity of such *Head of line blocking* was analyzed in the Section 5.3.5. This study addresses two issues in directional MAC. The first contribution highlights the inefficiency caused by FIFO Queuing mechanism while using directional antenna and proposes to use a different queuing policy which could take advantage of the channel utility factor provided by the underlying antenna system. Our results indicate that by using a greedy approach

to schedule the packet which has the least wait time increases the overall throughput and end-to-end delay. The second contribution identifies the inefficiency that exists in the virtual carrier sensing mechanism of the current directional MAC protocol and we propose a new mechanism to address this issue.

7.2 Existing Queuing policy

We now explain the queuing policy implemented in the existing MAC and routing protocols. This policy is observed by omni-directional 802.11 and DMAC. When a data packet needs to be transmitted, it is handed over by the routing layer to the MAC layer and the MAC layer transmits the packet with appropriate handshake. These packets are put into a queue by the routing layer and the MAC layer will pick up the packet to be transmitted from this queue.

The scheduling policy generally observed in the DMAC layer is "Strict priority scheduling". In such a scheduling, each packet is assigned a priority and it is ensured that among all the packets in the queue the higher priority packet always gets transmitted before the lower priority packet. There is a fixed number of priorities which can be assigned to the packet. For each of the priority a FIFO queue is maintained. For a given priority, the packets transmitted will be in the order in which they were inserted into the particular FIFO queue.

The routing layer will insert the packet into the appropriate FIFO queue based on the priority of the packet. When the MAC layer polls for the next packet to send, it first searches in the highest order priority queue. The MAC layer picks up the first packet inserted into the highest order priority queue. If there are no packets in this queue, then it will search in the lower order priority queues.

Although the control packets are sent with higher priority, all the data packets have the same priority. Hence, effectively there is a single queue with FIFO scheduling

for all data packets. This queuing policy works well for nodes using omni-directional antennas. Interestingly, our simulation results show that with the existing DMAC layer, we are not able to exploit the benefits of directional antenna to the full extent using the above queuing policy. The packet queuing policy at the MAC layer plays a significant role in the overall performance of the network. This chapter proposes a queuing policy tailored for directional antennas.

The following example illustrates "Head of line blocking" problem while using the FIFO queuing. Among the queue of packets to be transmitted by a node, the existing directional MAC layer chooses the head of the queue as the packet to be transmitted. The other packets must wait till the current transmission is complete even though the channel may not be busy in their respective directions.

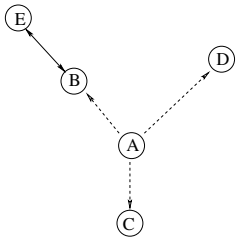


FIG. 7.1. Head of Line blocking

Consider the scenario in Figure 7.1 where a node A is communicating with nodes B, C and D. Let node A's queue have packets destined to nodes B,C and D waiting for transmission. Nodes B and E are engaged in communication. Node A has to wait till the communication between node B and E is complete. This is logical if the packets are being sent in omni-directional mode. If node A starts sending packets, then it can interfere with the ongoing communication between nodes B and E. However, in case of directional mode, node A can start transmitting the packet to node C when B and E are communicating since this does not interfere with the communication between B and E. With the current implementation, even if the channel is free in the direction

of node C and node D, node A cannot send the packet to node C and node D because of FIFO queuing policy. This creates a undesirable wait time for node A.

For example, if node A could sense that the medium is busy in the direction of B and the channel is idle in the direction of node C, then it could schedule the packet for node C instead of waiting on node B. To enable this, node A should scan the packet queue and choose the one which has the least wait time. In this study, we propose a scheme based upon a greedy approach, to minimize the wait time resulting in greater spatial reuse.

7.3 Improved Queuing to Eliminate HoL Blocking

To overcome the inefficiency of *Head of line blocking*, the MAC layer should be capable of finding out the channel state for the direction in which a packet needs to be sent. There should be a mechanism to find out the time interval for which the channel might be busy in a particular direction. If there are multiple packets to be sent and the above mechanism is present, then the packet to be transmitted next should be the one with the least wait time.

Sensing the channel for each packet before sending a packet can be ineffective. We now describe the mechanism by which a packet is selected for transmission based on the information present in the DNAV.

7.3.1 Using DNAV for scheduling

Each node maintains a directional NAV (DNAV) as explained in the Section 3.3. With the DNAV table in place, scanning the packet queue for the least wait time is now reduced to the task of checking the wait time in the DNAV for the packet's angle of transmission. The wait time can be found out by using Equation 3.6 if the direction for the packet is known.

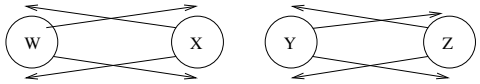


FIG. 7.2. Deafness causing failed DNAV updates

The actual state of the channel may not always be reflected by the DNAV. Consider nodes W, X, Y and Z in Figure 7.2. Let nodes X and Y be within transmission range of node W. When the node W is communicating with node X, node Y should ideally mark its DNAV appropriately indicating the wait time in the direction of node W. If node Y is busy communicating with node Z, then node Y will be "deaf" to node W. This inhibits the accurate DNAV update at node Y. Since the state of DNAV does not reflect the channel state in the direction of node W, all calculations using the DNAV entries may not be correct. The study in this thesis does not try to solve the deafness problem. Deafness causes under-performance of our protocol. In the presence of a reasonable mechanism to reduce the deafness, we conjecture that our protocol would perform better. An attempt to solve deafness are demonstrated by Choudhury et al. [4]. We now describe the approach taken to measure the angle of transmission and the scanning of packets for least wait time.

7.3.2 Transmission angle calculation method

When a node receives a packet from the physical layer, there is an inbuilt ability in the antenna to figure out the approximate angle of arrival of the signal. If the node is locked to coverage pattern (a sector, in case of switched beam), then the angle of arrival is considered as the angle of maximum gain. If the node is listening in omni-directional mode, then the angle of arrival is marked as the coverage pattern for which the gain is maximum. In switched beam antenna, this will be the main lobe of sector in which the transmission is heard.

The antenna coverage pattern is not an ideal conic section for each sector. There

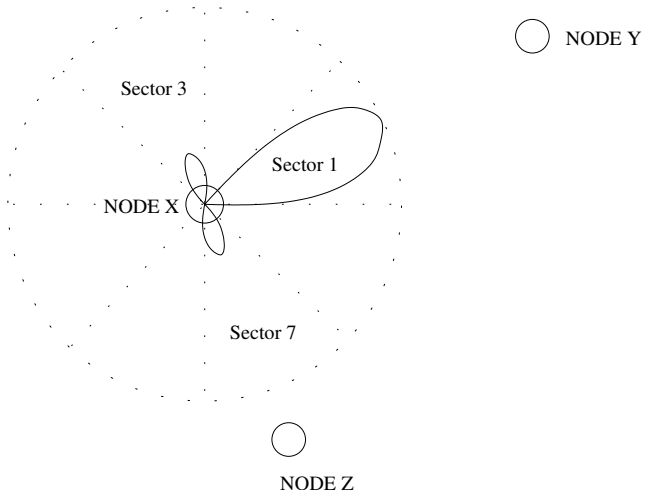


FIG. 7.3. Wrong angle of arrival marking when side lobes are present

will be side lobes and possibly tail lobe present in the actual coverage pattern of the antenna. Figure 7.3 shows a simple coverage pattern of one particular sector in a switched beam antenna for Node X. Let sector 1 be the active sector in which the node X is listening as shown in Figure 7.3. There are side lobes present which extend into sector 3 and sector 7. This means that any packet which arrives at sector 3 or sector 7 with sufficient power will also be intercepted by the node when it is listening to sector 1 because of the receiver gain present in the side lobes. So when node X is locked to sector 1, it can still hear from node Z.

In the ideal conic section coverage pattern, node Z could be heard only in sector 7. Even though the side lobes are shorter than the main lobe, the gain in the side lobes is higher than the omni directional gain. This makes the side lobe interference more vulnerable.

There is an inherent error in calculating the angle in the above manner if the side lobes' effect is introduced. Consider the case where node X is locked to sector 1 and is about to communicate with node Y as shown in the Figure 7.3. If node Z sends a packet, this packet can be still received with the side lobes. Since the maximum gain

is found in the main lobe of the coverage pattern, the angle of arrival for the packet transmitted from node Z is wrongly marked. It is marked as though node Z could be reached by sector 1, while the actual sector from which node Z could be reached is sector 7. Even though the node Z could listen to a packet transmitted from sector 1 of node X through the side lobe effect, we believe that the packet must always be transmitted along the main lobe pointing towards the direction of the recipient.

Consider the side effects of updating the DNAV with wrong angle of arrival for node Z at node X. If node X wants to send a packet to node Z and the channel along sector 1 is busy, X will wait till the channel is idle even if channel along sector 7 is idle. This is clearly undesirable as node X should sense sector 7. In fact, we observed that this effect is taking place in the simulation.

Our queuing policy will be inefficient if such false updates from DNAV are used to judge the wait time to send the packet. Hence another modification to the existing implementation of DMAC is made. When the node is not locked to any of the sectors, it will sense the channel in omni-directional mode. If the antenna is locked toward a sector, then the DNAV is not updated with the angle of arrival. It is updated only when the antenna is in omni-directional mode. This allows for the DNAV to record only the correct angles. We observed a increase in the throughput when such a policy to update DNAV was followed. Note that the angle update does not take place in such cases. However, since the packet was captured properly, the DNAV wait time will be updated for the previously recorded correct angle.

Note that only the angle of arrival is not updated if the node is not in omni-directional mode. If the node has captured the packet, then two important information can be got by the packet. One is the angle of arrival of the signal and other is the time period for which the channel is busy. The time period for which the transmission goes on has to be updated in the DNAV table. This is a relevant information

that is got from the packet. Hence, the DNAV time is updated when the node is in directional mode or omni-directional mode but the angle of arrival is updated only if the antenna is in the omni-directional mode.

7.3.3 Buffering of packets in the MAC layer

The existing routing layer inserts the packet to be transmitted by the MAC layer into a queue which is referred as *Interlinking queue*. There is an additional queue maintained in our implementation called as the *MAC Queue*. The MAC layer dequeues the packets from the *Interlinking queue* and buffers the packets in the *MAC queue*. The MAC layer always dequeues the packet with the least wait time for transmission from the *MAC Queue*.

By setting the appropriate buffer size for the MAC queue, the number of packets to be examined each time can be adjusted. By adjusting the MAC queue buffer size, we can insert a specified number of packets into the MAC queue. These packets can be scanned each time when a new packet needs to be transmitted. This reduces the computation at the node considerably while preserving the ability to examine various packets.

In the proposed protocol the MAC Queue is implemented as a linked list. Each entry of the MAC Queue has a pointer to the packet, next hop id, angle of Transmission, priority of the packet and the time at which the NAV expires.

Some of the design issues encountered for our protocol design are described in the following paragraphs.

Priority of the packet: The Interlinking queue in DMAC is implemented as a set of FIFO queues, one for each priority. While buffering the packets at MAC layer the priorities are respected. If there are two or more packets with the same priority and no other packets with higher priority, then the one with the least wait time is

scheduled for next transmission.

