# On exploiting spatial reuse in wireless ad hoc networks

Ziguang Yan
University of Wollongong

# On Exploiting Spatial Reuse in Wireless Ad Hoc Networks

**A thesis submitted in partial fulfillment of the requirements for the award of the degree**

**Master of Engineering by Research**

**From**

**UNIVERSITY OF WOLLONGONG**

**By**

**Ziguang Yan**

**School of Electrical, Computer and Telecommunications Engineering**

**March 2008**

# Statement of Originality

I, Ziguang Yan, declare that this thesis, submitted in partial fulfillment of the requirements for the award of Master of Engineering - Research, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

Ziguang Yan

March 2008

# Abstract

Wireless ad hoc networks have been increasingly popular in recent years with the development of mobile devices. However, both theoretical and simulation works show that the capacity of wireless ad hoc networks is bounded due to its nature of *distributed* and *multihop*. Spatial reuse is a promising technology to increase the capacity of wireless ad hoc networks by allowing more transmissions to occur simultaneously. In this thesis, we enhance 802.11 performances by exploiting the benefits of spatial reuse in wireless ad hoc networks which is achieved by *transmission power control* (TPC) and *directional antennas*.

We first propose *spatial TPC* based on basic TPC to fully exploit the benefits of spatial reuse achieved by transmission range control. Simulation results show that spatial TPC achieves higher throughput and lower power consumption compared to 802.11 and basic TPC. We also develop four schemes of directional MAC protocols with the intention of overcoming the new hidden node problem faced by directional antennas. By extensive simulations under different topologies and traffic patterns, we find the directional *RTS/CTS* (DD) scheme outperforms 802.11 as well as other three schemes by fully exploiting the benefits of spatial reuse achieved by directional antennas.



**Keywords:**

Wireless ad hoc networks, MAC, CSMA, 802.11 DCF, Spatial reuse, Power control, Directional antennas, OPNET simulation.

# Table of Contents

## List of Figures

## List of tables

## Abbreviations

| | |
|---|---|
| ACK | acknowledgement |
| AoA | angle of arrival |
| AP | access point |
| ATIM | announcement traffic indication message |
| BEB | binary exponential backoff |
| BER | bit error rate |
| BSS | basic service set |
| CCA | clear channel assessment |
| CFP | contention free period |
| CP | contention period |
| CS | carrier sense |
| CSMA | carrier sense multiple access |
| CTS | clear to send |
| CW | contention window |
| DCF | distributed coordination function |
| DIFS | distributed (coordination function) interframe space |
| DPSK | differential phase shift key |
| DMAC | basic directional MAC protocol |
| DNAV | directional network allocation vector |
| EIFS | extended interframe space |
| FSM | finite state machine |
| GPS | global positioning system |
| IFS | interframe space |
| LoS | line of sight |

| | |
|---|---|
| MAC | medium access control |
| MSDU | MAC service data unit |
| NAV | network allocation vector |
| PCF | point coordination function |
| PCS | physical carrier sensing |
| PLCP | physical layer convergence protocol |
| PS | power saving |
| RTS | request to send |
| RTT | round trip time |
| SIFS | shortest interframe space |
| SISO | single in single out |
| SNR | signal to noise ratio |
| STA | station |
| TPC | transmission power control |
| ToA | time of arrival |
| VCS | virtual carrier sensing |
| WLAN | wireless local area network |
| WM | wireless medium |

## Acknowledgement

I would like to express my deepest gratitude to my supervisor Dr. Raad Raad for his patience and guidance during my research. Without his support and valuable suggestions, none of this would have been possible.

I also would like to thank Darryn Lowe and Dr. Kwan-wu Chin for their working enthusiasms and kind support.

I want to thank my parents, who have been giving me spiritual and financial support to make this degree possible.

Finally I want to thank my friends and guys in TITR lab for their kindness, humor and support.

# Chapter 1: Introduction

Wireless local area networks (WLANs) have become increasingly popular in recent years with the development of mobile devices such as laptops, cell phones and PDAs (Personal Digital Assistant). WLANs have many advantages over traditional wired LANs. One such advantage for WLAN is the convenient access and mobility support. Another advantage is that it is easy to deploy and reduces the cost of network setup. WLANs are usually used in airports, universities, hotels or cafes which can be backhauled to wired network. Reports show that the WLAN market dramatically increases from $100 million in 1995 to about $4.5 billion in 2006.

With the market promotion, IEEE (Institute of Electrical and Electronics Engineers) LAN (Local Area Network)/MAN (Metropolitan Area Network) standard committee developed a set of standards named 802.11 for WLAN computer communication in the 5GHz and 2.4 GHz public spectrum bands[1]. 802.11 supports two network structures which are ad hoc networks and infrastructure networks respectively. For infrastructure networks, some of the wireless terminals are promoted as a centralised controller. For ad hoc networks, each terminal directly communicates with each other in a peer to peer mode and there is no centralised controller.

The physical layer used in 802.11 is fundamentally different from traditional wired media and it brings some challenges for the MAC (Medium Access Control) layer decisions. Firstly, wireless media (WM) has no absolute boundary and is unprotected from outside signals. With mutual interferences, communications over the WM is significantly less reliable than wired networks. Secondly, a wireless ad hoc network is a

multihop network due to the limited transmission range of wireless terminals. As a result, terminals may be out of the transmission range of each other. This feature causes hidden problems as we will see later in Chapter 2. Finally, power is valuable resource since most of the wireless terminals are powered by batteries. Therefore, how to reduce power consumption to prolong the battery life becomes one of the major issues in wireless ad hoc networks.

Both simulation and theoretical experiences [2-4] before show that the capacity of wireless ad hoc networks is bounded and sensitive to network size, topology and traffic pattern. From the MAC aspect, the network capacity is constrained by interferences from adjacent transmissions. From the routing aspect, the network capacity is constrained by forwarding burden due to the nature of multihop. In other words, not only does a node need to access the channel for its own traffic but also need to relay traffic from adjacent nodes to maintain a connected network.

Spatial reuse [5-7] plays an important role in improving the network capacity by allowing more simultaneous transmissions. Given a network with a fixed nodal density, one way to increase the spatial reuse is transmission range control (TPC) which can be achieved by controlling the transmission power at transmitters. Another promising technique for spatial reuse is directional antennas which can focus their power in an intended direction and allow transmission in other directions to happen concurrently. The objective of this thesis is to enhance 802.11 performances by exploiting the benefits of spatial reuse that is achieved by *TPC* and *directional antennas*.

The main contributions of this thesis are listed as follows:

- We study details of different functionalities in 802.11 DCF (Distributed Coordination Function) and argue the effectiveness of RTS (Request to Send)/CTS (Clear to Send) handshake in 802.11 through both theoretical and simulation results.
- We identify both benefits and challenges for transmission power control and develop *spatial TPC* based on *basic TPC* to fully exploit the benefits of spatial reuse achieved by transmission range control.
- We study the impact of implementing directional antennas in wireless ad hoc networks and develop four schemes of directional MAC protocols based on 802.11 in OPNET to fully exploit the benefits of spatial reuse achieved by directional antennas.

The rest of this thesis is organised as follows. Chapter 2 introduces background knowledge of wireless ad hoc networks including physical issues, neighbor discovery, MAC protocols as well as broadcasting algorithms. Chapter 3 presents the fundamental functions of 802.11 and argues the effectiveness of *RTS/CTS* handshake in 802.11, followed by introduction to OPNET simulation in Chapter 4. In Chapter 5, we study how performances of wireless ad hoc networks can be enhanced by exploiting the benefit of spatial reuse achieved by TPC. In Chapter 6, we firstly discuss the impact of implementing directional antennas on MAC decisions, then present four schemes based on basic directional MAC protocol followed by extensive simulation studies and analyses. Finally, we draw our conclusions and future work in Chapter 7.

# Chapter 2: Preliminaries

This chapter presents the background knowledge of wireless ad hoc networks including physical issues, neighbor discovery and location awareness, MAC protocols as well as broadcasting algorithms.

## 2.1 Physical issues

Physical layers are the most basic network layers providing services requested by MAC layers. Physical layer techniques directly impact on the design of MAC protocols. Throughout this thesis, we adopt the most common physical layer settings in 802.11 and they are listed as follows:

- Single channel with base frequency 2.4 GHz and bandwidth 22 MHz
- DPSK (Differential Phase Shift Keying) modulation with DSSS (Direct Sequence Spread Spectrum) spreading code
- Line of sight (LoS) propagation

Next, we introduce two key physical issues that affect our MAC decisions: *antenna* patterns and *propagation* models.

### A. Antennas

Antennas are devices that radiate and receive electromagnetic power from free spaces. According to the radiation pattern, antennas can be divided into omni-directional antennas and directional antennas[1]. An omni-directional antenna radiates its power uniformly in all directions while a directional antenna only radiates its power in one direction. According to beam-pattern selection, directional antennas can be divided into

[1] We only consider single-beam antennas in this thesis, multi-beam antennas are not considered.

*switched antennas* and *adaptive antennas*. S*witched antennas*[2] can only beamform its main lobe in certain fixed positions while *adaptive antennas*[3] can point its main lobe in any direction.

A directional antenna is composed of side and main lobes as shown in Figure 1. Though the side lobe can become harmful interference to other receivers in the nearby vicinity, we ignore the impact of side lobe in this thesis since the transmission range of the side lobe is usually much smaller than the main lobe. The beamwidth Θ of directional antennas is between 0 and 360 degree, and it quantifies the directionality of an antenna.

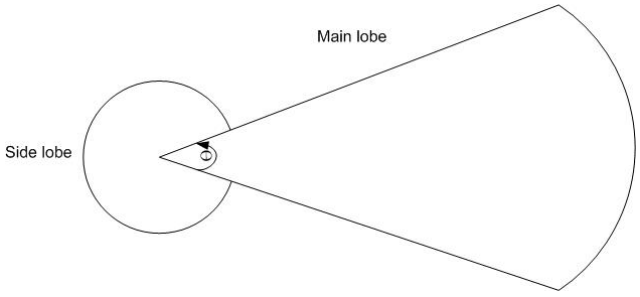



**Figure 1: Directional antenna model**

Antenna gain is a parameter that measures the directionality of an antenna. Given an antenna, assume $P_d$ is the power it radiates in the direction of strongest signal and $P_o$ is the power it radiates its power in all directions, then the antenna gain (dBi) can be calculated as follows:

$$G = 10 \bullet \log_{10} \frac{P_d}{P_o}$$

[2] Also referred to as sectored antennas
[3] Also referred to as steered antennas

One thing that we should note is that power is not added by the antenna but redistributed to radiate more power in certain directions. The antenna gain of omni-directional antennas is 0 dB for all directions. And for directional antennas, antenna gain is positive in certain directions and negative in other directions.

Before moving on to propagation section, we define three kinds of conflictions as shown in Figure 2 for SISO (Single In Single Out) systems:

1. At any time only one packet can be transmitting at the transmitter.
2. At any time only one packet can be received at receiver. If two or multiple packets arrive at receiver at the same time, all these packets are marked as bad packets.
3. Packet transmission and reception can not happen at the same time. All the received packets are marked as bad packets when the node is transmitting a packet.

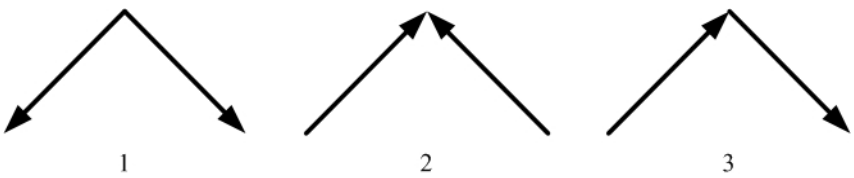



**Figure 2: Three conflictions scenarios**

**B. Propagation**

Propagation models study how radio signals behave when they are propagated between radio transmitters and receivers. The propagation mode we adopt in this thesis is free space which assumes the transmitter and receiver have a clear, unobstructed line of sight (LoS) path between them. We are interested in what the power level is when the signal arrives at the receiver. This power level is subject to a number of parameters, namely:

- Transmit power ($P_t$)
- Transmit antenna gain ($G_t$)
- Path loss ($L$)
- Receive antenna gain ($G_r$)

In free space, the path loss is proportional to the inverse of the square of $d$ where $d$ is the distance from the radio source. And the path loss $L$ for the given distance $d$ under base frequency 2.4 GHz is about $\frac{1}{(32\pi \bullet d)^2}$. So the received power ($P_r$) after signal attenuation can be calculated as follows:

$P_r = P_t \bullet G_t \bullet L \bullet G_r$.

One parameter at receiver is the packet reception power threshold $P_{reception}$. If the received power $P_r$ exceeds the $P_{reception}$, the received packet can be successfully decoded. Otherwise, the packet is considered as noise. Another optional parameter at the receiver used in most papers is the carrier sense power threshold $P_{noise}$[8, 9]. $P_{noise}$ is usually the power that is not big enough for successful packet reception but big enough to corrupt a concurrent packet reception[4].

Given a fixed transmit power and path loss factor, we can get two ranges namely *transmission range r* and *interference range R* according to two parameters $P_{reception}$ and $P_{noise}$. Except for the node overlap, a receiver will and must be one of three areas of a transmitter according to the distance between the transmitter and receiver $d$ as shown in Figure 3:

[4] Clearly, $P_{noise} < P_{reception}$.

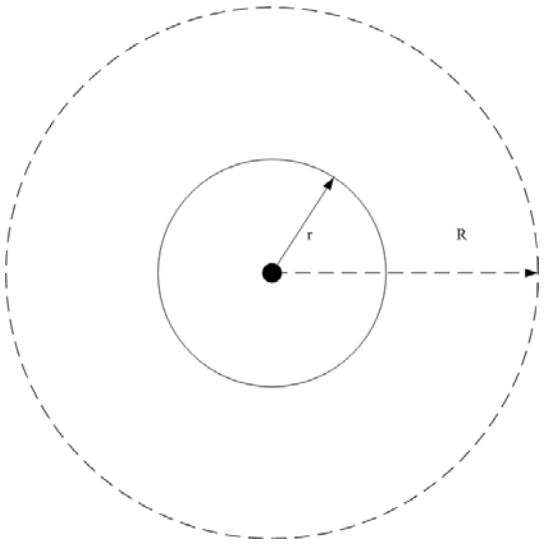



**Figure 3: Interference range**

- Transmission area when $0 < d \leq r$: For a receiver in this area, it can successfully receive packets from the transmitter.