Hence, if it is observed that there exists *packetA* that has a lesser wait time than *packetB* and if the priority of *packetB* is higher than that of *packetA*, then *packetB* is picked up as the one to be transmitted next. If there are two or more packets with the same priority and no other packets with higher priority, then the one with the least wait time is scheduled for next transmission.

Handling omni-directional packets: While dequeuing the packet from the Interlinking queue, it may happen that the head of the Interlinking queue is a broadcast packet or a packet whose next hop is not found in the AoA cache. Broadcast packets are destined to all neighbors and are sent omni-directionally. Further, if the node does not have an entry to the next hop in the AoA cache, then the packet cannot be transmitted directionally because the angle along which the packet needs to be sent is unknown. Such packets are sent in omni-directional mode.

Typically, omni-directional packets have the larger wait time than packets that need to be transmitted directionally because they are blocked by any transmission from any direction. There may be more packets in the MAC queue which are queued for transmission. Since we are dequeuing the packets from the Interlinking queue in a strict FIFO order, the other packets that are present in the MAC queue are scheduled before the omni-directional packet at the head of the Interlinking queue. Therefore we chose not to buffer these packets into the MAC queue.

An omni-directional packet is scheduled for transmission only after all the packets in the MAC queue are transmitted. Once the MAC queue is empty, the omni-directional packet is fetched from the Interlinking queue and transmitted. Buffering is resumed after the transmission of this omni-directional packet, if the new head of the Interlinking queue is not a omni-directional packet.

When the MAC layer decides to send the next packet it will first try to buffer

packets from the Interlinking queue and will insert it into the MAC Queue. The MAC queue is checked to find the best packet that can be transmitted by using a greedy algorithm.

The algorithm to pick up the packet to be sent is described in 7.3.3.

Algorithm 1 Algorithm to pick up the packet from the Interlinking queue

```
while (Interlinking Queue is not empty) ∧ (number of pkt in MAC-Queue <
QUEUESIZE) do
  {Check the packet at the head of Interlinking queue. Do not dequeue it.}
  P = Packet at the head of Interlinking queue
  if P is a packet that is to be sent directionally then
    P = Dequeue the packet from the Interlinking queue.
    Insert P to the MAC Queue
  else
    {It is an omni-directional packet}
    break the loop
  end if
  if MAC Queue is not empty then
    PktTransmit = Select the packet which has the least wait time respecting the
    priorities from the MAC Queue;
  else
    if Interlinking queue is not empty then
      PktTransmit = Fetch from the Interlinking queue
    else
      {There is no packet to be transmitted.}
      return
    end if
    return PktTransmit
  end if
end while
```

7.4 Performance Evaluation

The QualNet 3.6 simulator [20] which has inbuilt support for directional antenna was used for simulations. Table 7.1 lists the relevant simulation parameters. We used *Strict priority scheduling* for the packets with the number of priority values set to

Parameter	Value
Omni-directional range	250m
Directional range	450m
Directional antenna model	Switched beam
Mobility	none
Propagation Channel Frequency	9.14 * 10^8 Hz
Path loss Model	Two Ray
Transmission power	24.5 dBm
Receiver sensitivity	-68.1 dBm
Directional gain	10.0 dB
Antenna Model	Switched Beam
Directional NAV Delta Angle	22.5 degrees

Table 7.1. Simulation Parameters

three. Hence there are 3 FIFO queues.

In order to make sure that the improvements are not simply due to increase in the overall queue size because of the addition of MAC-Queue, we decided to keep the overall queue capacity in our implementation the same as that in the case of original implementation. More specifically, the length of the $MAC\ Queue + Interlinking\ Queue$ in our implementation is equal to the length of the $Interlinking\ Queue$ of the original implementation. The length of the $Interlinking\ Queue$ is set to 50000 bytes for the original implementation. The size of $Interlinking\ Queue$ in our implementation is reduced depending on the $MAC\ Queue$ size and the packet size. Static routes were used for the simple scenario and the grid topology.

Additional complexity is involved to manage the dynamically space allocation for MAC Queue because of linked list data structure. This can be avoided by having a fixed MAC queue of a constant size. Since the maximum MAC Queue size is fixed and does not vary, this does not change the behavior of the protocol while it eases the deployment complexity. There is also no extra space consumed by the proposed protocol because the size of the MAC Queue and Interlinking queue will be equal the the size of the Interlinking queue in the existing protocol.

In the remainder of this section we first present simulation results with a simple hand-crafted topology followed by a more complex grid topology. We do not claim that these topologies and connection patterns are representative scenarios, but instead use them as a vehicle to demonstrate the inefficiencies of the existing DMAC implementation. The reason for selecting these topologies is two fold. First they are simple to analyze and second we believe that in many real-world scenarios the problems shown with these topologies will reappear.

In case of CBR connection, packets are sent at fixed time intervals called "sending interval". Packet "sending rate" is the inverse of sending interval. Our protocol performs better when each node has to forward or originate packets for different destinations which can be reached by different sectors. Having such nodes creates *hotspots* in the scenario.

7.4.1 Simple Topology:

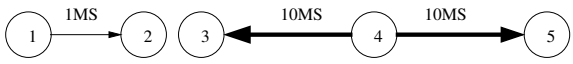


FIG. 7.4. UDP connection from 1-2 will block connection 4-3

Figure 7.4 shows a simple hand-crafted scenario with all connections of type CBR. The arrows indicate the direction of flow of the traffic. The packet sending interval (in milliseconds) is written across the lines. The nodes are set up such that node 4 is within the reception range from nodes 1 in directional mode but node 5 is out of reception range from node 1. In omni directional mode, node 4 is within range of nodes 3 and 5. Node 4 is the source of two CBR connections as shown in the Figure 7.4. CBR connection from nodes 1-2 is the *throttling connection* that is in line with the node 4-3 connection but running in the opposite direction.

As shown in Figure 7.6, we now systematically vary the rates of the connection

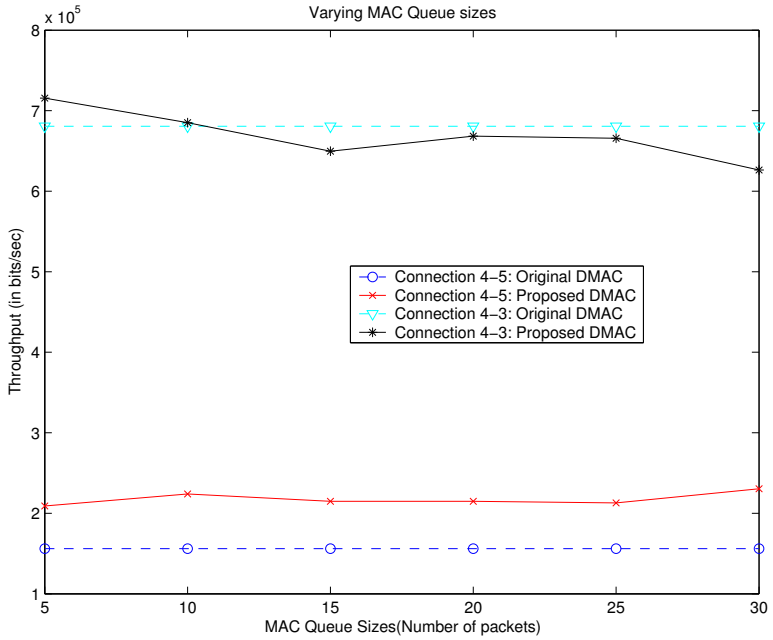


FIG. 7.5. Study of Throughput as MAC Queue size is varied

4-3 and 4-5 and study the effect with the proposed queuing policy. If the sending interval of the connections originating from 4 is low, there is more competition for the channel between the connection 4-3 and 1-2 since both are operating at low sending intervals. The connection 4-3 which in line with the throttling connection will have to compete for the channel in the specific direction more vigorously. For example, if the interval of connection 1-2 is set to a low value, then the rate at which this connection sends the packets is high. Hence, node 4 has to compete harder to send the packet to node 3. This results in increased wait time to send a packet to node 3. So, the throughput of connection 4-3 decreases. Since node 4 has two connections going out, the packets destined for node 5 will be blocked if the packet at the head of the queue is destined for node 3 and FIFO policy is observed.

The protocol implemented in this study is able to solve this problem. By observing that the channel is idle in the direction of node 5 and there are packets destined for node 5, node 4 picks up the packet and delivers it to the node 5 instead of spending

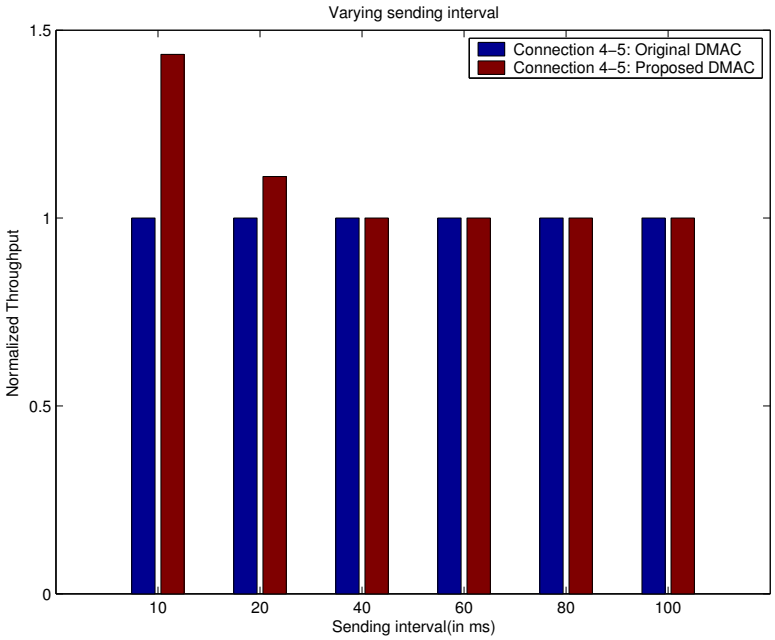


FIG. 7.6. Study of Throughput as Sending interval is varied

the time waiting for the channel to be idle in the direction of node 3. on for the channel between the connection 4-3 and 1-2 since both are operating at very low sending intervals. The connection 4-3 which in line with the throttling connection will have to compete for the channel in the specific direction more vigorously. For example, if the interval of connection 1-2 is set to a very low value, then the rate at which this connection sends the packets is very high. Hence, node 4 has to compete harder to send the packet to node 3. This results in increased wait time to send a packet to node 3. So, the throughput of connection 4-3 decreases. Since node 4 has two connections going out, the packets destined for node 5 will be blocked if the packet at the head of the queue is destined for node 3 and FIFO policy is observed. The protocol implemented in this study will be able to solve this problem. By observing that the channel is idle in the direction of node 5 and there are packets destined for node 5, node 4 picks up the packet and delivers it to the node 5 instead of spending the time waiting for the channel to be idle in the direction of node 3.

One can see that for lower sending rates (higher sending intervals), the original implementation and our implementation will give similar results. With higher sending interval node 4 has a longer time to send a packet to node 3 and 5 before the next set of packets are generated. This makes it possible for node 4 to flush out the packets before the arrival of the next set of packets. The maximum gain of our implementation is realized when the *hotspots* created in the topology.