- Interference area when $r < d \leq R$: For a receiver in this area, the packet from the transmitter cannot be successfully received, however, the packet reception at receiver can be interfered by the transmitter.

- No correlated area when $d > R$: For a receiver is in this area, the receiver can neither receive packets from the transmitter nor be interfered by it.

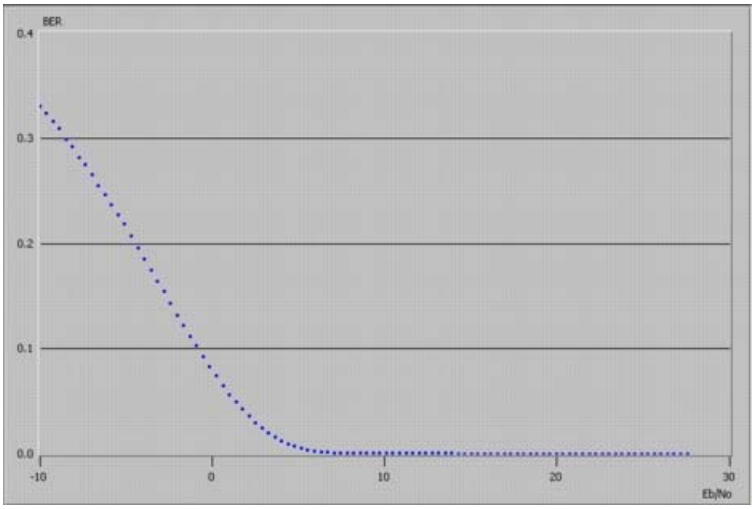



**Figure 4: DPSK modulation**

To get a general idea of the relationship between transmission range and interference range, assume there is only one interference transmitter, and then the *SNR* at the receiver can be calculated as follows:

$$SNR = \frac{P_r}{P_i} = \frac{P_t \bullet G_t \bullet L_t \bullet G_r}{P_t \bullet G_t \bullet L_i \bullet G_r} = \left(\frac{d_i}{d_t}\right)^2 \geq SNR_{thresh}$$

$$d_i \geq \sqrt{SNR_{thresh}} \bullet d_t$$

From Figure 4, we can see that when $SNR_{thresh}$ exceeds nine, the BER is low enough for successful packet reception. Hence, we have $d_i \geq 3d_t$.

Under two circumstances, a receiver cannot successfully receive a packet:

1. One or two transmitters in the receiver's transmission range are transmitting packets while the receiver is receiving a packet. In this case, two or more packets collide at the receiver.
2. Transmitters in the receiver's interference range are transmitting packets while the receiver is receiving a packet and the interference noise exceeds $SNR_{thresh}$. In this case, the packet is corrupted.

## 2.2 Neighbor discovery and location awareness

Neighbors of a node are a collection of nodes that are within its transmission range. A complete knowledge of neighbor information is essential for most of MAC and routing protocols [10-14] as well as topology control algorithms [15, 16]. The procedure of neighbor information collection is called neighbor discovery. Neighbor discovery needs to be initiated under two cases. The first case is when the network firstly initiated and the second happens when neighbor nodes move in or out of the transmission range of each other. So how often the neighbor discovery should be initiated is dependent on the network nodal mobility.

The method for a node to discover its neighbors is usually to send periodical *Hello* messages with own ID encapsulated. For one way discovery mechanism, a node's neighbors find the node after they received the *Hello* message from that node. However, for a node, if more than one of its neighbors send *Hello* at the same time, then *Hello* messages collide at the node and none of them will be discovered by that node. Since it is impossible to obtain topology and traffic information beforehand for neighbor discovery, the random access mechanism CSMA (Carrier Sense Multiple Access) can be used to avoid collisions during neighbor discovery.

Location information is very useful information especially for some topology aided broadcasting and routing protocols. However, location of each node is ever changing for a mobile ad hoc network. One easy way to obtain a terminal's location is to use Global position system (GPS) [17]. With the aid of GPS, a terminal can get the global location

information of itself. By exchange of each other's global location position, adjacent nodes can calculate the distance and direction to a neighbor node.

However, sometimes, the assumption of GPS is hard to achieve. In this case, several radio location techniques can be used to decide the reference location of each other which are time of arrival, signal strength or angle of arrival (AoA) [18]. For time of arrival (ToA), the trip time is measured after receiving a message from an adjacent node. And the distance between the two nodes can be obtained by multiplying the trip time and propagation delay. For signal strength, given the transmit power, path loss factor and the received power, we can determine the distance from source node to destination node. For AoA, when a receiver receives a signal from transmitter, it can decide in which direction the signal comes from [18-20].

### 2.3 MAC protocols

The purpose of MAC protocols is to manage contention wherein multiple terminals compete for the same communication channel. Different diversities can be used when contending the shared medium such as *Frequency* Division Multiple Access (FDMA), *Time* Division Multiple Access (TDMA). In this thesis, we only consider TDMA based MAC protocols [21-24].

In a TDMA system, time is divided into slots and all nodes are synchronised to slot boundaries. Previous studies on aloha protocols show that by simple synchronisation, we decrease the packet confliction time to half the one without synchronisation and hence double the channel throughput. The question now is how to assign time slots to a

number of backlogged nodes so that packets queued at different nodes can be successfully delivered without any collision.

According to slots assignment, medium access approaches can be broadly categorised into *reservation based and contention based*. For *reservation based* protocols, a node that wants to access the channel makes reservation, and transmissions only happen after successful rendezvous. It is more suitable for a network that either is static or has central access control.

Fixed assignment is a simple approach of *reservation based* access. For *fixed access* approach, time slots are partitioned and each station is assigned a portion of time slots. This approach works well if all the terminals are constantly backlogged. However, if the traffic is bursty, a number of slots are wasted since some of the terminals have no packet to transmit. Assume there are n terminals each with a probability $p$ to transmit packets, the probability of $k$ time slots are wasted in one round access is $p^{n-k} \bullet (1-p)^k$. When $n$ is large or $p$ is small, $k$ increases dramatically indicating more time slots are wasted.

Different from fixed assignment, *random access* approaches save the wasted time slots by dynamically assigning the channel resource to different terminals. In other words, time slots are only assigned to the terminals which are backlogged. The problem is how to ensure a slot is only assigned to one terminal. Since collisions happen if more than one terminal are assigned to a same time slot.

CSMA is a well known random access protocol firstly proposed by Kleinrock [25]. Different from aloha protocols, CSMA avoids collisions by listening to the channel before transmission. CSMA protocol works as follows: A station listens to the channel before it transmits packets. If the channel is free according to channel clear access (CCA), the station immediately transmits packets. Otherwise, it waits until the channel is free, and then chooses a random backoff time. When backoff finishes, the station transmits packets. The random backoff avoids collisions by preventing packet burst once the channel becomes free.

One fatal issue that greatly degrades the performance of CSMA is the hidden node problem due to the intrinsic multi-hop that not all the terminals are within transmission range of each other. Some terminals may be hidden from each other. This causes the unawareness of transmission happening between hidden terminal and one of its neighbors and collision happens when the node transmits packet to the neighbor.

To address the hidden node problem, Kleinrock proposed busy tone multiple access (BTMA) [26]. In BTMA, the total available channel is divided into two separate channels: a narrow- bandwidth message channel and a busy tone channel. And a central station resides in the network within range of all terminals. Once the station senses the message channel busy, it sends a busy tone (BT) on the busy tone channel to inform all terminals that the message channel is being occupied.

Haas et. al. [27] proposed dual busy tone multiple access (DBTMA). Instead of one busy tone in BTMA, DBTMA implements two busy tones: the transmit tone ($BT_t$) and the receive tone ($BT_r$) on separate narrow-bandwidth channels. The message channel is

considered to be busy when one of the busy tone channels is sensed busy. If one terminal has data packet to transmit and the channel is free, it turns on its $BT_t$ and transmits a RTS on the message channel. When RTS is received by the destination terminal, it turns on its $BT_r$ to inform the successful reception of RTS. The use of RTS and two busy tones avoids the probability of RTS collisions and increases the network throughput.

As another solution for the hidden node problem which is adopted by 802.11, multiple access collision avoidance (MACA) was proposed by Karn [28]. MACA uses *RTS* and *CTS* handshake before the data communication. The exchange of short control messages increases network throughput by informing neighbors of both transmitter and receiver about the transmission going between the transmitter and receiver and silencing their neighbors to avoid collisions. [29-31]

### 2.4 Broadcasting

Broadcasting is a fundamental operation in which a message is sent from a source node to all other nodes in the network. Due to the ever-changing topology, it is frequently used in wireless ad hoc networks such as route discovery stage in routing protocols [32, 33]. The simplest and most common approach of broadcasting is known as flooding in which each node rebroadcasts the first copy of received message to its neighbors. Two things should be noted for broadcasting. Firstly, broadcast messages do not need to be acknowledged. Secondly, broadcast messages also follow MAC protocol to access the channel.

Despite its simplicity, flooding generates a great number of redundant messages wasting valuable network resources such as bandwidth and power known as the broadcast storm problem [34, 35]. This redundancy is mainly caused by the overlap of transmission range between flooding nodes in geographical areas. Consider the scenario shown in Figure 5, where node A initiates a broadcast message and node B decides to rebroadcast it. Let *I* and *II* denote the transmission areas covered by node A and B. The additional area covered by node B is

$$II - I \cap II = \pi \bullet r^2 - 4\int_{d/2}^{r} \sqrt{r^2 - x^2}\, dx$$

Where $r$ is the transmission radii and $d$ is the distance between node A and B, clearly $d \in (0, r]$.

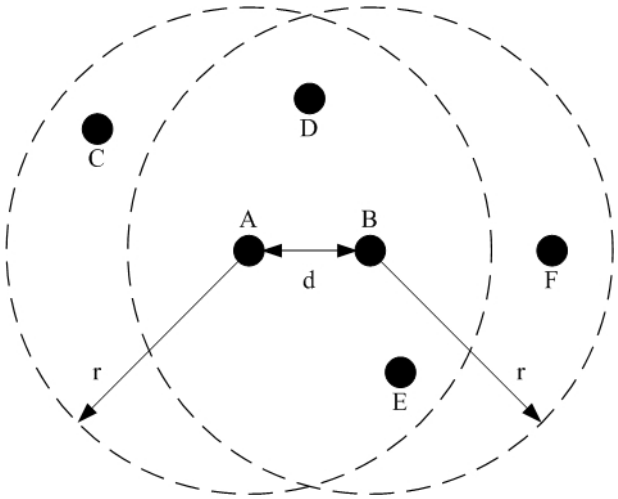



**Figure 5: Broadcasting intersection**

We can calculate the smallest additional rebroadcast area is about $0.6\pi r^2$ when $d$ is equal to $r$. In other words, given a fixed nodal density $\lambda$, nodes that reside in an area size of $0.4\lambda\pi r^2$ receive the broadcast message twice. And the redundancy becomes more serious when the network nodal density is high.

Extensive work have been done to design an efficient broadcasting algorithm that can reduce as many redundancies as possible without losing delivery reliability. We can broadly divide them into two categories: *statistical based* and *location based*. Next, we briefly introduce the related work of two broadcasting schemes.

**A. Statistical based**

Counter based broadcasting is one of statistical based algorithms. The counter based broadcasting is under the observation that the more times a node has heard the same broadcast message, the less additional coverage is if the node rebroadcast the message. In counter based broadcasting, each node maintains a counter recording the number of times it has received the same broadcast message, and the node only rebroadcasts the received broadcast message when its counter does not exceed a predefined threshold *C*. The question is, giving a network with a fixed nodal density, how we define the *C*. If *C* is too large, redundancy cannot be efficiently decreased, and if C is too small, some nodes may not receive the broadcast message. Tseng's work [34, 35] show that a threshold of 3 or 4 is feasible.

Different from counter based broadcasting, gossiping is a probabilistic approach based on the percolation theory [36], which can be used to improve the broadcasting. When gossiping, there is a probability *p* that a node may forward the first received copy of a packet to its neighbors. There is therefore a complementary probability of *1-p* that a packet will be discarded. This means that gossiping is equivalent to flooding when $p$=1 since every packet will be forwarded. Gossiping exhibits a bimodal behavior or phase transition phenomenon based on the percolation theory. There is a critical threshold $p_0$.

When $p < p_0$, the gossip message will quickly die out and when $p > p_0$, nearly all the nodes in the network receive the gossip message. In other words, with a propagation probability *p*, either no node receives the message or almost all of them do.

Haas et. al. [36] studies the performance of gossiping through simulation results and suggests several optimisations such as preventing premature gossip death. He also incorporates gossiping in an on-demand routing protocol *AODV*. Haas concludes that the gossiping probability between 0.6 and 0.8 is sufficient to ensure successful message propagation. Haas also claims that gossiping can achieve 35% fewer messages compared to flooding.

Li et. al. [37] proposes regional gossip routing with the aid of GPS. Instead of gossiping messages to the whole network, regional gossiping only gossips the routing *RREQ* messages within some restricted region to reduce the overhead of the route discovery in ad hoc networks. The gossip region is restricted in an elliptical area which is defined by the ellipse factor l. Li claims up to 94% messages can be saved compared to global gossiping. This high performance is achieved by restricting the flooding area into a small area compared to the global flooding. So the overhead out of the flooding area is zero, and the overhead in the flooding area is also reduced by gossiping. As a result, the performance of regional gossiping is much better than simple flooding.

**B. Location based**

Location based broadcast algorithms are based on the location information of its neighbors such as distance and direction obtained either from GPS or neighbor discovery stage. We can further divide location based broadcast algorithms into

*neighbor-designating* (*sender-based*) algorithms and *self-pruning* (*receiver-based*) [38] algorithms. For *neighbor-designating* algorithms, each broadcasting node selects only a subset of its neighbors to rebroadcast the broadcast message. For *self-pruning* algorithms, upon receiving a broadcast message, the receiver decides whether to forward this broadcast message using local information by itself.

Directional antennas also present the opportunity to mitigate the impact of the broadcast storm problem. Hu [39] proposes the three schemes for directional broadcasting including *on/off* directional broadcasts*, relay-node-based* directional broadcasts and *location-based* directional broadcasts. In the *on/off* directional broadcast scheme as shown in Figure 6, the first time a node receives a broadcast packet, it forwards the packet in all directions other than the direction that the packet came. In the *relay-node-based* directional broadcast scheme as shown in Figure 6, a node only forwards the broadcast packet to its neighboring nodes that are furthest from it in each direction. The necessary distance information is obtained by measuring the signal strength of periodic "*Hello"* messages. This scheme is similar to the Most Forward with Variable Radius (MVR) routing protocol [40]. In the *location-based* directional broadcast scheme, a node also forwards the packet in the directions other than the incoming direction similar to the on/off scheme. However, the delays for different beamforms are not uniform, which is determined by the extra coverage that beamform covers. The larger the extra coverage is, the smaller the delay will be.