We now vary the *MAC Queue* length from 5 packets to 30 packets keeping CBR packet size constant (set to 1536 bytes). Sending interval of the throttling connection 1-2 is set as 1 packet every 1 ms. The goal of this particular experiment is to demonstrate the effectiveness of greedy queuing policy compared to FIFO policy. Therefore the sending interval of connection 1-2 is set to such a low value. It will therefore keep the channel on left side of node 4 busy most of the time. The sending intervals of connections 4-3 and 4-5 are set to 1 packet every 10 ms. In Figure 7.5, the X-axis indicates the MAC-Queue capacity (in terms of packet) and the Y-axis indicates the overall average throughput in bits/sec. As shown in the figure 7.5 the connection 4-5 has has higher throughput, even as high as 47% of the original throughput is observed. The unfairness that is present in the original DMAC is reduced when compared to the original implementation.

An interesting observation can be seen in Figure 7.5 where the throughput of connection 4-3 is higher than the throughput of connection 4-5. Since there is a throttling connection next to node 3, the throughput of connection 4-3 is expected to be lesser than that of connection 4-5. The results prove the opposite. The result can be explained by considering the interference caused by directional transmission. At node 3, there is lesser interference since it will be locked toward node 4 and the throttling connection packets will not interfere with the communication. This is also true because node 4 is in-directional-range with node 1. Hence 4 will not transmit

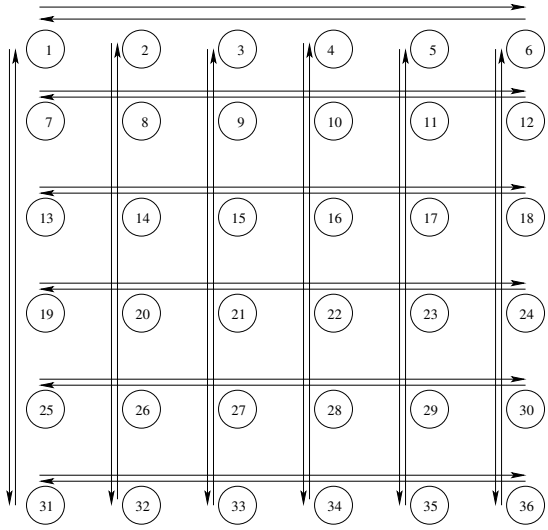


FIG. 7.7. Grid topology

when 1 is transmitting. Node 5 cannot hear to node 1 but it is within the interference distance of node 1. This interference will cause more packet drops at node 5, thus reducing the throughput. This happens for both the existing implementation and the proposed implementation

7.4.2 Grid topology:

We now present our results with the grid topology consisting of 36 nodes arranged in a 6×6 grid. Each node is 250 m apart from the vertical and horizontal neighbor. We present our results with a connection pattern as shown in the Figure 7.7. In this case, the routes are configured statically so that the packets always flow either in horizontal straight line or vertical straight line across the grid. This was done to force the packets to follow the predefined routes and to avoid taking other paths. The simulation consisted of 24 CBR connections as shown by arrows in Figure 7.7. Each connection runs from one end of the grid to another, either in horizontal or vertical direction. There are 12 sources, 6 sources in the first row and 6 sources in the first column of the grid. The nodes which are in the interior of the grid forward exactly 4

MAC Queue length	Average Jitter(in s)	IFQ Drops	Retry limit drops
Original	0.770165284	41633	1722
10	0.720823299	41175	1905
20	0.653387439	40828	1991
30	0.658129317	41148	1955

Table 7.2. Jitter and packet drops when MAC queue length is altered

CBR connections. Each connection has a sending interval selected randomly in the range of 20MS to 30MS. We keep the packet size constant, set to 1024 bytes.

The MAC Queue length is varied and results were analyzed. The packet sending interval is also varied and the performance was studied. The results are described in the below subsections.

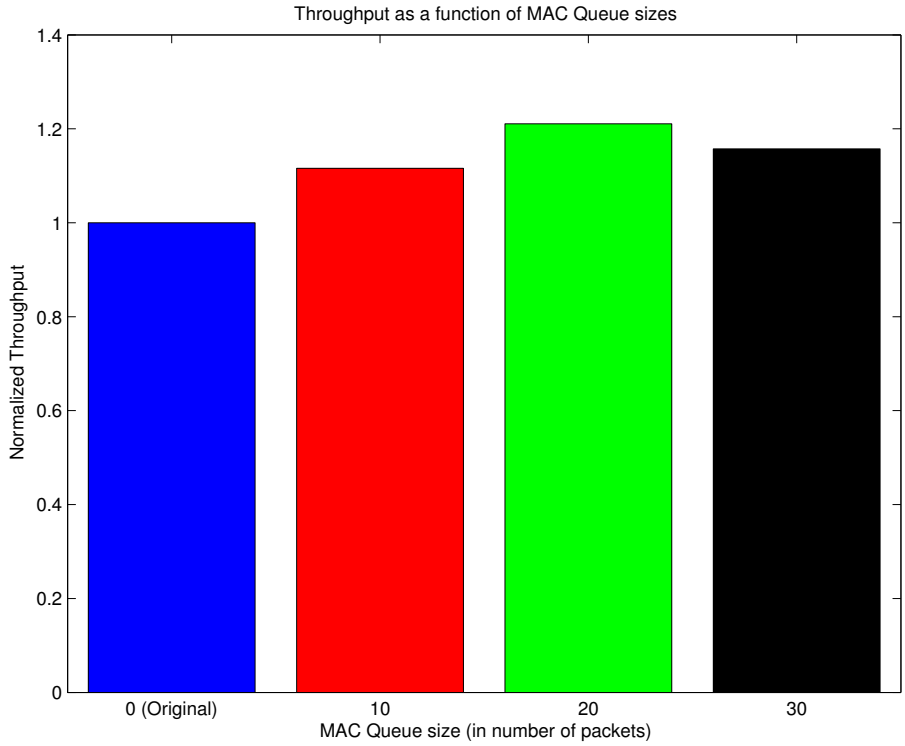


FIG. 7.8. Study of normalized throughput as MAC queue length is varied

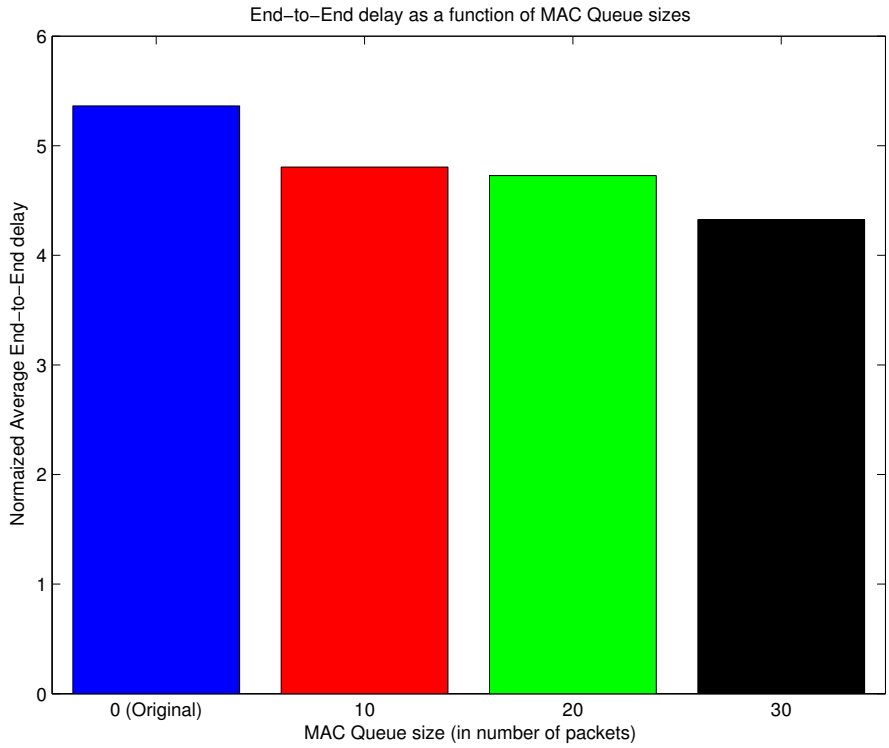


FIG. 7.9. Study of normalized average End-to-End delay as MAC queue length is varied

Effect of MAC Queue length The DMAC with proposed queuing mechanism which is tuned to directional antenna can be seen to outperform the existing protocol as shown in Figure 7.8. The improvement as much as 21% can be seen in the Figure 7.8. The proposed DMAC protocol has considerable lower end-to-end delay compared with the original protocol. From the Figure 7.8, one can see that there is as much as 20% reduction average end-to-end delay. We attribute it to the absence of head of line blocking and the reduced wait times while the packet is being sent.

The proposed solution also performs better in terms of average jitter as shown in Table 7.2. The packet drops because of the *Interlinking queue* or the IFQ drops are almost same as that of the original implementation. We gain marginally in the IFQ drops. The packets dropped due to the exceeded retransmit limit are more in

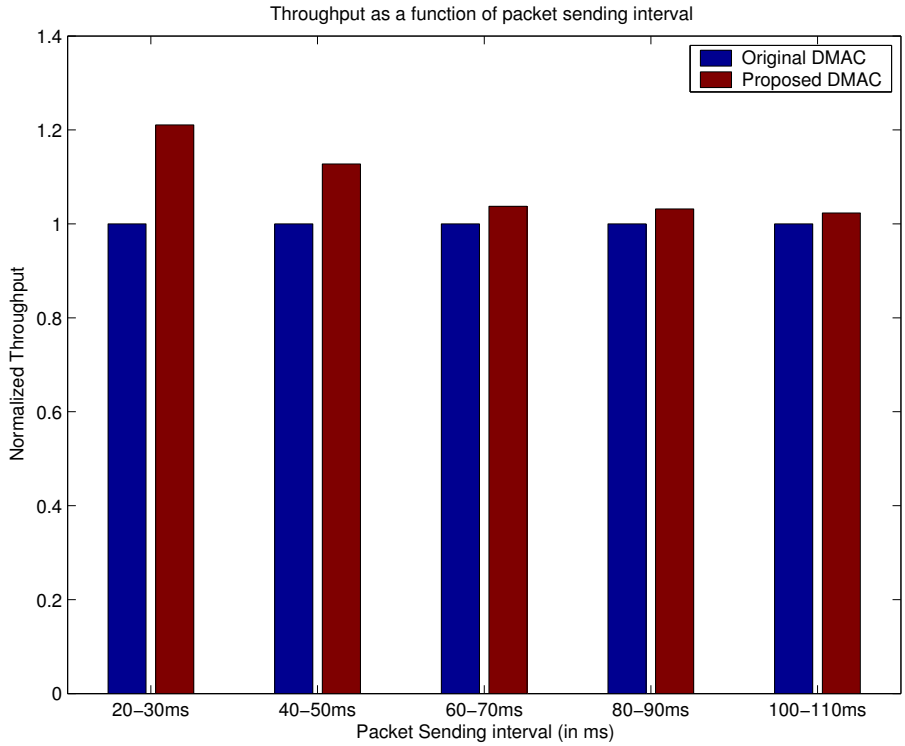


FIG. 7.10. Study of normalized throughput as sending interval is varied

our implementation than in the original implementation as shown in the Table 7.2. This is one of the parameters which we can improve upon. One of our future work is to analyze the root cause of this behavior and try to come up with a solution to solve it.