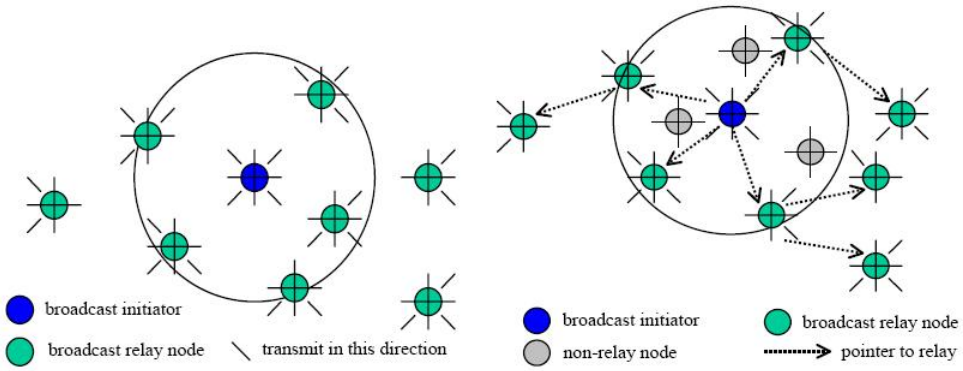



**Figure 6: *on/off* and *relay-node-based* broadcast schemes**

# Chapter 3: 802.11 DCF

802.11 MAC supplies the functionality providing reliable delivery mechanism for data packets over unreliable wireless media. There are two different access mechanisms: the fundamental access mechanism DCF and an optional access mechanism PCF. Since both spatial TPC in Chapter 5 and directional MAC in Chapter 6 are based on 802.11 DCF, we present functional details of 802.11 DCF in this chapter before moving on to the next stage.

This chapter consists of three sections. In the first section, we introduce how DCF timing, CS and backoff work together to provide reliable data delivery in DCF followed by a brief introduction to power saving mechanism in 802.11. In the second section, we study the performance of 802.11 DCF and argue the effectiveness of RTS/CTS handshake. We summarize this chapter in Section 3.

## 3.1 Functional description

### 3.1.1 DCF

DCF is the fundamental access method in 802.11 MAC. It is based on the CSMA/CA as presented in Section 2.3. However, more mechanisms in DCF such as DCF timing, virtual CS and RTS/CTS handshake make it more reliable as compared to CSMA/CA. We now present these different mechanisms in more details in this section.

#### A. DCF Timing

In 802.11, the signal propagation delay is supposed to be less than 1 $\mu s$, so the distance between any two STAs should be within 300 metres. A larger distance will degrade the performance of 802.11 since a longer propagation delay will be considered as either

packet timeout for the transmission initiator or channel free for other terminals even when a packet is still on its way.

For 802.11 DCF, in addition to slot time, there are three interframe spaces (IFS) namely: *short interframe space (SIFS)*, *DCF interframe space (DIFS)* and *extended interframe space (EIFS)*. The default time duration of time slots and three IFSs for 802.11b are listed in Table 1.

| | |
|---|---|
| *Slot time* | *20 μs* |
| *SIFS* | *10 μs* |
| *DIFS* | *50 μs* |
| *EIFS* | *364 μs* |

**Table 1: IFS times**

Calculations of different time durations are defined in 802.11 as follows:

$$aSIFSTime = aRxRFDelay + aRxPLCPDelay + aMAC\,\mathrm{Pr}\,oces\,\mathrm{sin}\,gDelaly + aRxTxTurnaroundTime$$

$$aSlotTime = aCCATime + aRxTxTurnaroundTime + aAir\,\mathrm{Pr}\,opagationTime + aMAC\,\mathrm{Pr}\,oces\,\mathrm{sin}\,gDelay$$

$$aDIFSTime = aSIFSTime + 2 \times aSlotTime$$

$$aEIFSTime = aSIFSTime + (8 \times ACKSize) + a\,\mathrm{Pr}\,eambleLength + aPLCPHeaderLngth + aDIFSTime$$

From equations above, we can see that a SIFS time is the sum of physical and MAC processing delays plus the transition time from reception to transmission. In other words, SIFS duration is the time from the last bit of physical signal received at the receiver till

the receiver responds to the first bit of signal to the physical channel. SIFS is the shortest time slot in 802.11 and has the highest priority to access the channel. In 802.11 DCF, SIFS is used when a STA responds a packet or retransmits a packet to keep the frame exchange sequence consistent so that its frame burst will not be interrupted by other STAs.

DIFS is the sum of two slot times and one SIFS time. In other word, a DIFS time is the period that two STAs finish a round handshake. When a STA tries to initiate a new transmission, it can access the channel only if the channel is free for longer than *aDIFSTime*. EIFS is the largest time duration in IFSs and it has the lowest priority. A STA has to wait for an EIFS time to try to access channel again, if it detects a collision or a corrupted packet by high interferences at its receiver.

**B. VCS and NAV**

The mechanism of sensing the station's adjacent power level is called physical carrier sensing (PCS). Another mechanism is called virtual carrier sensing (VCS) that is implemented at MAC layer. It encapsulates duration information in MAC control or data frames indicating how long the channel will be occupied by its transmission. Other stations that overhear the frame update their NAV (Network Allocation Vector) and keep silence until the NAV expires. The channel is considered to be free only if both PCS and VCS indicate the channel is free. The calculation for NAV duration of different packets is listed below:

$$Duration_{RTS} = \frac{S_{CTS} + S_{ACK}}{R_C} + \frac{S_{Data}}{R_D} + 3 \times aSIFSTime$$

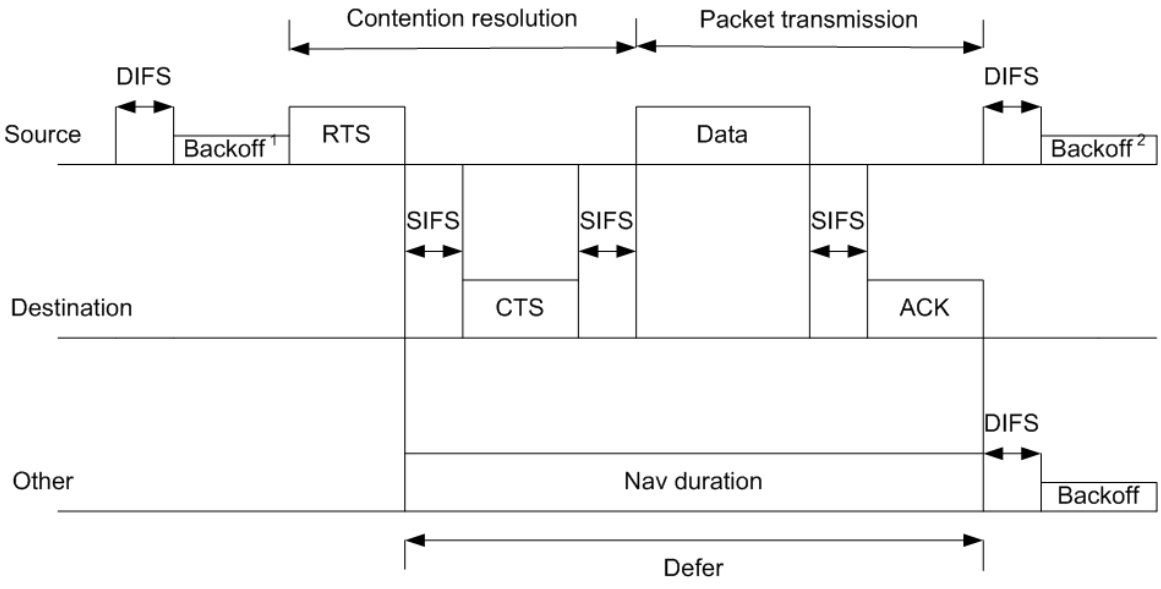
$$Duration_{CTS} = \frac{S_{Data}}{R_D} + \frac{S_{ACK}}{R_C} + 2 \times aSIFSTime$$

$$Duration_{Data} = \frac{S_{ACK}}{R_C} + aSIFSTime$$

$$Duration_{ACK} = 0$$

where $R_C$ and $R_D$ are data rates for control packets and data packets respectively and $S$ stands for the packet size.

When a STA receives a valid packet, if the packet is not destined to itself and the NAV value in that packet is greater than its current NAV value, the STA should update its NAV value with the duration information in that packet. One thing that should be noted is when a STA updates its NAV according to a *RTS* packet, its NAV value should be reset to the most recent value before the *RTS* was received, if the channel is free for longer than $2 \times aSIFSTime + 2 \times aSlotTime + \frac{S_{CTS}}{R_C}$ time since the end of that *RTS* reception.




[1] This backoff is skipped if the media is idle.

[2] CW is set to the minimum value.

**Figure 7: DCF channel access**

**C. RTS/CTS**

*RTS/CTS* is an optional function in 802.11. It works as follows: Before a STA sends a *Data* packet to the destination, it first sends a *RTS* to the channel. Upon receiving the *RTS* message, the destination STA sends back a *CTS* to the source STA. If the source STA successfully receives the *CTS*, it can then start the data packet transmission. Otherwise, it tries to access the channel by initiating the *RTS/CTS* handshake again. By using small control packets to access the channel before real data transmission, *RTS/CTS* decreases the data packet collision probabilities[28].

Combined with NAV and DCF timing, *RTS/CTS* handshake silences all neighbors of both source and destination nodes during their transmission. Assuming node T initiates a transmission to node *R* and the transmission range covered by node *T* and *R* are *I* and *II* respectively as shown in Figure 8, the *RTS* and *CTS* exchange silences all the nodes residing in $I \bigcup II$ that could potentially interfere with their transmission. However, all the nodes in the area $\overline{I \bigcup II}$ such as node *D* are free to initiate transmissions.

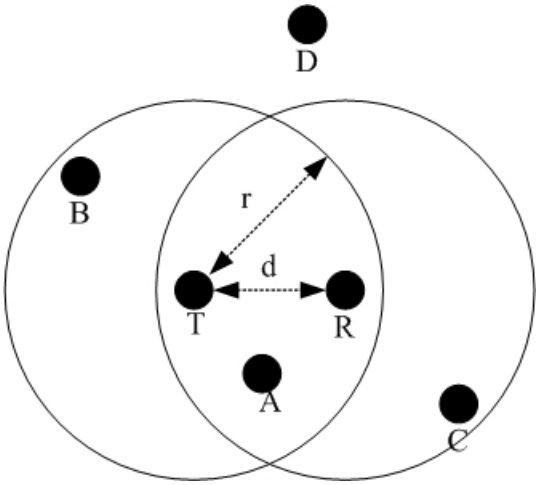



**Figure 8: RTS/CTS exchange**

Note that all the description above is under the assumption that *RTS/CTS* are transmitted using omni-directional antennas. As we will see later in Chapter 6, if directional antennas are used at a node's transmitter, some of the node's neighbors may not be able to receive *RTS* or *CTS* from that node which is called the new hidden node problem. In this case, we would like one of RTS/CTS or both of them to be transmitted omni-directionally to inform its neighbors of the on-going transmission.

**D. Backoff**

Backoff is a well known method to generate time differences for multiple terminals that contend for the same medium to avoid collisions. The binary exponential backoff (BEB) algorithm is adopted in 802.11. Each STA maintains a *contention window (CW)*. The CW is doubled after a collision and it is set back to a minimum value after a successful transmission. The backoff time is generated uniformly from [0, CW]. The backoff timer starts to count down when the medium has been free for DIFS period indicating by both PCS and VCS and it is frozen once the medium becomes busy.

In 802.11, backoff must be executed in the following cases:

1. A STA initiates a new transmission and the medium is sensed busy
2. A STA tries to retransmit a packet
3. After each successful transmission

The first two backoff cases increase the network throughput by generating time difference for multiple terminal nodes that compete for the channel once the channel becomes free. And the third backoff case provides fairness by giving other nodes

opportunities to compete for the channel after a successful transmission. One thing to note is that for case 2, the contention window size is always the minimum value of contention window.

And the backoff should be skipped in the following cases:

1. A STA initiates a new transmission and the medium is sensed idle for DIFS time.
2. A STA receives a packet (*RTS* or *Data*) from a transmitter that needs to respond back.

From the DCF timing and backoff mechanism, we can see that the priority for different types of packets is: *response packets* > *retry packets* > *packets waiting in upper stream buffer*.

### 3.1.2 Power saving

For power saving (PS) schemes, a wireless terminal may be in one of two different power states: *awake* as the terminal is fully powered and *doze* as the terminal is in a low power level and not able to transmit or receive.

In 802.11 PS model, all the STAs are awake during ATIM window. As shown in Figure 9, during the ATIM window, if a STA has packets to transmit, it transmits an ATIM frame to its destination STA. If the destination STA receives the ATIM frame, it acknowledges the ATIM frame and stays *awake* waiting for the packet transmission. If a STA does not receive any ATIM frame during ATIM window, it enters the *Doze* state at the end of the ATIM window.

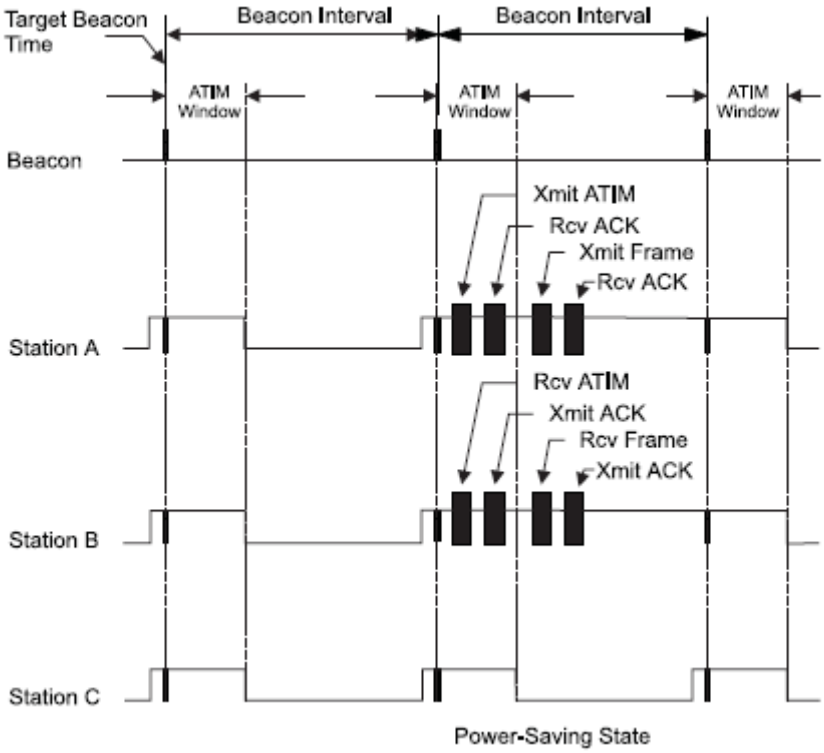



**Figure 9: 802.11 power saving**

This PS scheme in 802.11 saves the unnecessary power consumption for nodes in idle waiting status. As we will see later in Chapter 5, a more efficient power saving scheme named transmission power control that saves the power consumed during packet transmissions will be introduced.

### 3.2 802.11 performance analysis

We now study the performance of 802.11 DCF under three different topologies as shown in Figure 10. The three topologies are the most basic topologies underlying other more complicated topologies and traffic patterns. We also study the effectiveness of *RTS/CTS* handshake.

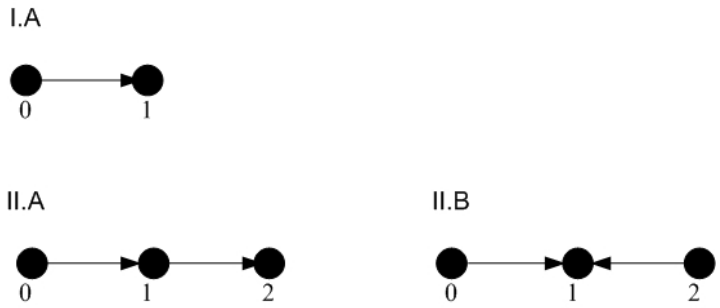



**Figure 10: Atomic topologies**

#### 3.2.1 Numerical results

As mentioned before, *RTS/CTS* is an optional function in 802.11. The intention of RTS/CTS handshake is to announce the coming data transmission to adjacent nodes to avoid collision and hence increase the channel throughput. However, the exchange of *RTS/CTS* takes considerable time. There are two things we are interested in. Firstly, giving a network with random topologies and unpredictable traffics, what is the probability that a hidden problem happens at a node. If all adjacent nodes are idle, and there is no risk of collisions, then the *RTS/CTS* exchange is meaningless and it wastes valuable bandwidths. Another thing is that even if collisions have happened, does the channel bandwidth saved by collision avoidance outweigh the one wasted on *RTS/CTS* exchange.

Figure 11 shows the time axis during 4-way or 2-way handshake when each packet is interspaced by *aSIFSTime*. $T_{off}$ is the time consumption for basic *Data-ACK* handshake and $T_{on}$ is the time consumption for a complete 4-way handshake. Time $T$ can be considered as the time consumption for the *RTS/CTS* exchange.

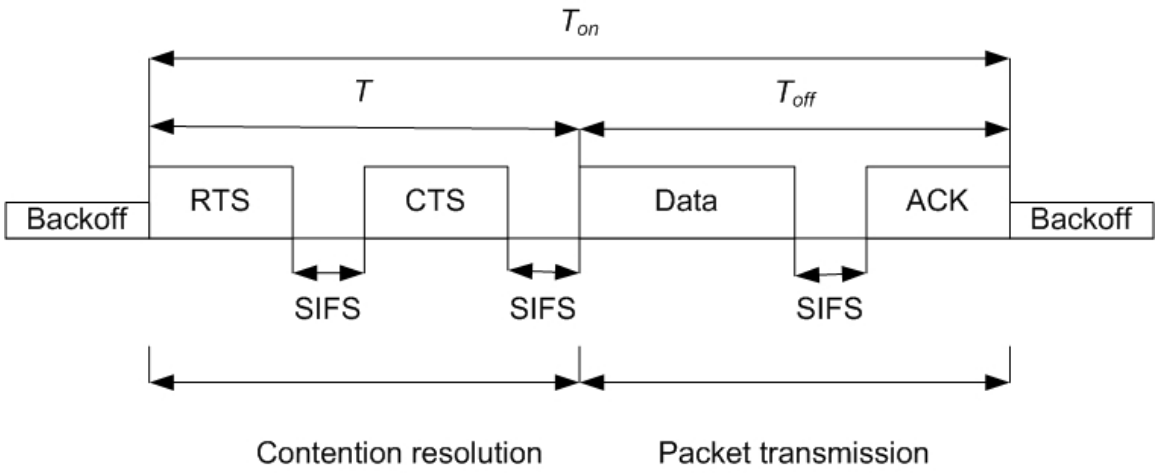



**Figure 11: Four way handshake**

Note that the data rate for control packets (*RTS/CTS/ACK*) is different from the data rate for data packets. The control packet data rate for 802.11b and 802.11a are 6M bps and 1M bps respectively. And the packet size for *RTS, CTS*, and *ACK* are 160 bits, 112 bits and 112 bits respectively. Ignoring the MAC header, the transmission time for one round handshake with *RTS/CTS* on ($T_{on}$) and off ($T_{off}$) are

$$T = \frac{S_{RTS} + S_{CTS}}{R_C} + 2 * SIFS \qquad (1)$$

$$T_{off} = \frac{S_{Data}}{R_D} + \frac{S_{ACK}}{R_C} + SIFS \qquad (2)$$

$$T_{on} = \frac{S_{RTS} + S_{CTS} + S_{ACK}}{R_C} + \frac{S_{Data}}{R_D} + 3 * SIFS = T_{off} + T \qquad (3)$$

where $R_C$ and $R_D$ are data rates for control packets and data packets respectively and $S$ stands for the packet size.

**A. Throughput without risk of collisions**

For a pair of nodes that communicate with each other, there is no risk of collisions. We keep the traffic heavy enough to saturate channel and for simplicity, we also ignore the effect of packet header and backoff procedure. For a given time, throughput with *RTS/CTS* on ($Thr_{on}$) and off ($Thr_{off}$) are

$$Thr_{on} = \frac{S_{Data}}{T_{on}}$$

$$Thr_{off} = \frac{S_{Data}}{T_{off}}$$

Since the maximum MSDU in 802.11 is 2304 bytes, we generate the throughput for packet size of 128, 256, 512, 1024, 2048 bytes respectively under data rate 11M bps. From the result shown in Figure 12, we can see that the throughput with RTS on dramatically surpasses the throughput with RTS off. As mentioned before, this throughput increase is achieved by saving the time of unnecessary RTS/CTS exchange for more data transmission.

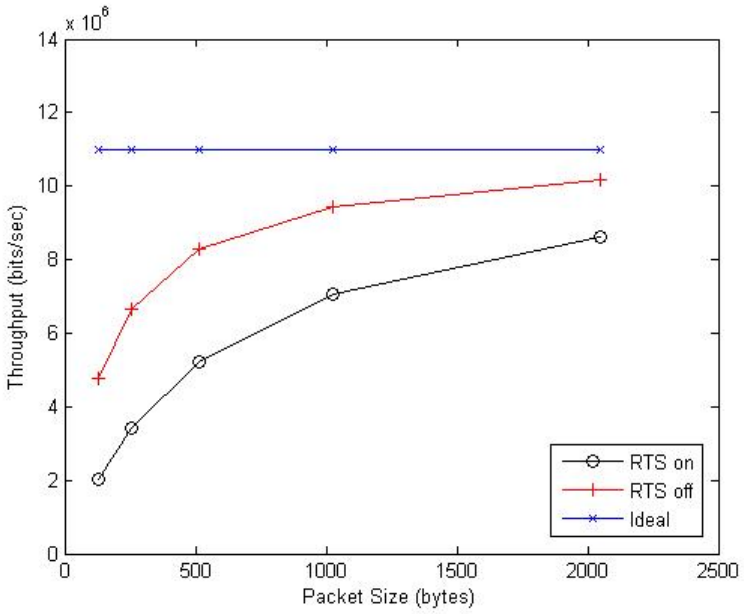



**Figure 12: Throughput without collision risks**

**B. Throughput with collisions**

All the results we got above are under the assumption that there is no risk of collisions. This is obviously untrue and against the intention of *RTS/CTS*. Now we study how collisions impact the network throughput when turning off *RTS/CTS*.

Assume a node is receiving a packet from a transmitter and there are $n$ other neighbor nodes for the receiver. Each of them has a probability $p$ to transmit packets which represent quantity of the traffic load. To ensure there is no collision at the receiver, all its neighbors should keep silent. So the probability of successful transmission for receiver is $(1-p)^n$ and the throughput with *RTS* off ($Thr_{off}$) is

$$Thr_{off} = (1-p)^n \bullet \frac{S_{Data}}{T_{off}}$$

Since *RTS/CTS* can silence all the nodes' transmissions and hence avoid collisions, the throughput with *RTS* on ( $Thr_{on}$ ) is same with the one under no collision:

$$Thr_{on} = \frac{S_{Data}}{T_{on}}$$

We generate throughput for 8 neighbors and data packet size of 512 bytes under different transmission probabilities as shown in Figure 13. We can see that the throughput for *RTS on* is about 5.2M bps and it does not change as the transmission probability increases. In contrast, the throughput with *RTS* off drops dramatically as the probability increases. However, when probability is below 0.06, the throughput still surpasses the one with *RTS* on. This is because even some of its packets collide at the receiver, the number of successfully received packets is still great than the one with *RTS* on. But when transmission probability surpassed a threshold, collisions become more serious. As a result, nearly none of its packets can be successfully received, and hence the throughput drops to nearly zero.

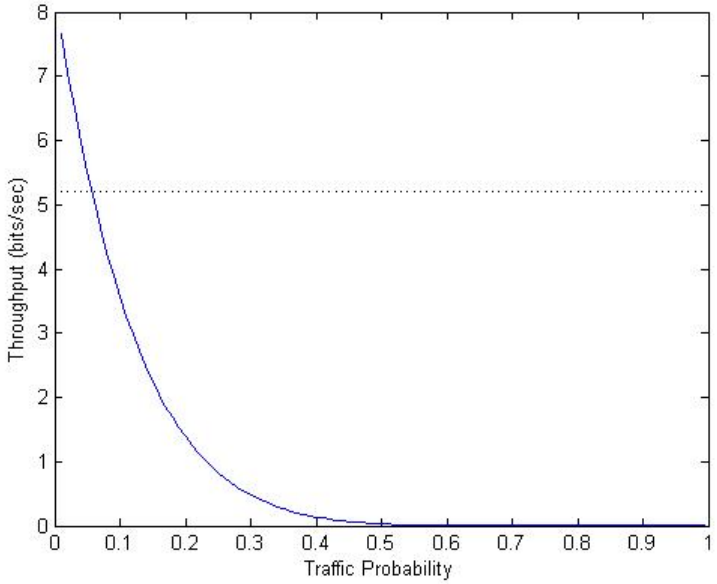



**Figure 13: Throughput under collisions**

### 3.2.2 Simulation results and analysis

We also generate *throughput* and *delay* in OPNET simulation for three basic topologies as shown in Figure 10. We make the packet interval time small to saturate the channel and the packet size are 128, 256, 512, 1024, 2048 bytes respectively.

### A. Topology I

For topology I.A, there is no confliction mentioned in Chapter 2. Throughput and delay under saturated channel is constrained by the backoff time after each successful transmission, physical and MAC headers. For this topology, *RTS/CTS* and backoff becomes harmful for the network performance since there is no risk of conflictions. As a result, throughput with *RTS* off always surpasses the one with *RTS* on as shown in Figure 14. Meanwhile, the packet delay with *RTS* off is also smaller than the one with *RTS* on which is saved by the time consumption on *RTS/CTS* exchanges. But the throughput is slightly smaller than the mathematical results. This is due to the backoff after each transmission as well as physical and MAC headers.

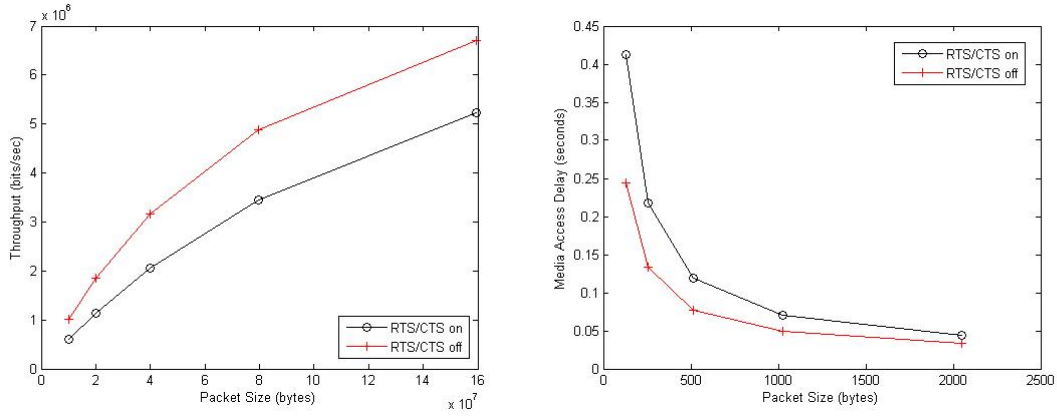



**Figure 14: Throughput (left) and delay (right) for topology I**

**B. Topology II.A**

For topology II.A, node 1 acts as a transmit initiator for node 2 as well as a receiver for node 0. When node 1 is in communication with node 0, since the communication initiator is itself, no communication will happen between node 1 and node 2 and hence there is no risk of conflictions. However, when node 1 is in communication with node 2, if node 0 does not know the ongoing communication and tries to transmit a packet to node 1, then collision happens if node 1 is receiving responding packet from node 2 and confliction case 3 happens if node 1 is transmitting packet to node 2.

However, for omni-directional antennas with symmetric transmission ranges, node 0 can always hear packets from node 1 destined to node 2. Upon receiving either *RTS* or *Data* packet from node 1 destined to node 2, node 0 updates its NAV duration and keeps silence till the end of the communication between node 1 and 2. Even the *RTS* is not turned on, it can still hear the *Data* packet from node 1, so *RTS/CTS* exchange is not necessary and in contrast harmful for the network performance. Same as topology I, throughput of *RTS* off always surpasses the one with *RTS* on as shown in Figure 15. Meanwhile, the packet delay with *RTS* off is also smaller than the one with *RTS* on.