Effect of packet sending interval We now vary the sending interval and its effect on the throughput is shown in Figure 7.10. As the sending interval increases there is fewer *hotspot* created. Note that the effectiveness of the proposed queuing mechanism varies directly with the number of *hotspots* created. Tapping the channel reuse can be exploited only in such cases because of the ability to pick the right packet from the queue. Otherwise, the proposed implementation will perform as good as the original one. The best case improvement in throughput was around 21% higher than

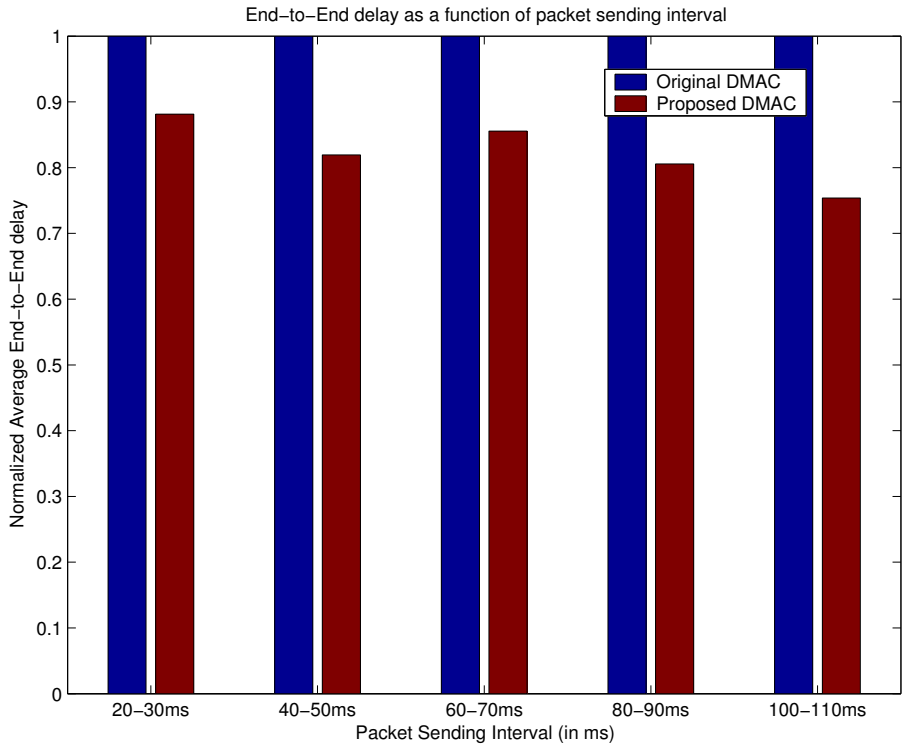


FIG. 7.11. Study of normalized average End-to-End delay as sending interval is varied

the original implementation and was observed in the case when sending interval was set to 20ms as shown in the graph 7.10.

The end-to-end delay is shown in the graph 7.11. Our protocol provides much lower end-to-end delay reductions. However, in this case, we did not observe any significant jitter improvements. In fact, in two out of the five cases, the jitter in the proposed implementation is higher. More analysis is needed to find out the cause of variation of this parameter. The IFQ drops has significantly reduced in the proposed protocol. When the sending interval is set to 100 ms, we get improvement as high as 40% in the IFQ drops. The graphs for IFQ drops are not shown due to space constraints.

As explained in the previous section, the packet drops due to exceeded retry

limit is higher in the proposed protocol. We would like to consider the analysis of this factor in our future work.

7.4.3 Grid topology with random connections:

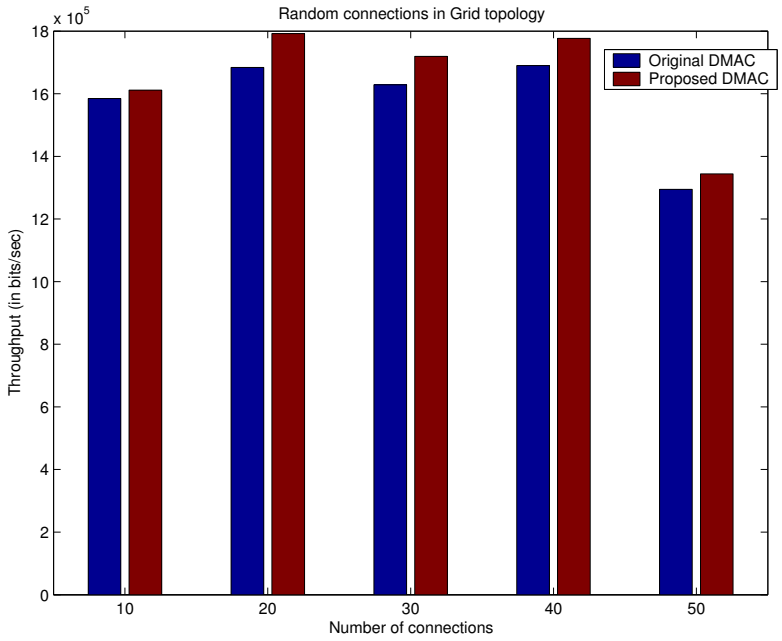


FIG. 7.12. Study of throughput as the number of random connections in Grid Topology is varied

Finally, we consider a grid topology with random connection pattern. The results reported is an average of five independent runs. The above grid topology is maintained as the same and the connection is chosen randomly. We keep the same setting of topology as in the previous case in terms of grid size. However, in this case we vary the the number of CBR connections from 10 to 50. The sending interval of the packets are chosen randomly from 20MS to 30MS. Figure 7.12 shows that the throughput is higher with the proposed implementation than in the original implementation.

In the individual cases, there was a single case where the throughput of the proposed implementation was 10% worse than the existing implementation. The best

improvement we got was around 66% higher than the existing implementation. The analysis for the throughput degradation for the 10% decrease was not done in detail because there was an improvement in most of the cases. However, we wish to analyze the case in near future which may throw some light on the behavior of the DMAC.

7.5 Future work

The design of the new protocol can be improved in many directions. Since the updates in DNAV are improper because of the deafness, we would like to study the performance of the protocol with some mechanism like Choudhury's [4] that reduces the deafness problem.

When an omni-directional packet is present in the Interlinking queue, the current design will block the packets that are present behind the omni-directional packet in the Interlinking queue. If such packets are not blocked, then since the omni-directional packet has the maximum wait time, it may happen that the omni-directional packet will starve. We would like to investigate the design of allowing the other packets to pass the omni-directional packet without letting the omni-directional packet to starve for a longer time.

The experiments in this section was done with a 8-sectored switched beam antenna. The effect of altering the number of sectors on the performance of the proposed protocol will be interesting. We would like to study the effect in future.

7.6 Conclusion

The directional antenna provides many useful characteristics at the physical layer that can be used to provide superior performance to omni-directional antennas. Existing higher layer protocols do not effectively use these features. One of the characteristics that can be exploited is the channel reuse. Under medium to heavy loads, the

channel is underutilized in omni-directional antenna. The existing directional MAC protocol also does not address this issue.

In this study we identify two problems with the existing directional MAC layer. First, we describe the Head of line Blocking problem due to FIFO queuing policy with the existing directional MAC protocol. Second, we examine the current inefficient DNAV updating mechanism. Based on these observations, we propose a new Directional MAC protocol that implements a greedy queuing policy and uses a modified D-NAV update mechanism. Our simulation results indicate that the proposed protocol outperforms the existing directional MAC protocol in almost all the cases by increasing the overall throughput and lowering average end-to-end delay.

Chapter 8

DIRECTIONAL ROUTING

The increased directional range of the signal facilitates to reach a larger distance by directional transmission. The current routing protocols will find only the omni-directional routes, ignoring the nodes that can be reached by directional transmission. This study proposes two approaches to take advantage of the higher range. Firstly, we propose a mechanism to notify the routing layer about the nodes that can be reached by directional transmission. Secondly, we propose a modified routing protocol which uses the directional range for shortening the number of hops in the source route.

8.1 Overview

The popular design hint advocates upper layers to exploit the full power of the lower level layers. In the paper [17], Lampson states exactly the same by his rule *"Don't hide power"*. He articulates *"When a low level of abstraction allows something to be done quickly, higher levels should not bury this power inside something more general"*. Is the *power* greater reachability of directional antenna used anywhere in the above layers other than the physical layer? Unfortunately the answer for this question is "no". The DMAC in the MAC layer seems to use some of its capabilities, but, as we discuss below, only in a limited manner.

Let us consider one of the primary advantages of directional antennas: the

"greater range" of the signal. This property is completely utilized in the current architecture. When the nodes are transmitting in omni-directional mode, the advantage of *greater range* is absent. Hence, they cannot reach all the nodes which can be reached by directional mode.

Let us define the terms directional neighbors and omni-directional neighbors. *Directional neighbors* are the nodes which can be reached from directional mode of transmission but not by omni-directional mode. Any node that can be reached with omni-directional transmission is called as *Omni-directional neighbor*. Consider the Figure 3.4. We see that node C is directly reachable from node A in directional mode but not in omni-directional mode. In the existing architecture, node A does not even try to reach C directly. Node B is a omni-directional neighbor of A but node C is a directional neighbor for node A.

Most of the routing layer protocols like AODV [19] and DSR [12], broadcasts the route request packets to find out the routes. All such requests use omni-directional mode of transmission for broadcasting. Hence directional neighbors are not discovered by using omni-directional route discovery. This is a major hindrance which makes the directional neighbors invisible to the routing layer. Hence, the next hop for a packet will always be a omni-directional neighbor. Even though the packet may be transmitted directionally, it is transmitted to a node who can be reached by omni-directional communication. This excludes the use of *greater range* of directionally focused signals.

The idea of sending directional beams in all sectors instead of a single omni-directional transmission was suggested by Korakis et al. in [16] which was used for sending RTS. Choudhury et al. [5] works on discovering the directional neighbors by sweeping the directional beam for all broadcast packets over each sector to transmit a omni-directional packet. This makes it possible to discover the directional neighbors.

The overhead involved in such an approach is very high. For an 8 sectored antenna, to transmit one omni-directional beam, there should be 8 directional transmissions, which is a significant onus. We believe that a lower cost approach will be more beneficial. Furthermore, using this process for route request floods is likely to be counter-productive due to the deafness that occurs as many nodes are concurrently performing sweeping broadcasts.

The MAC layer keeps a track of the directional neighbors in its *Angle of Arrival (AoA) cache* as explained in section 3.3. Whenever the node listens to any data packet being transmitted on the channel, it will mark the sender as a neighbor and add the entry in the AoA cache. This part of the thesis attempts to create a generic interface called as the *"Upcall interface"* from the DMAC layer to the routing layer to make use of the directional neighbors that are discovered. As a proof of concept, a sample routing protocol DSR is modified and the results are analyzed. A modified DSR protocol called *Directional DSR(DDSR)*. We also discuss the generalization of this idea to actively attempt to compact routes (by performing localized searches for directional neighbors) as a topic of future work towards the end of this chapter.