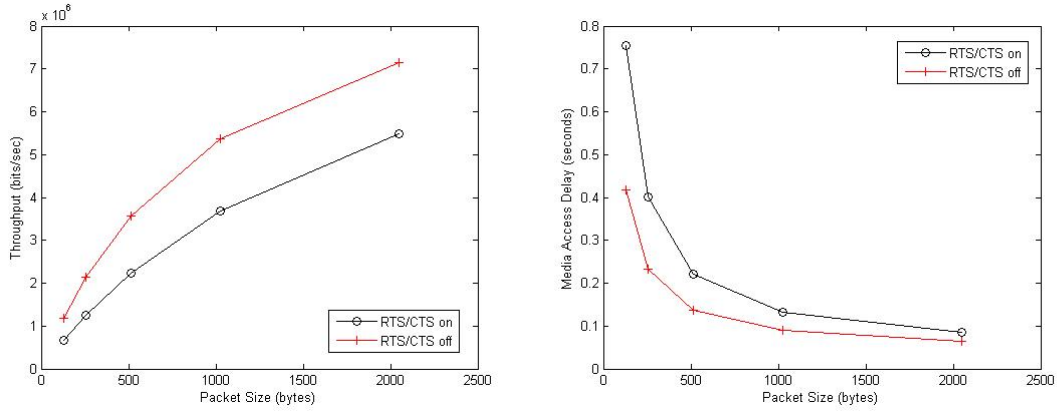



**Figure 15: Throughput (left) and delay (right) for topology II.A**

### C. Topology II.B

For topology II.B, node 1 receives packet from both node 0 and node 2. This is the well-known hidden problem. Since node 0 and 2 can not hear each other's packets, if *RTS* is off, the only packet node 0 or 2 can hear is the *Ack* packet from node 1 with NAV duration set to zero. Both physical and virtual carrier sensing indicate the channel is free and node 2 can transmit data packet to node 1 at any time. Collisions can happen the time from node 0 transmits the data packet to node 1 till node 1 receives the data. However, if *RTS* is on, node 2 can hear the *CTS* from node 1 with the NAV duration set to the end of the communication and node 2 will keep silence until the end of transmission. And there is no risk of collisions.

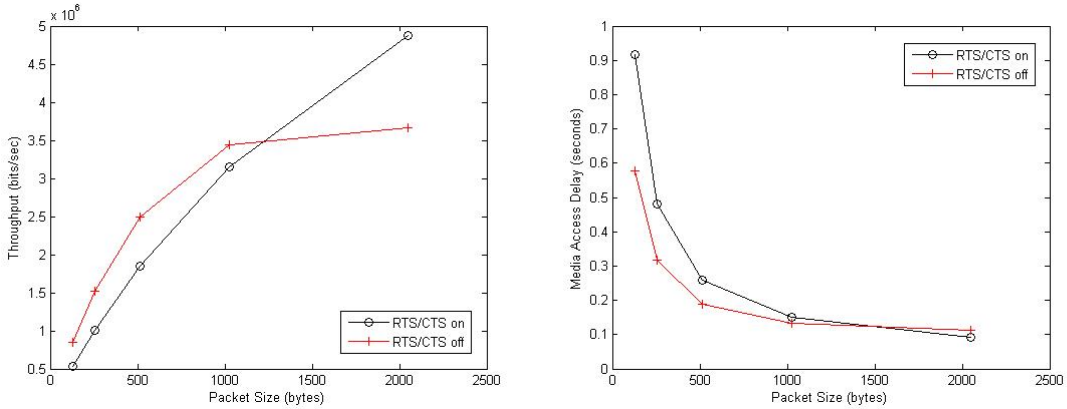



**Figure 16: Throughput (left) and delay (right) for topology II.B**

From the simulation results shown in Figure 16, we can see that when the packet size is smaller than 1024 bytes, both throughput and delay with *RTS* off outperform the ones with *RTS* on. However when the packet size exceeds 1024 bytes, bandwidth lost at data packet collisions surpasses the bandwidth saved by *RTS/CTS* overhead. The throughput and delay with *RTS on* outperforms the one with *RTS off*.

### 3.3 Chapter Summary

In this chapter, we firstly introduce different mechanisms in 802.11 DCF including DCF timing, CS, backoff as well as *RTS/CTS*. We can see how these different mechanisms work together to ensure that a sequence of control and data packets exchanges is not interrupted by other adjacent transmissions. Then we argue the effectiveness of *RTS/CTS* handshake with the intention of increasing network capacity by reducing unnecessary network overhead. By observations from both theoretical and simulation results, we suggest that *RTS/CTS* function should only be turned on when the data packet size is large or the network nodal density is high.

# Chapter 4: OPNET simulation

In this chapter, we introduce fundamental knowledge for OPNET [41] simulation including modeling mechanisms, wireless pipelines as well as details of 802.11 model in OPNET.

## 4.1 OPNET modeling mechanisms

To specifically model behaviors of a network at different levels, OPNET divides a network into three domains: *network domain*, *node domain* and *process domain*. As shown in Figure 17, the network domain concerns *nodes* and *links*. The node domain concerns internal node behavior consisting of modules and wires connecting different modules. And the process domain concerns the behaviors of the process defined by Proto C represented by a finite state machine.

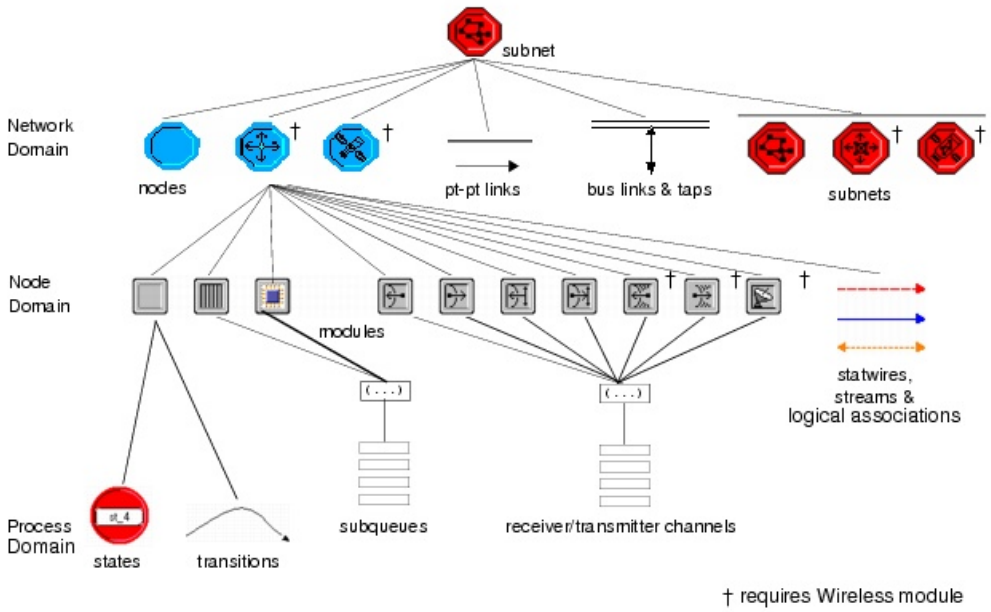



**Figure 17: OPNET domain hierarchy**

### 4.1.1 Network domain

A network consists of a number of subnetworks, nodes and links between nodes. When setting up a network, several issues should be considered such as network topologies, traffic flows and terrain models. For wireless communication, the links between wireless nodes are always radio channels as shown in Figure 18. The details of wireless pipeline stages are in Section 4.1.2.

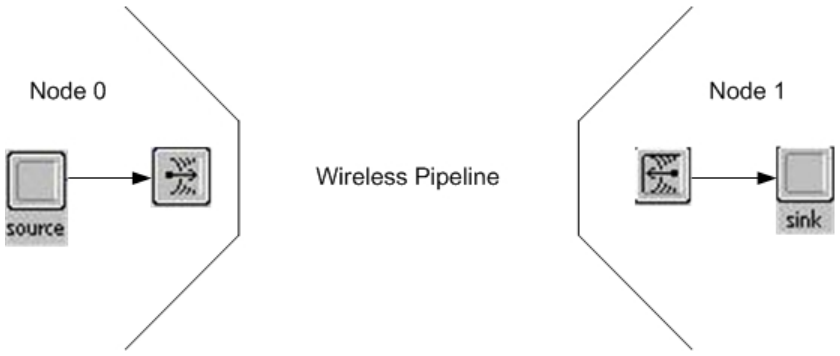



**Figure 18: Wireless pipeline**

OPNET supports two physical coordinate systems: *global coordinate system* and *geocentric coordinate system*. The global coordinate system describes the position of a node using *latitude*, *longitude*, and *altitude*. As shown in Figure 19, the geocentric coordinate system is a 3-dimensional Cartesian with the origin at the center of the earth. The three axes (x, y, z) are defined by a vector starting from the center of the earth to the intersection point on the surface. Given a unique node id, a node position for both global and geocentric coordinate system can be obtained by calling kernel procedure *op_ima_obj_pos_get ()*.The relation between global coordinate system and right-sphere geocentric coordinate system is shown below.

$$R = altitude + earth's\_radius$$
$$X = R \bullet \cos(latitude) \bullet \cos(longitude)$$
$$Y = R \bullet \cos(latitude) \bullet \sin(longtidue)$$
$$Z = R \bullet \sin(latitude)$$

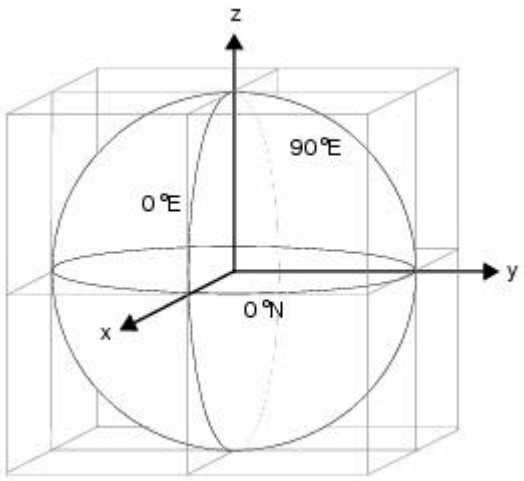



**Figure 19: Coordinate system**

### 4.1.2 Node domain

As shown in Figure 20, in OPNET, a node model is composed of a set of connected blocks called modules. Each module has specific behaviors defined by the process model residing in it. OPNET also predefines link modules for different links as the interface between node modules and the communication medium between different nodes. For wireless communication, those predefined modules are radio transmitter and receiver.

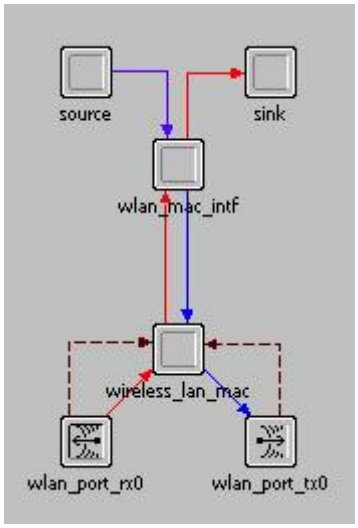



**Figure 20: Node domain**

There are two types of connection wires between modules which are stream and statistic wires that are presented by solid line and dashed line respectively. Usually a module has several input and output stream or statistic wires. To distinguish different wires, each input and output wire for a module is assigned a unique wire index as shown in Figure 21.

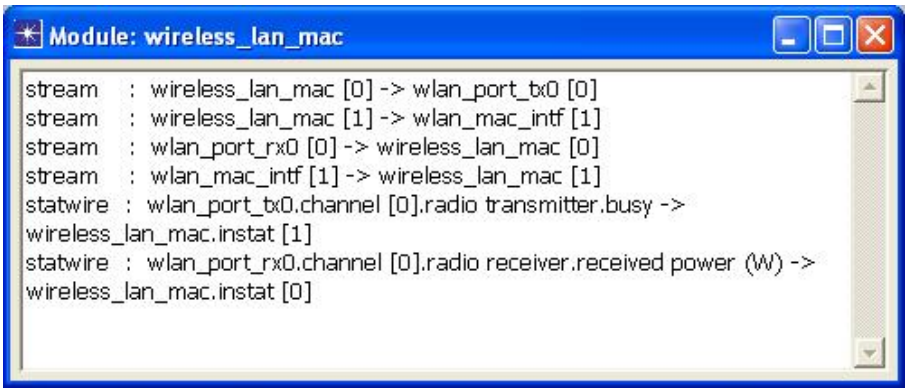



**Figure 21: Statistics index**

**Antenna pattern**

Antenna modules specify the antenna properties for radio transmitters and receivers modules. An antenna module can be connected to multiple radio transmitters and radio receivers, however, a radio transmitter or receiver can only connect to one antenna module only. As shown in Figure 22, the connection on the right is permitted while the left one is not. If there is no antenna module connected to a radio module, OPNET assumes the antenna is an omni-directional antenna. And to speed up the simulation, OPNET sets the *tagain* and *ragain* stages to *None* in pipeline. So if a directional antenna module is used, it is essential to set both *tagain* and *ragain* to *dra_tagain* and *dra_ragain* respectively.

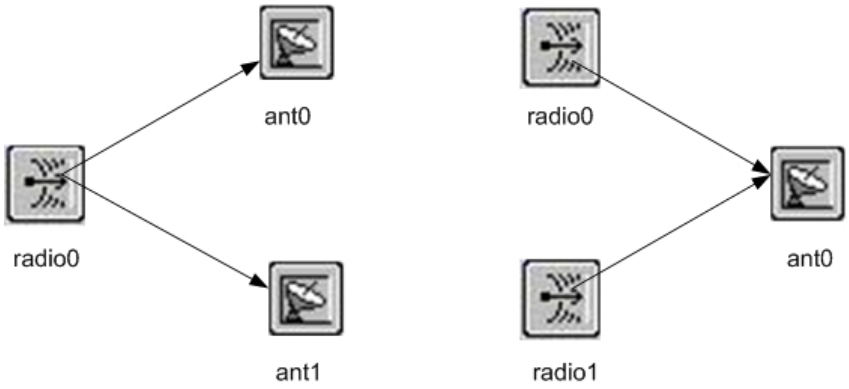



**Figure 22: Antenna module sharing**

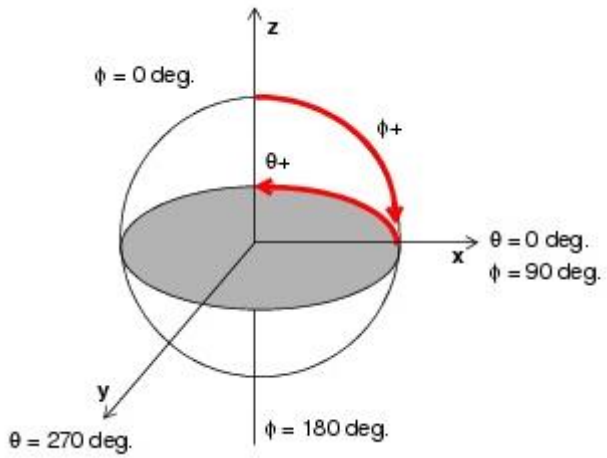



**Figure 23: Antenna coordinate system (φ [0, 180] θ [0, 360])**

To map gain to all directions in a three-dimensional space, OPNET represents it as a function of two angle variables φ (phi) and θ (theta) as shown in Figure 23. A user defined antenna pattern can be created by either *Antenna Pattern Editor* or *External Model Access* (Ema) packet. Both of them create a gain table for the antenna pattern at different directions specified by combination of phi and theta. Figure 24 shows examples of an omni-directional antenna on the left and a directional antenna on the right. Besides the gain table, there are two sets of attributes for an antenna module which are *target location* and *pointing reference*. Target location specified by latitude, longitude and altitude is the space location of destination node. Pointing reference specified by phi and theta is the point on the antenna pattern that is pointed at the target[5]. The gain table, target location and pointing reference together will be used at tagain and ragain pipeline stages for path attenuation calculation.