8.2 Proposed protocol

This section describes the relevant data structures and the algorithm used to implement the upcall interface and the design of the DDSR protocol.

8.2.1 Using AoA Cache to detect directional and omni-directional neighbors

DMAC stores a table known as AoA cache as explained in Section 3.3. This table contains the information about the nodes from which the transmission has been heard and the respective angle of arrival. If an entry for a node X exists in AoA cache

of node Y, then by the bi-directional nature of the channel we can conclude that node X can be reached by transmitting a directed beam in the recorded angle from node Y. Thus, we can say that node X is either a directional or omni-directional neighbor of node Y. Thus, the AoA cache stores a subset of the neighbors of a node, including some nodes that are learned from their own transmission and are not discoverable using omni-directional route discovery and neighbor discovery transmissions.

Since this data is readily available in the AoA cache, there is *no extra overhead* to detect these directional neighbors that are already in this list. This knowledge can be exploited at the MAC layer and possibly at the upper layers to tap the advantages of the directional transmission. There is already a mechanism to find the omni-directional neighbors in the routing layer. Unfortunately, no such mechanism exists to find the directional neighbors except for the high cost sweeping procedure of omni-directional packets described by Choudhury in [5]. By using the information in the cache, directional neighbors as well as omni-directional neighbors can also be found with zero overhead. Note that the AoA cache may not contain all the neighbors for the given node. Only the nodes from which a given node has heard a data packet will be stored in the AoA cache. We conjecture that there is no loss in using the information that is already present to our disposal.

In the existing protocol, the AoA cache adds or updates an entry if it listens to an RTS or a DATA packet. In the future work, we plan to modify the implementation to update AoA cache on all kinds of packets received by the MAC layer.

8.2.2 A Generic Interface to Expose MAC Information to routing layer

The number of hops is a critical factor to choose the routes. It is believed that the lesser the number of hops, the better is the route. This has been the de-

facto standard to select the route. Consider the Figure 3.4 in which node A has the directional neighbor C and omni-directional neighbor B. If A needs to send a packet to C, then the one-hop directional path as shown in 3.4(b) is better than the path A-B-C which is two hops. The existing routing layer is incapable of finding this one-hop route because of the mechanism in which it finds the routes. It searches the routes by broadcasting packets in a omni-directional nature. If this is the case, then the directional neighbors are not discovered during the handshake messages of the routing layer. The knowledge of directional neighbors is completely absent in the existing architecture of the routing layer.

If the routing layer can use the directional neighbors for finding the routes, a large improvement in throughput and end-to-end delay can be observed. The routing layer can get the data of neighbors that includes directional neighbors. This information can be used to restructure the existing routes or to create alternative shorter path routes. A generic interface from the DMAC layer called as "Upcall interface" to the routing layer is to be created to enable the transfer of neighbor information to the routing layer.

The upcall interface needs to be triggered to send two kinds of information:

1. **Addition of neighbor into AoA cache:** When an entry for a node is added to the AoA cache, the routing layer should be informed about the newly added neighbor.

2. **Purging of neighbor from AoA cache:** It may happen that the neighbor has moved out of range. In such cases, the AoA entry for that node is purged. If such information is not propagated up to the routing layer, then there will be an inconsistent image of the neighbors in the routing layer. To avoid this, there is another upcall performed during purging of an entry from AoA cache to the routing layer. The decision to purge an entry is described in section

subsec:aoaCache.

The routing layer may do the necessary alterations to its data structures based on this upcall. This is entirely dependent on the routing layer and has no connection with the generic interface. In the next section, we explain the design of DDSR, a modified DSR to take advantage of the above mentioned upcall.

8.2.3 Design of DDSR

The routing protocol should handle the upcalls from the MAC layer to utilize the *upcalls*. The DDSR protocol maintains a table which is updated according to the upcall information. This table is called "one-hop table". This table stores all the nodes that can reached by one hop. Effectively, this contains the neighbors of the given node. When the upcall function is called by the MAC layer when a node was added, then the node was added to the one-hop table. If the upcall for purging of the node then the entry is purged from the one-hop table too.

Since the DSR works on source routing, the packet contains the complete route which the packet needs to take. While the packet is to be forwarded, the source route is searched to select the next hop that is reachable from directional transmission and which is nearest to the source in the source route. The one-hop table will aid the search process by revealing if a given node in the source route is a directional neighbor. This makes it possible for the packet in DDSR to take a shorter route than the one specified in its source route if a directional neighbor can be reached in one hop instead of two hops.

8.3 Implementation

This section describes the relevant data structures and the algorithm used to implement the upcall interface and the design of the DDSR protocol which were

explained in the Section 8.2.

8.3.1 Upcall interface:

In the proposed protocol, the routing layer needs to register a single simple function that will handle the events from MAC layer about neighborhood information. This function should have the following signature:

```
void upcallFromMACLayer(Node neighborNode, BOOL wasAdded);
```

where $neighborNode$ is the neighbor node that is to be added or purged. This can be an IP address of the neighbor node. The $wasAdded$ is a boolean value that is $true$ if the $neighborNode$ was added to cache and $false$ if it was purged from the cache. The MAC layer will call this function with appropriate parameters when any node is added to or purged from the AoA cache. It is to be noted that there AoA will be updated every time a signal is heard from the neighbor node. There may be changes in the angle of arrival for a given neighbor, in which case an AoA update takes place. All these updates need not be known to the upcall. It is enough if addition and purging is exposed to the upper layer. This reduces the number of calls made to the routing layer which inturn reduces the processing to be done in the routing layer.

8.3.2 Directional DSR

As the proof of concept and to show the results for the above idea, the DSR routing protocol was modified to accept the upcalls from the MAC layer and to change its routing structures accordingly. This new protocol is called "Directional DSR" or DDSR. A interface was registered with MAC layer with the given signature of the upcall interface.

One-hop table: In the proposed implementation, one-hop table is maintained as a link list sorted according to the neighbor node index. Adding and deleting an entry from the one-hop table is a simple procedure to manipulate this link list.

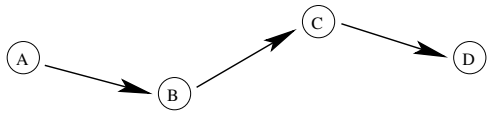


FIG. 8.1. Routing in DSR

Searching for one hop route: The DSR protocol includes the complete route path (the sequence of nodes through which the packet should be routed to reach the destination) in the *Source Route* field in the packet header. For example, in the Figure 8.1, if A needs to send a packet to node D, then it will route it through the nodes B,C and finally to D. Hence the *Source Route* header field will contain the path $\langle B, C \rangle$. Routes are discovered during the route discovery phase and are cached in a table called *Route Cache*. When there is a packet generated by the source node for a given destination, the route cache is checked to see if there is a path to the destination. If one or more routes exist, then the full path of the shortest route is inserted in the *Source Route* header field. If there is no route to the destination, then the route discovery phase is executed which finds the path to the destination. This discovery phase finds the route by flooding using omni-directional broadcasts. Hence only hops that can be reached by omni-directional transmission are found; strictly directional neighbors will not be used in routes. The route discovery phase is not given in detail in this thesis. Interested readers may refer to Johnson's paper [12] for the details of operation of DSR protocol.

The field $segsLeft$ in the header of DSR protocol will indicate the number of segments left to reach the destination. Each node will decrement the $segsLeft$ field

before sending the packet to the next hop. In the above case, since there are 3 hops to reach the destination, node A will have the value of 2 for $segsLeft$. This is because when node B receives the packet, it will see that there are 2 more segments left. Node B then decrements the $segsLeft$ field to 1 and sends it to node C. Node C will have the value 1 when it receives the packet and will send the packet with $segsLeft$ value as 0 before transmitting it to the destination D.

We modify the DSR protocol to reduce the number the hops by skipping the hops if a directional neighbor is found in the source route. In Figure 8.1, node A has discovered only node B to be the neighbor. If node C is discovered as the directional neighbor by the DMAC layer, then the entry to node C will be present in the one-hop table of node A. If such an entry exists, then node A can send the packet directly to node C instead of sending it to node B and node B sending the same packet to node C. This reduces the number of hops.

To optimize further the number of hops, the DDSR protocol tries to send the packet to the hop that is nearest to the destination and which is present in the source route. The algorithm followed to select the next hop in the DDSR protocol is given in Algorithm 8.3.2. It is to be mentioned that this algorithm is executed before decrementing the $segsLeft$ field. Hence, $segsLeft$ field will still contain the value which was present when the packet was received. For example, when node B processes this packet the $segsLeft$ will have a value of 2 and not 1. After the algorithm is executed the $segsLeft$ is altered accordingly to indicate the actual number of segments left. For example, if node A finds that it can reach to node C directly then the $segsLeft$ field is set to 1.

DDSR route maintenance: Consider the case when the directional node is not reachable. In such a case, we have to update the one-hop table. This is automatically taken care by the upcall from the MAC layer. Consider the scenario in

Algorithm 2 Algorithm to find the best next hop in DDSR protocol

```
bestNextHop = NULL {The value of bestNextHop will be returned from this algorithm}
if destination is present in one hop table then
  {It can be directly sent to the destination. Select destination as the next hop}
  bestNextHop = destination
else
  {Parse the source route from right to left to find next hop nearest to the destination}
  numNodes = Number of elements in source route
  for i = numNodes backto (numNodes − segsLeft + 2) do
    node = Node at i − th position in the source route
    if node is present in one hop table then
      bestNextHop = node
      break the loop
    end if
  end for
end if
return bestNextHop
```

Figure 8.1 when node C moves out of directional range from node A. When node A has not yet detected this movement, it will try to route the packet to node C rather than node B. Since, node C cannot be reached, the MAC layer will try to transmit it till the retry limit is exceeded and then the packet is discarded and the routing layer is informed about the drop of the packet. As explained in section 3.3.1, the entry in the AoA cache of node A for node C is also purged. When this happens, the upcall is invoked informing the routing layer to remove the entry to node C. Since, node C will no more be present in the one-hop table, node A will not further try to send the packet to node C directly. It will switch to normal mode of sending packet to node B. Hence, the upcall will control the route maintainance mechanism when the directional neighbor moves out of range.

But when the route error occurs, the path is purged from the route cache of the DDSR in the same way as it happens in DSR. It may happen that the error is due

Parameter	Value
Omni-directional range	250m
Directional range	450m
Directional antenna model	Switched beam
Mobility	none
Propagation Channel Frequency	9.14 * 10^8 Hz
Path loss Model	Two Ray
Transmission power	24.5 dBm
Receiver sensitivity	-68.1 dBm
Directional gain	10.0 dB
Antenna Model	Switched Beam
Directional NAV Delta Angle	22.5 degrees

Table 8.1. Simulation Parameters for directional routing

to directional transmission by skipping the hop. The route cache is unaware of the one-hop table and cannot distinguish between a route error because of skipping hops or a generic route error. As a part of future work, we would like to implement route repair/maintainance mechanism which is tuned to DDSR protocol.

8.4 Performance Evaluation

To verify the effectiveness of the upcall interface and the DDSR protocol, Qualnet simulator [20] is used. The table 8.1 lists some of the important simulation parameters.

8.4.1 Unfairness in retry limit:

In the directional mode, each node will send a directional beam to transmit a packet if the next hop node is present in the AoA cache. If the RTS does not get a CTS reply, then the packet is retransmitted. This retransmission goes on for a specified number of times called as the *Directional retransmit limit*. If the number of retransmissions exceed the *directional retransmit limit*, then the packet is transmitted

in the omni-directional mode till the *retransmit limit* is reached. If the RTS still does not get any reply, then it is dropped and the routing layer is informed about the drop. The expression (*Directional Retransmit limit* $<$ *Retransmit limit*) is satisfied for most of the cases. Usually the value of *Directional Retransmit limit* is 4 and the value of *Retransmit limit* is 7. Hence the packet is tried to be transmitted for 4 times in directional mode if the next hop is present in the AoA cache. The node tries for another 3 times in omni-directional mode if the attempt is unsuccessful.

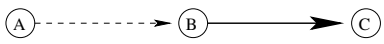


FIG. 8.2. Unfairness in retry limit

The objective of trying to send in omni-directional mode after *directional retransmit limit*, is to enable to find the next hop if it has moved out of the sector in which the AoA was recorded. If such a policy is followed in DDSR, there will be unfairness for the DDSR protocol over the normal DSR protocol. Consider the Figure 8.2 where node A is trying to send to node B while node B is already sending a packet to node C. Node B will be deaf towards node A because of this reason. In case of DMAC using normal DSR routing layer, the packet is retransmitted totally for retry limit times from node A and then if it is not able to reach node B, it is dropped. We are trying to use directional routes in DDSR protocol. It is already known that the directional route is the one that will be found by the algorithm 8.3.2. If none of the directional next hop is found then the default route as specified by the source route is followed. If the above mentioned retry limits are used, then if a node finds a directional neighbor in the source route which is nearer to the destination, then it will try to transmit the packet to that node. If the next hop node is deaf towards the sending node, then there will be RTS drop inevitably. The sender tries to transmit the packet for *Directional retransmit limit* times. Once this limit

is crossed, it tries to transmit in omni-directional mode till *Retransmit limit* number of times is reached. This is unfair to DDSR because a directional neighbor has been found as the next hop and it is known that it cannot be reached in omni-directional mode. Hence, the attempts to send the packet after the *Directional retransmit limit* but before *Retransmit limit* are futile. Effectively, the DDSR gets lesser chance to retransmit the packet before it is dropped and the routing layer is informed about the drop. To overcome this unfairness, we set the value of *Directional retransmit limit* same as the *Retransmit limit*. The results in this section should not be affected because we are considering only static nodes which do not move. Hence there is no need to search in omni-directional mode when the node is unreachable for *Directional retransmit limit* times in the directional mode.

Chain topology was simulated using DDSR protocol. We wish to simulate other kind of topologies in the future. The following sections gives the results obtained.

8.4.2 Chain topology:

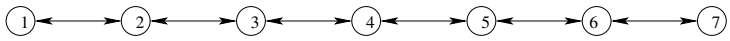


FIG. 8.3. Chain topology

Nodes were set up in chain topology as show in Figure 8.3. Two CBR connections were made to run. One connection from 1-7 and another from 7-1. Each node was 150 m away from its adjacent node. This distance was chosen because the node will be reachable using directional transmission and omni-reception for a distance upto 340m. However, for directional transmission and directional reception, the range is 450m as specified in the table 8.1. The reason for choosing two connections going in opposite direction was because of the nature of AoA cache updates in the DMAC layer. In the current implementation, the AoA cache adds/updates a node entry only when

it listens to the RTS or the DATA packet as explained in section 8.2.1. We would like to change this update policy in future and measure the results with even a single connection. Hence, if a single connection flows, every node would know the directional neighbors in the direction in which the packet reached the node. For finding the best next hop routes, we need directional neighbors in the opposite direction. In the above case, if a single connection from 1-7 was simulated, then consider the AoA cache of node 4. Node 4 would have heard RTS and DATA of node 2 and node 3 but not of nodes 5 and 6. Node 4 will know that node 2 is a directional neighbor but would not know that node 6 is a directional neighbor. Hence, when it receives a packet destined to 7 with source route $\langle 2, 3, 4, 5, 6 \rangle$, it does not know that it can reach node 6 with a single directional transmission. Thus, setting a single connection cannot prove the effectiveness of the DDSR protocol.

The sending rate of CBR connection was varied and throughput, end-to-end delay and number of packets dropped because of exceeded retransmit limit was measured. Very high rates would yield a good throughput improvement. End-to-end delay improvements were found consistently. The number of retransmits are very high in the proposed protocol. The explanation is given below:

Average End to End delay: As seen from the Figure 8.3, the packets would take 6 hops to reach from node 1 to node 7 when the simulation was run with normal DSR. The reduced number of hops in the proposed DDSR protocol allows to reach destination in 3 to 4 hops. The number of hops being lesser, the packets will now take lesser time to reach the destination. There was a consistent improvement in the average End-to-End delay as shown in the Figure 8.4. Improvement as high as 99% was observed. On an average, the improvement was around 50%.

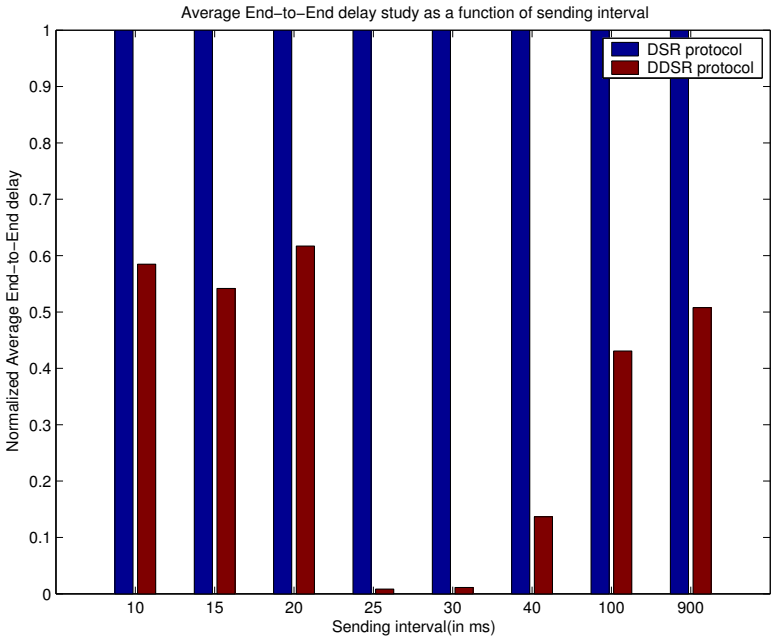


FIG. 8.4. Study of Average End-to-End delay when sending interval is varied

Overall Throughput: When the sending rate is very high(say, 1 packet every 10ms), reducing the number of hops not only decreases the end-to-end delay, but also increases the throughput. There is a very high contention for the channel and if the number of nodes fighting for channel acquisition is lesser, there is a better chance of winning the channel. Figure 8.5. If the sending rates are very low(say, 1 packet every 500ms), then the throughput improvement will not be found. This is because of the time gap between the packets. By the time the next packet is generated, there is sufficient time to send the previous packet between 6 hops. So reducing the number of hops does not increase the throughput even though the number of hops are reduced.

There is one case (20ms sending interval) when the throughput while using DDSR is almost half the throughput using DSR. The packet retransmits in this case is seen to be very high when compared to the DSR counterpart. More study needs to be done to analyze this case.

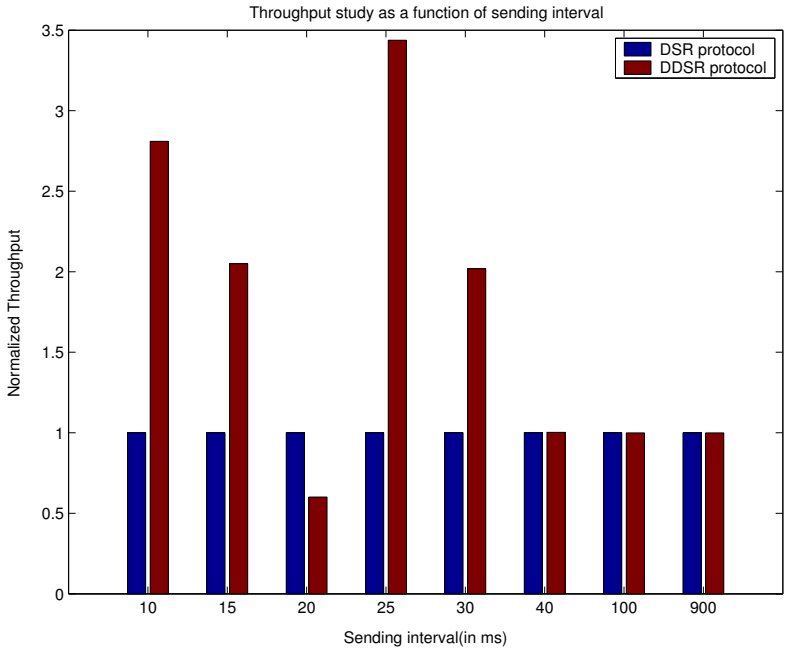


FIG. 8.5. Study of Throughput when sending interval is varied

Number of packets dropped due to exceeded retransmit limit: There is an highly increased packet drop due to exceeded retransmit limit in the proposed DDSR protocol when compared to the normal DSR during very low sending rates. This happens even though we have set the *Directional Retransmit Limit* = *Retransmit limit*. This is because of the lower route repair/maintainance mechanism in DDSR. The DDSR route repair and maintainance mechanism is same as that of the old DSR protocol. Its unaware of the one-hop table and more work needs to be done in this area to improve the performance of DDSR under high loads and constant route breaks. Figure 8.6 shows the degradation in packet drops because of exceeded retransmit limit.