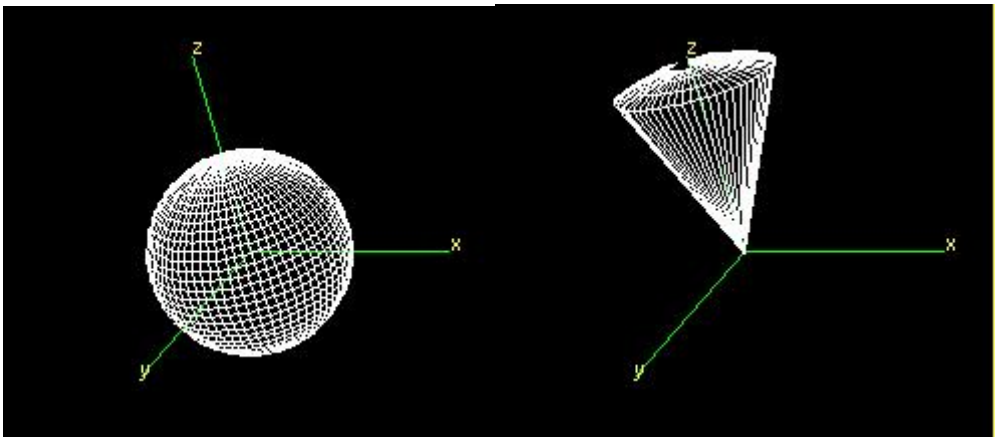



**Figure 24: Omni-directional and directional antennas**

[5] For directional antennas, it is usually the antenna's boresight.

### 4.1.3 Process domain

**A. Event-driven and interrupts**

OPNET is an event-driven simulation software. The simulation kernel keeps an event list containing a unique *event id*, *time* when the event is ready and an *event handle*. When an event is delivered to a process within a specified module it is destined for, we call it an interrupt for that process. The process is invoked by that interrupt and allowed to take some action in response to the interrupt, and then it is blocked again waiting for another interrupt. There are three important interrupts in OPNET which are *stream interrupts*, *statistic interrupts* and *self interrupts* respectively. And the type of interrupt can be obtained by calling kernel procedure *op_intrpt_type ()*.

- **Stream interrupt**

Packet streams are connections that carry data packets from a source module to a destination module. There are two types of packet: formatted packet and unformatted packet. The difference between unformatted packets and formatted packets is that fields in unformatted packet are specified by field index number while for formatted packet they are specified by field name. A packet can be created and destroyed by calling kernel procedure *op_pk_create()* and *op_pk_destroy()*. When a packet arrives on an input packet stream, an stream interrupt is invoked and the packet pointer can be obtained by calling kernel procedure *op_pk_get()*. A packet can be forwarded to an output packet stream by calling kernel procedure *op_pk_send()*. The kernel procedure *op_pk_stamp ()* stores current module and simulation time in specified packets. It is useful to calculate packet delay and locate the packet source object id. The module and time values of a stamp can be accessed at the destination node by calling kernel procedure *op_pk_stamp_mod_get ()* and *op_pk_stamp_time_get ()*.

- **Statistic interrupt**

Statistic wires carry the statistic from a source module to a destination module. The destination module can access the current value of the specified particular statistic by calling kernel procedure *op_stat_local_read ()*. There are also six statistic trigger attributes for statistic wires as shown in Figure 25, which are *rising edge*, *falling edge*, *repeated value*, *zero crossing*, *low threshold* and *high threshold*. When the source module statistic is changed, statistic interrupts can be triggered at destination module according to different configuration of these six trigger attributes.

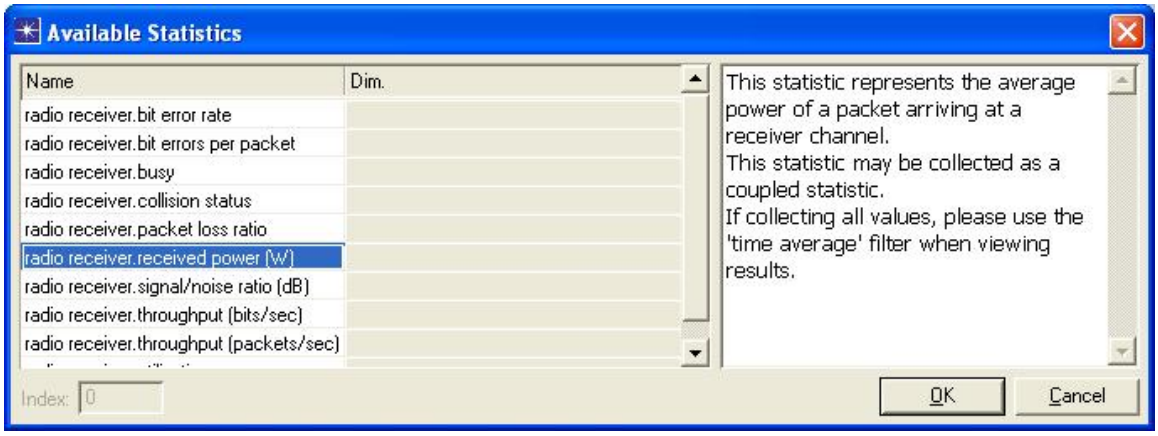



**Figure 25: Receiver statistics collection**

For wireless MAC protocol study, statistic wires are usually used to indicate the end of a packet transmission at the radio transmitter module or to monitor the channel status. Note that both the packet reception at receiver and packet transmission at transmitter are not instant. For packet transmission, we choose *radio transmitter.busy* as the source statistic and enable *falling edge trigger* as shown in Figure 26, so a static interrupt will be triggered indicating the completion of a packet transmission.

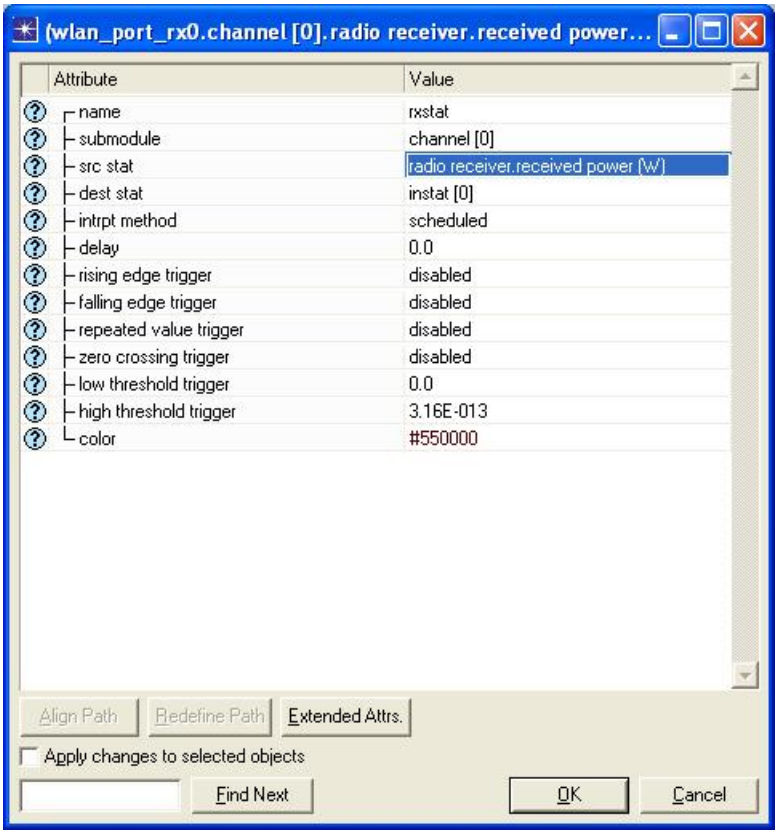



**Figure 26: Receiver interrupt trigger setting**

In CSMA based MAC protocols, the receiver module has two functions. The first function is to monitor the channel status and the second one is for packet receptions. For packet reception, we choose *radio receiver.received power* as the source static and enable *low and high threshold trigger* as shown in Figure 26, so a static interrupt will be triggered at the start or end of a packet reception. As shown in Figure 27, a static interrupt is triggered when a packet arrives at the receiver, and a static and a stream interrupt are triggered at the same simulation time indicating the end of packet reception. By monitoring the received channel status, we can get two important parameters, *receiver_busy* and *packet_collided*. The *receiver_busy* flag indicates whether the channel is free or busy and *packet_collided* flag indicates whether two or more packets collide at receiver, so that we can correctly calculate network throughput. The pseudo code of these two flags depending on the receiver statistics is shown below.

```
static void rcv_status_update (void)
        {
        FIN (rcv_status_update (void));

        /* new packet arrived */
        if (op_stat_local_read (RX_STAT) > rx_power_threshold)
                {
                /* collision since the receiver is already busy*/
                if (flags->receiver_busy)
                        flags->collided_packet = OPC_TRUE;
                else
                /* set the receiver_busy flag true */
                        flags->receiver_busy = OPC_TRUE;
                }

        /* end of a packet reception*/
        else if (flags->receiver_busy)
                flags->receiver_busy = OPC_FALSE;

        FOUT;
        }
```

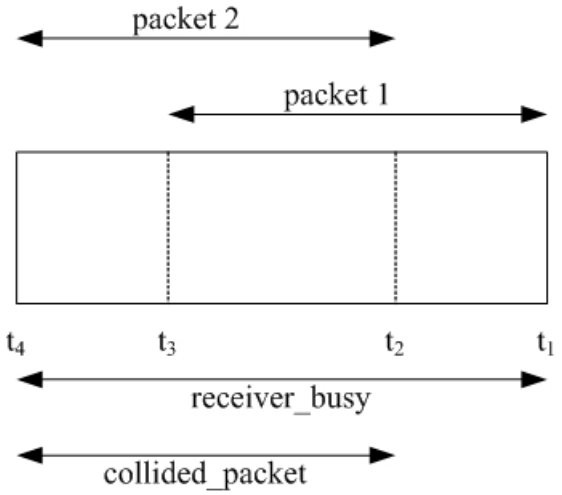



**Figure 27: Packet overlapping**

- **Self interrupt**

Different from stream or statistic interrupt which is generated by processes from other modules. Self interrupt schedules an interrupt to the module itself at a user defined time by calling the kernel procedure *op_intrpt_schedule_self (time, code)*. The parameter *code* is a user defined integer to distinguish different self interrupts for the same module and it can be obtained by calling kernel procedure *op_intrpt_code()*. The self interrupt can also be cancelled by calling kernel procedure *op_intrpt_cancel()*.

The typical use of self interrupt in MAC protocols is to implement *backoff* waiting and *packet timeout* indication. For example, a node needs to schedule a self interrupt for a packet timeout time after sending a data packet. If the node receives the acknowledgement before timeout time expires, it cancels the self interrupt. Else if that data packet is not acknowledged, an interrupt is delivered to the node indicating packet timeout.

**B. FSM**

Process model is the lowest level domain that specifies the behavior of processor module within a node domain. One feature that OPNET distinguishes from other simulators is that OPNET uses finite state machine (FSM) to represent the process model. A process is divided into several states connected by transitions. One advantage is that it limits the number of interrupts that could be delivered to a state and makes the flow clear. Actually we can merge all the states into one state with a default transition

delivered back to it. In this case, all we need to do is to predict all the possible interrupts and put all reactions for different interrupts in the exit execution.


**Figure 28: Unforced and forced states**

State indicates that a process has accumulated over a period. For event-driven simulator, the process suspends in the same state until an interrupt invokes the process. At any time, a process is always and only in one state. A state is split into two executives named enter executive and exit executive. As shown in Figure 28, there are two types of states called unforced state and forced state. The difference between forced state and unforced state is that unforced state allows a pause between enter executive and exit executive while forced state does not. Unforced states can represent true states of system while forced state is usually used to make the process flow clear.

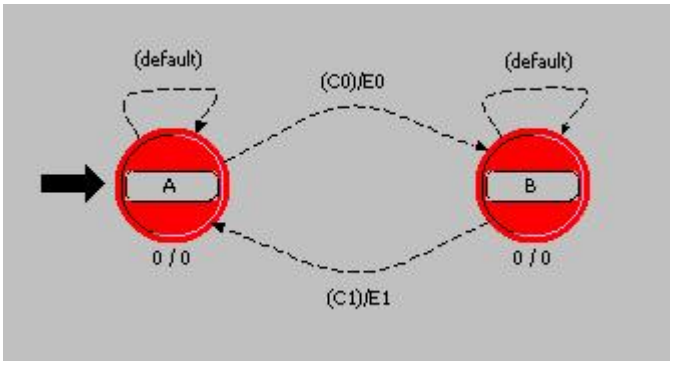



**Figure 29: State flow**

The process flow is shown in Figure 29 for a simple FSM. Assuming the process is suspended at state A and an interrupt happens, the interrupt invokes the process. The process takes the actions in exit executive of state A, since there are two output transitions for state A, if transition condition is C0, the process takes the action E0 specified by transition, moves to state B, takes the actions in enter executive of state B and blocked at state B waiting for the next interrupt. Otherwise, the process moves back to state A directed by transition default, takes the actions in enter executive of state A and blocked at state A. So the actions flow that a process takes after an interrupt is shown below:

invoked by an interrupt -> exit executive -> transition action (if specified) -> enter executive -> blocked waiting for another interrupt

Usually there are a number of interrupts that could happen for a specified state. The actions are Proto C code indicating what we want to do depending on different possible interrupts. The transition conditions are usually interrupt type, variables or Boolean flags changed by actions depending on different interrupts.