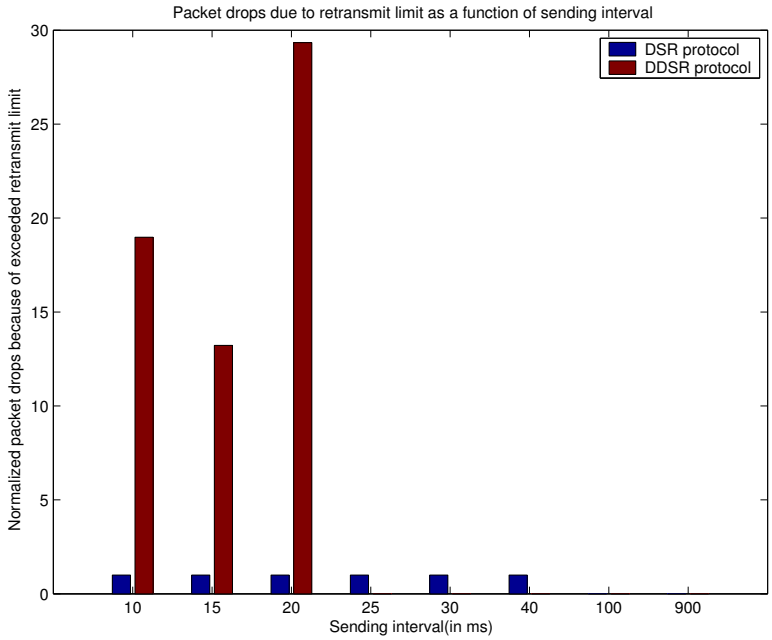


FIG. 8.6. Study of packet drops when sending interval is varied

8.4.3 Grid topology:

A 6×6 grid as shown in Figure 7.7 was simulated. Each node is 150m apart from its adjacent horizontal and vertical node. 12 connections run from one end of the grid to another as shown in the Figure 7.7. The sending interval for each connection is varied and the effect was observed. It was seen that the loss in throughput is significant when a lot of route error messages are generated by the DDSR. This makes the need for a coherent route error handling in DDSR more important. It was seen that in case of very high route error messages, the end-to-end delay also is very high in DDSR. We conclude the simulations in this topology by arguing that without proper route repair mechanism in place, its hard to get the advantages of the reduced number of hops of DDSR.

From the graphs 8.7, we can see that the average End-to-End delay is better in DDSR protocol except for one case. When the sending rate of connections is set

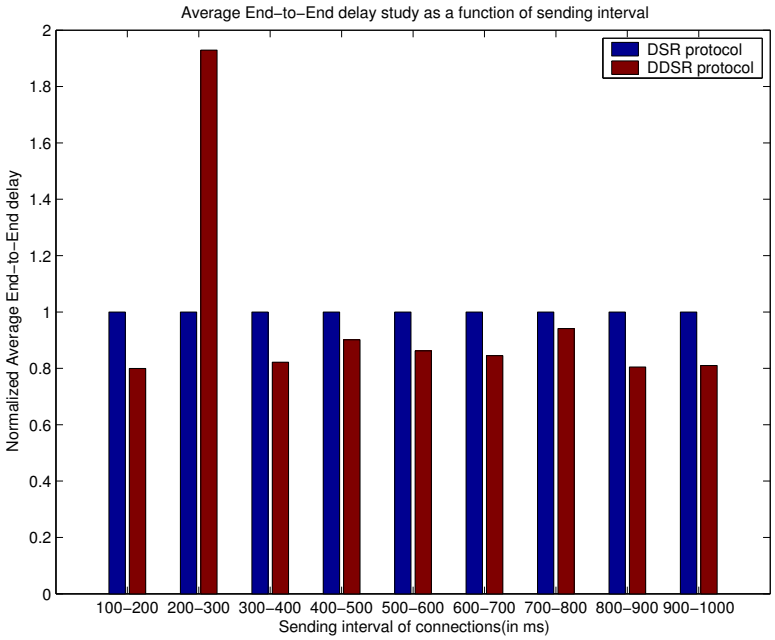


FIG. 8.7. Study of Average End-to-End delay when sending interval of the connections are varied

between 200ms to 300ms, we see that the end to end delay in DDSR is the double that of DSR value. This effect can be explained by figure 8.9. DDSR will give an improvement in the Average End-to-End delay even if the route error messages are high in DDSR as shown in figure 8.9. If the number of route error messages is very high, then we can see the degradation in Average End-to-End delay. We can see that when sending rate is at 200ms to 300ms, the number of route error messages generated are around 35 times higher. We expect that after the route repair mechanism in DDSR is in place, there will be a better improvement in throughput and end-to-end delay.

8.4.4 Grid scenario with lesser number of connections:

The scenario described in 8.4.3 is very dense. There are 12 connections running through the grid with each node taking handling 4 connections. The number of route errors generated in such cases will be irrepressible. Hence a sparse grid topology as

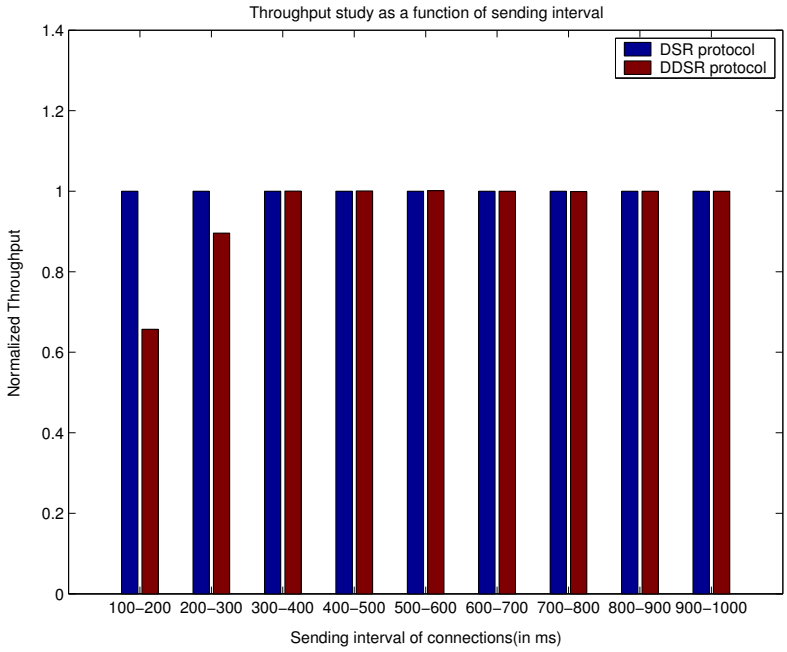


FIG. 8.8. Study of Throughput when sending interval of the connections are varied

shown in Figure 8.10 was simulated. The number of route error messages for the same connection sending intervals are lesser in this scene. Significant improvement in the end-to-end delay can be observed in Figure 8.11 even though the number of route error messages are large as shown in Figure 8.13. The throughput will remain almost the same as shown in Figure 8.12 because of the sparse network.

8.5 Future Work

This study can be further improved in many directions. We would like to implement the update policy for AoA as our first step. Instead of updating AoA for just RTS and DATA packets, it would be more sensible to update it on every packet heard. We believe that this will be more beneficial.

The DDSR protocol now is in a very immature stage. The route error and route maintainance mechanism which is very important for a contended channel is

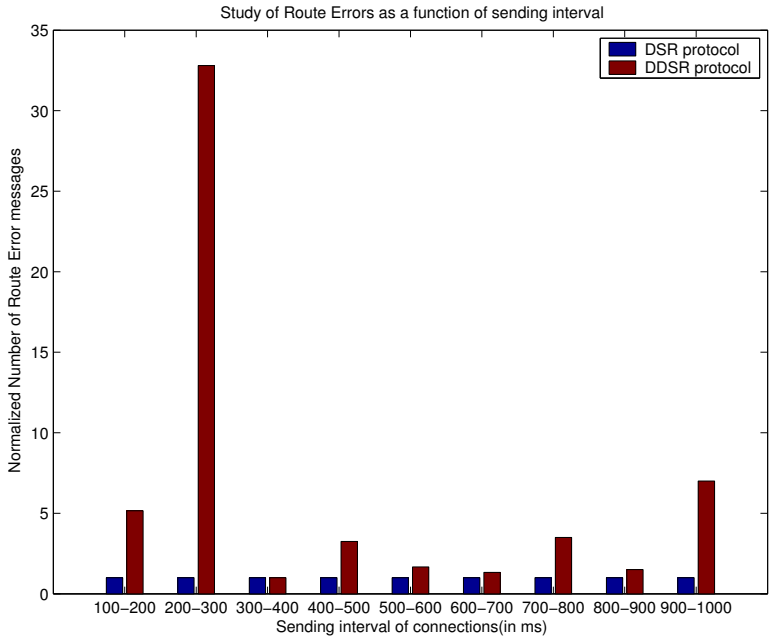


FIG. 8.9. Study of Route error messages when sending interval of the connections are varied

not aware of the one-hop table. We would like to add the route repair/maintenance mechanism which is completely tuned to DDSR as a part of future work.

After the repair mechanism in place, scenarios with mobility can be studied. This would give a very good analysis of the DDSR protocol stability.

In the current DMAC layer, the AoA cache contains only nodes that have transmitted the packet towards the given node. Nodes that are inactive are not included in the AoA cache. This makes AoA cache contain only a subset of neighbors. If a low cost neighbor discovery mechanism can be implemented in the DMAC layer and use these neighbors in routing layer, then better routes can be found.

The current DDSR protocol only tries to skip the hops in a given *Source route*. The protocol would be more coherent, if we can search for nodes that are not in the source route, but can be used to reach destination with lesser number of hops and delay.

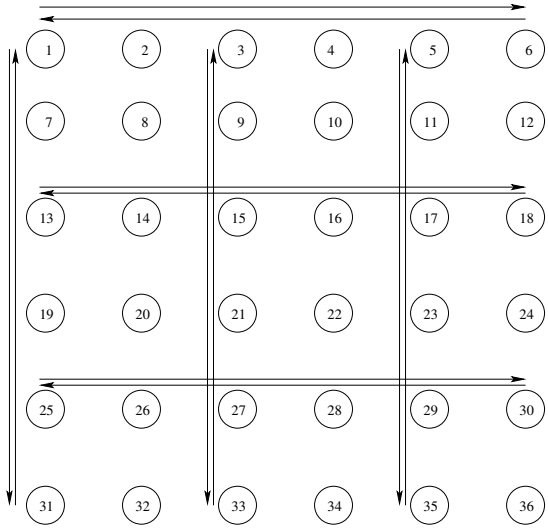


FIG. 8.10. Sparse Grid

8.6 Conclusion

From the simulation studies, we can conclude that the need for the routing layer to recognize the directional neighbors can lead to a great advantage in terms of end-to-end delay. The current implementations hide the power of the higher range in directional antennas even though some information is present in the MAC layer. By exposing this information to the routing layer, the channel can be made used in a better manner. The number of hops will be reduce and throughput improvement can be seen for lower sending rate connections. The DDSR protocol was shown to reap some of the benefits in spite of not tuning the protocol for route repair mechanism. We believe that with the implementation of ideas in the future work, the DDSR protocol will provide great gains for directional antennas.