### 4.2 OPNET Pipeline

OPNET has different pipelines to model signal propagations between transmitters and receivers for different physical mediums such as *bus* or *wireless*. There are two types of wireless pipelines which are *default radio (dra)* pipeline and *wireless LAN (wlan)* pipeline respectively. And C codes with suffix *ps.c* for *dra* and *wlan* pipeline are in directory <*opnet_dir*>*models**wireless* and <*opnet_dir*>*models**wireless_lan*.

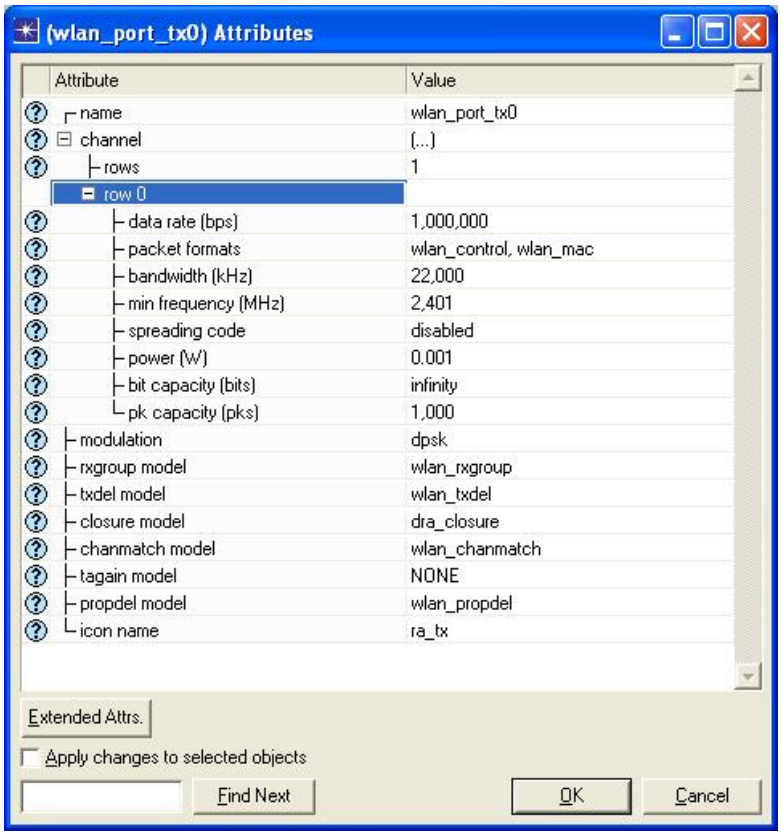



**Figure 30: Radio transmitter attributes**

There are several important channel attributes at *radio transmitter* and *radio receiver* modules which will be used at pipeline stages as shown in Figure 30. Some of them are listed in Table 2.

| Attribute name | Annotation |
|---|---|
| Rows | Number of channels |
| Bandwidth (Hz) | 22 M default |
| Min frequency (Hz) | 2.4 G for wlan |
| Power (W) | Transmit power (transmitter module only) |
| Spreading code | DSSS, OFDM, FHSS, IR |
| Data rate (bps) | 1 M default |
| Noise figure | 1.0 default used in pipeline stage 9 for background noise (receiver module only) |
| Processing gain | Used in pipeline stage 11 for bit error rate (receiver module only) |
| Modulation | Used in pipeline stage 11 for bit error rate (receiver module only) |
| Ecc threshold | 0.0 default used in stage 13 (receiver module only) |

**Table 2: Radio module attributes**

As shown in Figure 31, there are 14 stages (0 - 13) for the wireless pipeline. Stages 0 – 5 are associated with *radio transmitters* while stages 6 – 13 are computed at *radio receivers*. During these 14 stages, different signal propagation information are calculated and stored in different *transmission data attributes (TDA)* for further study. Most importantly, at the final stage, the simulation kernel decides whether a packet is successfully received based on the obtained *TDAs*. Next, we briefly introduce how the 14 pipeline stages affect the packet delivery decisions.

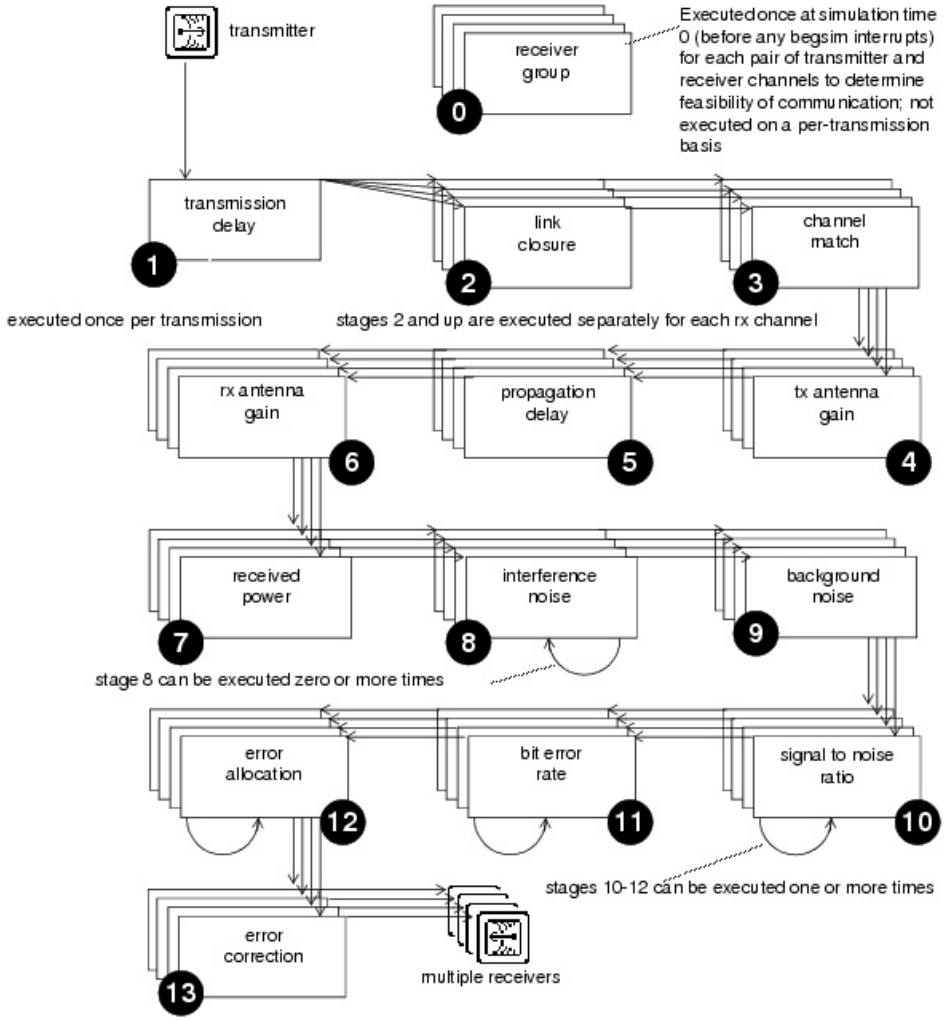



**Figure 31: 14 pipeline stages in OPNET**

- **Stage 0: Receiver group**

At the beginning of transmission, each transmitter calculates its receiver group based on *three* criterions listed below. Given a transmitter, this stage returns *OPC_TRUE* or *OPC_FALSE* for different receivers indicating whether the receiver is a valid receiver for that transmitter. Invalid receivers will not be involved in later stage calculations to optimise simulation speed.

- Frequency disjunction: According to *bandwidth* and *min frequency* attributes, if the frequencies at transmitter and receiver do not overlap, then the transmitting frame is either valid or noise.
- Physical separation: Whether there are obstacles between the transmitter and receiver.
- Antenna nulls: When directional antennas are used, the receivers in the directions where antenna nulls point to are not considered as valid receivers.

One thing we should note is that this stage does not take transmission power into consideration. In other word, a receiver that fulfills the criterions mentioned above is considered as a valid receiver even if the transmitter cannot reach the receiver depending on its transmission power. So in later section, there is a *destination address* attribute in the 802.11 model which is a random MAC address obtained from a list of valid receivers. However, we have to be very careful, so that the random destination receiver is in the transmission range of a transmitter.

- **Stage 1: Transmission delay**

This stage calculates the amount of time required for a packet to complete transmission as follows.

$$tx\_delay = pklen / tx\_drate \quad (1)$$

Where *pklen* stands for the *packet length* and *tx_drate* is the channel *data rate* shown in Table 2. And this result is placed in packet *OPC_TDA_RA_TX_DELAY* TDA.

- **Stage 2: Link closure**

This stage decides whether the signal from transmitter can physically affects a receiver for a given transmission path. It has three basic modes:

1. The transmission path will never be occluded.
2. Light of sight (LOS) model
3. Terrain model module (TMM) if user defines

The Boolean result is placed in packet *OPC_TDA_RA_CLOSURE* transmission data attribute.

- **Stage 3: Channel match**

This stage classifies the transmission into the following three categories with respect of different receiver channels according to the *bandwidth* and *min frequency* attributes in Table 1.

  - **Valid**: If the transmitter channel and receiver channel are *totally matched*, then the packet is considered as a valid packet that will be successfully received at the receiver.

  - **Noise**: If the transmitter channel and receiver channel are only *partly overlapped*, then the packet is considered as a noise packet that will interfere with other packets reception at the receiver.

- **Ignored**: If the transmitter channel and receiver channel are *not matched* at all, then the packet is considered as an ignored packet that will not have any affect at the receiver. In this case, the following stages are skipped to optimise the simulation speed.

At the end of this stage, simulation kernel places three constant values *OPC_TDA_RA_MATCH_VALID*, *OPC_TDA_RA_MATCH_NOISE* and *OPC_TDA_RA_MATCH_IGNORE* corresponding to three categories mentioned above respectively in the packet *OPC_TDA_RA_MATCH_STATUS* TDA.

- **Stage 4 and 6: Transmit and receiver antenna gain**

These two stage calculate the transmit antenna gain for directional antennas. If it is an isotropic antenna, these stages are skipped. An antenna pattern can be established by using either *Antenna Pattern Editor* or *EMA*, so that simulation kernel will have an antenna gain table on 3-dimonsional space. Simulation kernel gets the transmit gain by looking up the gain table specified by *pointing ref. phi* and *pointing ref. theta* as shown in Figure 32. The distance between transmitter and receiver is indicated by *target latitude*, *target longitude* and *target altitude* attributes. And all these information will be used at stage 7 for received power.

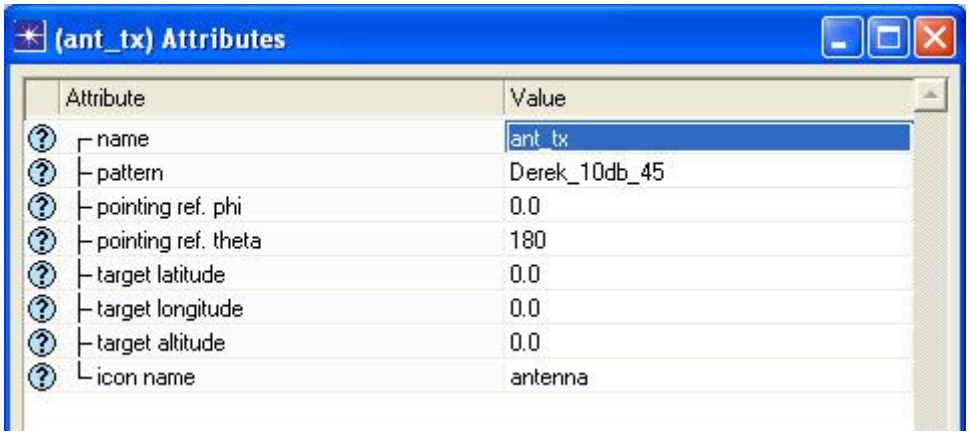



**Figure 32: Antenna pattern and attributes**

- **Stage 5: Propagation delay**

This stage calculates the propagation delay for the packet signal traveling from transmitter to receiver which is the distance between transmitter and receiver divided by the propagation velocity of radio signal (3.0E + 08). For mobile nodes, the distance between transmitter and receiver can vary during the packet propagation, so the simulation kernel calculates *start_prop_delay* and *end_prop_delay* and places them in packet *OPC_TDA_RA_START_PROPDEL* and *OPC_TDA_RA_END_PROPDEL* TDA respectively.

- **Stage 7: Received power**

This stage calculates the received power for either *valid* packet or *noise* packet (in Watts). The received power is a key factor to determine whether the signal is powerful enough for the receiver to successfully capture the receiving packet and it is calculated based on parameters such as the *transmission power*, the *propagation distance*, transmitter and receiver *antenna gain*, or the *transmission frequency*. The calculation for LoS is shown below.

$$tx\_center\_freq = tx\_base\_freq + (tx\_bandwidth / 2.0) \quad (1)$$

$$\lambda = C / tx\_center\_freq$$

$$path\_loss = \left( \frac{\lambda}{4\pi \bullet prop\_dis\tan ce} \right)^2 \quad (2)$$

$$ant\_gain = 10^{\frac{gain}{10}} \quad (3)$$

$$rcvd\_power = tx\_power * tx\_ant\_gain * path\_loss * rx\_ant\_gain \quad (4)$$

Where $C$ is 3.0E + 08, $\lambda$ for 2.4 GHz is about 0.125 and *tx_base_freq* and *tx_bandwidth* are *min frequency* and *bandwidth* attributes in Table 1 respectively. Finally, at the end of this stage, the received power is placed in packet *OPC_TDA_RA_RCVD_POWER* TDA. The transmission range for different transmission power under LoS model is shown in Figure 33.