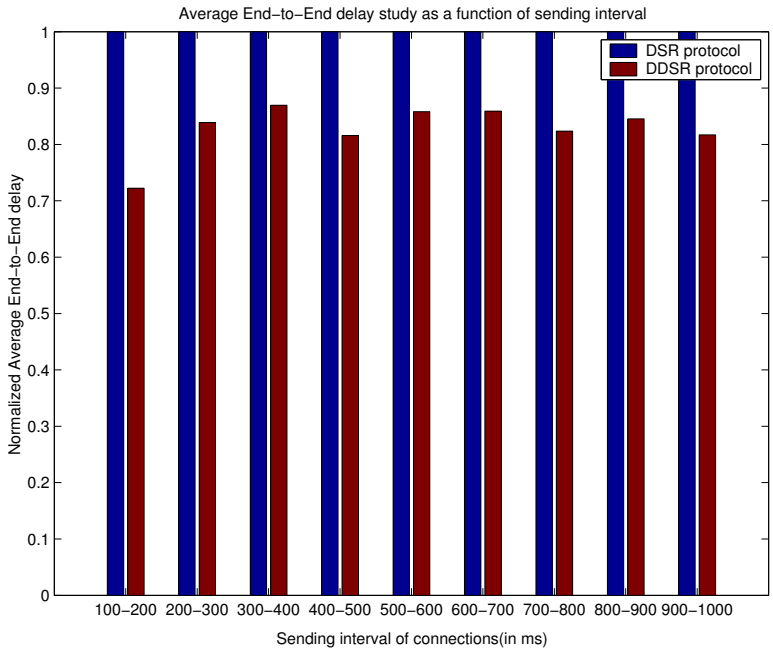


FIG. 8.11. Study of Average End-to-End delay in a sparse grid when sending interval of the connections are varied

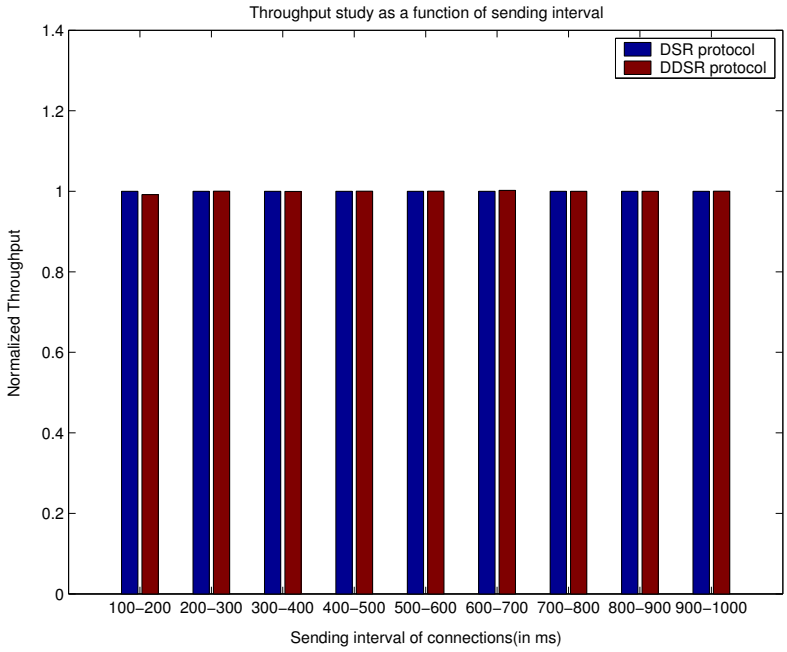


FIG. 8.12. Study of Throughput in a sparse grid when sending interval of the connections are varied

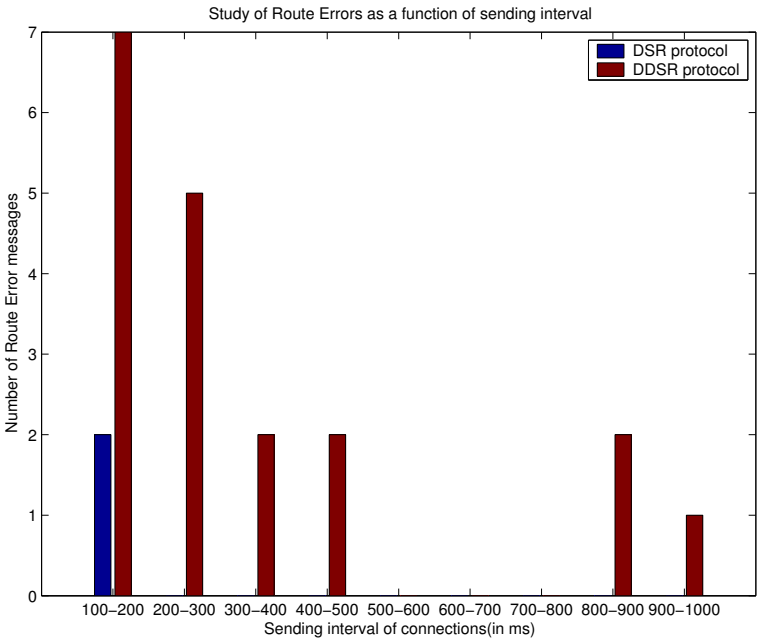


FIG. 8.13. Study of Route error messages in a sparse grid when sending interval of the connections are varied

Chapter 9

CONCLUSIONS

The interaction of DMAC layer and the chain connection was studied in the Chapter 5 and several aspects which are different from the 802.11 were observed in DMAC. Deafness is one of the major problem which affects the performance of the DMAC. It was found that significant packet losses are because of deafness which lead to longer backoff intervals causing NRTEs and route errors. The hidden terminal problem, which was absent in the 802.11, recurs in the DMAC because of deafness. Deafness caused inappropriate virtual carrier sensing because of inconsistent DNAV updates.

The powerful features of directional antennas were not fully utilized by DMAC. The longer range of directional antennas was suppressed by the routing layer which discovered only omni-directional routes. We later proposed a scheme in Chapter 8 to make use of the information stored at DMAC layer about some directional neighbors for effective routing. The channel reuse was suppressed by the *Head of Line blocking*. A scheme to subdue this blocking was suggested in Chapter 7.

The DMAC performance was also found to be an intricate function of the geometry of the topology. Several kinds of drops which did not occur in linear chain was observed when the chain was twisted. As observed in omni-directional antennas, it was found in the analysis that a well-behaved source in chain topologies will help to

improve the throughput of the channel. A greedy source, trying to push too many packets may result in damaging the performance of the network.

The Chapter 7 identifies the low channel utilization by DMAC because of the "Head of Line" blocking. A greedy queuing policy was proposed to solve this problem. Results indicated that under medium to heavy loads, the HoL blocking will offer an improvement in end-to-end delays and throughput. The chapter also identified the incorrect AoA updates in the cache and proposed a scheme to overcome the such updates. Deafness caused inconsistent DNAV updates and the proposed protocol proved effective even by using hints from such an inconsistent DNAV to schedule the packets. We expect a greater improvement in the protocol's performance when a accurate DNAV was used.

Routing in DMAC does not try to benefit from the longer range of the directional signals. Chapter 8 identifies that the information at MAC layer can be used at routing layer for utilizing this longer range. It proposes a scheme in which the routing layer can use the knowledge present at the DMAC layer about directional neighbors. A sample routing protocol was proposed to shorten the number of hops in a source routing protocol DSR. The initial results were positive but the sample protocol is still in a very premature stage. Many functions like route maintenance and route error handling should be incorporated for a complete working directional routing protocol.

REFERENCES

[1] Asis Nasipuri, Jothsna Mandava, H. M., and Hiromoto, R. E. On-Demand Routing Using Directional Antennas in Mobile Ad Hoc Networks. In *IEEE Wireless Communications and Networking Conference (WCNC)* (2000).

[2] Bandyopadhyay, S., Hasuike, K., Horisawa, S., and Tawara, S. An adaptive MAC and directional routing protocol for ad hoc wireless network using espar antenna. In *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing* (2001), ACM Press, pp. 243–246.

[3] Bharghavan, V., Demers, A., Shenker, S., and Zhang, L. MACAW: A Media Access Protocol for Wireless LAN's. In *ACM SIGCOMM 1994* (London, UK, August 31–September 2 1994), pp. 212–225.

[4] Choudhury, R. R., and Vaidya, N. H. Deafness: A Problem in Ad Hoc Networks when using Directional Antennas. In *University of Illinois at Urbana-Champaign Technical report* (2003).

[5] Choudhury, R. R., and Vaidya, N. H. Impact of Directional Antennas on Ad Hoc Networks Routing. In *Personal and Wireless Communication (PWC)* (2003).

[6] Choudhury, R. R., Yang, X., Vaidya, N. H., and Ramanathan, R. Using directional antennas for medium access control in ad hoc networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking* (2002), ACM Press, pp. 59–70.

[7] Chryssomallis, M. Smart antennas. In *IEEE Antennas and Propagation Magazine, Vol.42, No 3* (2003), pp. 129–136.

[8] Gain of directional antennas.
http://www.wj.com/pdf/technotes/Direct_antennas.pdf.

[9] Guney, K., and Akdagli, A. Null steering of linear antenna arrays using a modified tabu search algorithm. In *Journal of Electromagnetic Waves and Applications, Volume 15, No. 7* (2001), pp. 915–916.

[10] Huang, Z., Shen, C.-C., Srisathapornphat, C., and Jaikaeo, C. A MAC Protocol Based on Directional Antenna and Busy-tone for Ad Hoc Networks. In *IEEE MILCOM* (2002).

[11] IEEE standard definitions of terms for antennas. IEEE std 145-1993.

[12] Johnson, D. B., Maltz, D. A., Hu, Y.-C., and Jetcheva, J. G. The dynamic source routing protocol for mobile ad hoc networks (DSR), February 2002.

[13] Karn, P. MACA - A New Channel Access Method for Packet Radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference* (London, Ontario, Canada, September 22 1990), pp. 134–140.

[14] Ko, C. A fast null steering algorithm for linearly constrained adaptive arrays. In *IEEE Trans. Antennas and Propagation, U.S.A, Vol. AP-39* (U.S.A, 1991), pp. 1098–1104.

[15] Ko, Y.-B., Shankarkumar, V., and Vaidya, N. H. Medium access control protocols using directional antennas in ad hoc networks. In *Proceedings of the 2000 IEEE Computer and Communications Societies Conference on Computer Communications (INFOCOM-00)* (2000).

[16] Korakis, T., Jakllari, G., and Tassiulas, L. A mac protocol for full exploitation of directional antennas in ad-hoc wireless networks. In *Proceedings of*

the 4th ACM international symposium on Mobile ad hoc networking & computing (2003), ACM Press, pp. 98–107.

[17] Lampson, B. Hints for computer system design. *IEEE Software 1* (Oct. 1984), 11–29.

[18] Nasipuri, A., Ye, S., You, J., and Hiromoto, R. A MAC protocol for mobile ad hoc networks using directional antennas. In *WCNC* (2000).

[19] Perkins, C. E., Belding-Royer, E. M., and Das, S. R. Ad hoc on-demand distance vector (AODV) routing, January 2002.

[20] Qualnet network simulator, version 3.6. http://www.scalable-networks.com/.

[21] Ramanathan, R., and Redi, J. A brief overview of ad hoc networks: Challenges and directions. *IEEE Communications Magazine* (May 2002), 20–22.

[22] Royer, E., and Chai-Keong, O. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications 6*, 2 (Apr 1999), 46–55.

[23] Smart antenna systems. http://www.iec.org/online/tutorials/smart_ant/.

[24] Takai, M., Martin, J., Bagrodia, R., and Ren, A. Directional virtual carrier sensing for directional antennas in mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing* (2002), ACM Press, pp. 183–193.

[25] Wireless LAN antenna terminology. http://www.wlanantennas.com/antenna_terminology.htm.

[26] Xu, S., and Saadawi, T. Revealing the problems with 802.11 medium access control protocol in multi-hop wireless ad hoc networks. *Computer Networks 38*, 4 (March 2002), 531–548.