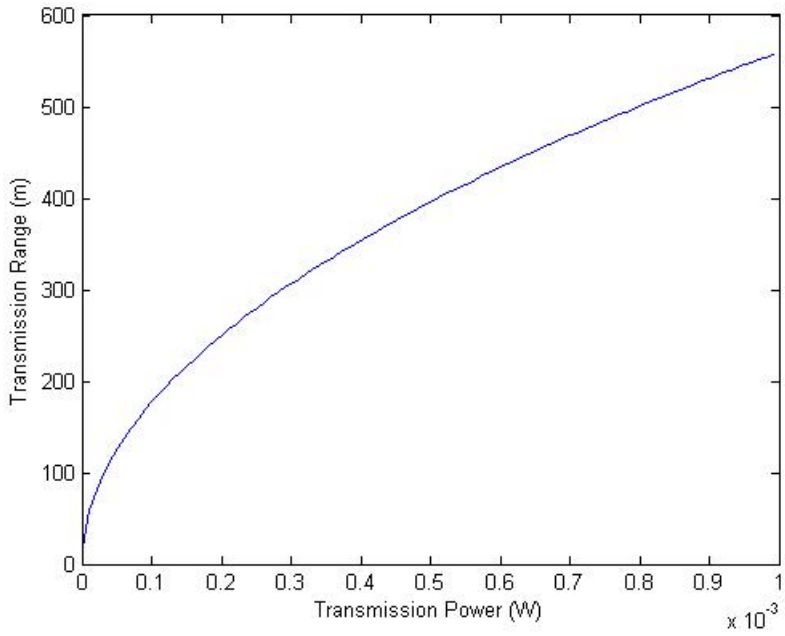



**Figure 33: Transmission range under LoS propagation model**

- **Stage 8: Interference noise**

This stage calculates the interference which occurs when two or more transmissions arrive at a same receiver channel concurrently. Generally, there are two circumstances here: the valid packet arrives at the receiver while another packet is already being received or the packet is being received while other packets (either valid or noise) arrive. At the end of this stage, the simulation kernel places the accumulative power of valid packet[6] *OPC_TDA_RA_NOISE_ACCUM* TDA.

- **Stage 9: Background noise**

This stage accounts for background noise other than noises from other transmitters in stage 8. These sources can be galactic, thermal, or urban noise. The calculation is shown below.

```
/* Calculate effective receiver temperature. */
rx_temp = (rx_noisefig - 1.0) * 290.0;

/* Calculate in-band noise from both background and thermal sources. */
bkg_noise = (rx_temp + bkg_temp) * rx_bw * BOLTZMANN;

/* Calculate in-band ambient noise. */
amb_noise = rx_bw * AMB_NOISE_LEVEL;
```

Where default noise Figure *rx_noisefig* is 1.0, default *bkg_temp* is 290K, the value for *BOLTZMANN* and *AMB_NOISE_LEVEL* are 1.379E – 23 and 1.0E – 26 respectively

[6] There is no need to calculate interference noise for noise packets.

and *rx_bw* stands for the receiver bandwidth. At the end of this stage, the sum of *bkg_noise* and *amb_noise* is placed in packet *OPC_TDA_RA_BKGNOISE* TDA.

- **Stage 10: Signal to noise ratio**

This stage calculates the average power SNR for arriving packet. The calculation is based on the TDA value obtained at earlier pipeline stage including *received power (rcvd_power)*, *interference noise* (*inoise*) and *background noise (bkgnoise)*. The calculation is shown below.

$$SNR = 10\log_{10}\left(\frac{rcvd\_power}{bkgnoise + inoise}\right) \qquad (1)$$

And the result is placed in packet *OPC_TDA_RA_SNR* transmission data attribute.

- **Stage 11: Bit error rate**

This stage derives the probability of bit errors during the past interval of constant BER. This calculation is based on the *SNR* obtained in stage 10 and *processing gain* defined at receiver attribute in Table 2. To precisely model BER, the BER is not evaluated on per packet basis but in each packet segment divided by the BER changing points. The simulation kernel first calculates effective SNR by adding SNR (in dB) and processing gain (*proc_gain* in dB) together, and then the BER is derived from effective SNR (*eff_snr*) based on a modulation curve assigned to the receiver. The calculation is shown below.

```
/* Calculate effective SNR incorporating processing gain. */
eff_snr = snr + proc_gain;
```

```
/* Derive expected BER from effective SNR. */
ber = op_tbl_mod_ber (modulation_table, eff_snr);
```

At the end of stage, the BER value is placed in packet *OPC_TDA_RA_BER* TDA.

- **Stage 12: Error location**

This stage calculates the number of bit errors in a packet segment based on the BER obtained in stage 11. One accurate method is to test the packet segment bit per bit, and compare the BER to a random number *r* derived uniformly from 0 to 1. If the BER is greater than *r*, this bit is considered as an error bit. Otherwise, this bit is considered as a correct bit. The algorithm is shown below.

```
/* test each bit in the segment */
for (i = 0; i < num_bits ; i++)
        {
        /* obtain a uniformly distributed random */
        /* number between 0.0 and 1.0          */
        r = random_number (0.0, 1.0);

        /* compare the number against the probability of bit error */
        /* to evaluate the correctness of the bit              */
        if (r < bit_err_rate)
                {
                /* augment the bit error accumulator */
```

```
            err_accum++;
            }
        }
```

At the end of this stage, simulation kernel places the number of errors in packet *OPC_TDA_RA_NUM_ERRORS* transmission data attribute and the BER which is the number of errors divided by packet length in *OPC_TDA_RA_ACTUAL_BER* TDA respectively.

- **Stage 13: Error correction**

This stage makes the decision that whether a packet can be successfully accepted by the receiver based on the BER obtained from earlier stages. It can be expressed as following.

*((((double) num_errs) / pklen) <= ecc_thresh) ? OPC_TRUE : OPC_FALSE;*

Where *ecc_thresh* is obtained as the receiver attribute in Table 2.

This Boolean value is placed in packet *OPC_TDA_RA_PK_ACCEPT* TDA.

### 4.3 802.11 Model

**A. Possible interrupts**

The 802.11 node model in OPNET is shown in Figure 34. The *source* module is a traffic generator which can be either OPNET built-in model such as *simple_source* and *bursty_source*, or the traffic model built by yourself. The *wlan_mac_intf* module registers its process as "*arp*", captures the packet from *source*, gets a random destination, attaches the destination address to the packet and sends the packet to *wireless_lan_mac*. The *wlan_port_rx0* module receives packets from the wireless medium and forwards packets to module *wireless_lan_mac*. Meanwhile, *wlan_port_rx0* module also delivers statistic of wireless medium to *wireless_lan_mac* as a function of CCA as mentioned in Section 4.1. The *wlan_port_tx0* module gets packets from *wireless_lan_mac* and sends it to wireless medium. Meanwhile, it generates a statistic interrupt indicating the completion of packet transmission as mentioned in Section 4.1.

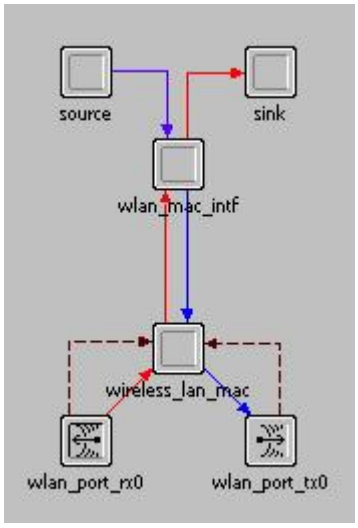



**Figure 34: 802.11 node model**

According to the node model, several types of interrupts could be delivered to the process in *wireless_lan_mac* module and they are listed in Table 3.

| Interrupts | Indication | Function to call | State |
|---|---|---|---|
| i. Stream interrupt from receiver | Packet received | *wlan_physical_layer_data_arrival ()* | Any |
| ii. Static interrupt from receiver | Start or end of a packet reception | *wlan_mac_rcv_channel_status_update ()* | Any |
| iii. Static interrupt from transmitter | Packet transmission completed | None | TRANSMIT |
| iv. Stream interrupt from *wlan_mac_intf* | Packet arrives from upper layer | *wlan_higher_layer_data_arrival ()* | Any |
| v. Self interrupt | 1. Deference finished (Interrupt Code: *WlanC_Deferenc e_Off*) | None | DEFER |
| | 2. Backoff finished | | BACKOFF |

| | | | |
|---|---|---|---|
| | or CW finished (Interrupt Code: *WlanC_Backoff_Elapsed* or *WlanC_CW_Elapsed*) | | |
| | 3. Packet timeout (Interrupt Code: *WlanC_Frame_Timeout*) | | WAIT_FOR_RESPONSE |
| | 4. Initialisation | | INIT |

**Table 3: Interrupts for 802.11 DCF**

**B. State actions and flow**

For 802.11 DCF, a node is always in one of the following five states: *idle*, *defer*, *backoff*, *transmit* or *wait for response* as shown in Figure 35. Next, we introduce actions and state flow for each state activated by different interrupts.

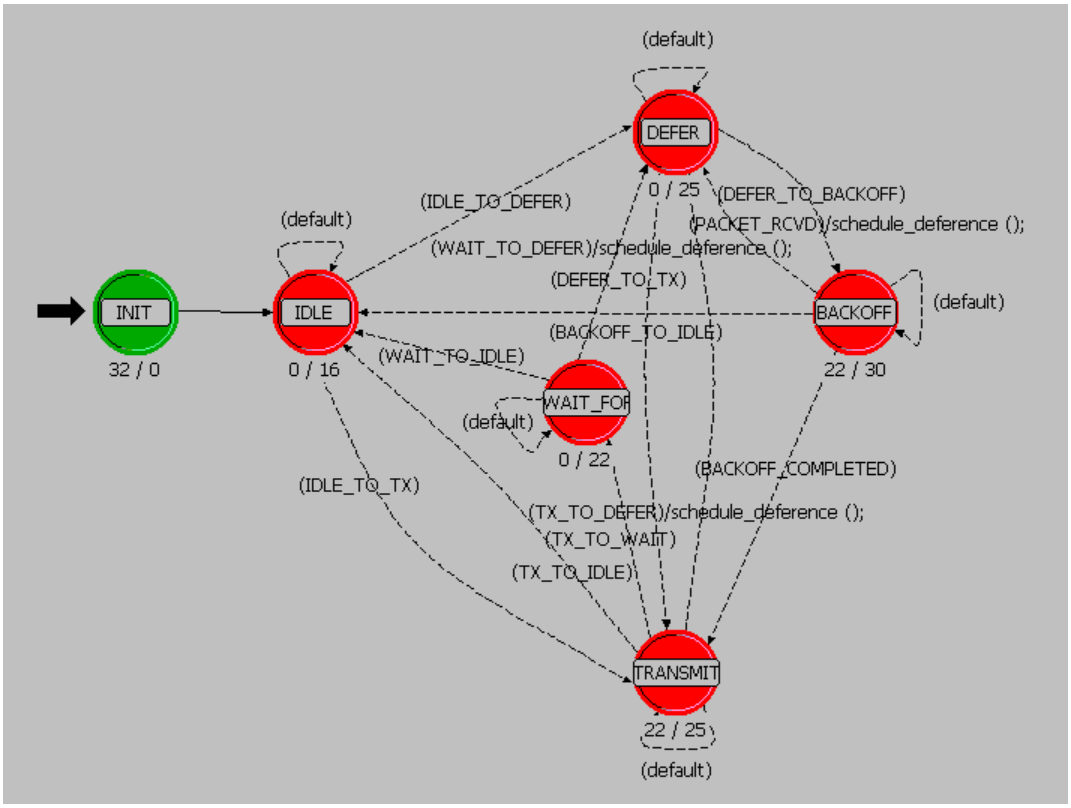



**Figure 35: State machine for 802.11 DCF**

**IDLE**

When a node has no packet backlogged at the upper stream, it is at idle state. Three possible interrupts could happen at state idle, and the actions and state flow for different interrupts are listed below:

1. A stream interrupt from *wlan_mac_intf* indicating a new packet is received from upper layer. In this case, if the channel is free for more than *aDIFSTme*, the process goes to state TRANSMIT, otherwise, it goes to state DEFER.
2. A stream interrupt indicating a *RTS* or *Data* packet is received from *wlan_port_rx0* which needs to be responded. In this case, if the channel is idle, the process goes to state TRANSMIT, otherwise, it goes to state DEFER.
3. A statistic interrupt from *wlan_port_rx0* indicating a beginning or end of packet reception. In this case, the process updates channel status flags and stays in state idle.

**DEFER**

When a node has packets to transmit but the channel is busy, it stays in state *defer*. This state is a most important state for CSMA/CA based protocols that implement *sense-before-send* function. Before the process moves into state DEFER, it schedules deference according to different situations by calling function *wlan_schedule_deference ()*. The scheduled deference for different packets is listed in Table 4 below:

| Cases | Deference duration |
|---|---|
| *fresp_to_send* is true indicating the packet is a respond packet | *aSIFSTime* |
| *retry_count* ≠ 0 indicating the packet is a retried packet | *aSIFSTime* |
| *wait_eifs_dur* is true | *current_time* + *aEIFSTime* |
| Normal *RTS* or *Data* packet | *nav_duration* + *aDIFSTime* |

**Table 4: Deference**

For this state, four possible interrupts could happen, and the actions and state flow for different interrupts are listed below:

1. A stream interrupt from *wlan_mac_intf* indicating a new packet is received from upper layer. In this case, the process inserts the packet into upper layer stream buffer for later access and stays in state DEFER.
2. A statistic interrupt from *wlan_port_rx0* happens indicating the channel becomes busy. In this case, the process cancels the deference and reschedules one until the channel becomes idle again.
3. A stream interrupt from *wlan_port_rx0* happens indicating a new packet is received from receiver. In this case, if the packet is destined to this node and needs to be

responded, then the process cancels the current transmission, schedules a new deference and stays in state DEFER. If the packet is not destined to this node, but the nav_duration is bigger than the node's current nav_duration, the process updates its nav_duration, reschedules deference and stays in state DEFER.

4. A self interrupt happens indicating the completion of deference. In this case, if the transmit packet is *Data* or *RTS*, the process goes to BACKOFF state. Otherwise, if the transmit packet is a response packet such as *CTS* or *Ack*, the process goes to TRANSMIT state directly, since respond packets do not need to backoff in 802.11.

**BACKOFF**

For 802.11, under two circumstances a process needs to go to backoff state. The first one happens when a node tries to initiate a new transmission and the channel is busy. The second one is after each successful transmission when its data packet is acknowledged. For BACKOFF state, three possible interrupts could happen, and the actions and state flow for different interrupts are listed below:

1. A stream interrupt from *wlan_mac_intf* indicating a new packet is received from upper layer. In this case, the process inserts the packet into upper layer stream buffer for later access and stays in state BACKOFF.
2. A statistic interrupt happens from *wlan_port_rx0* indicating the channel becomes busy. In this case, the process pauses the backoff by calling op_ev_cancel (*WlanC_Backoff_Elapsed* or *WlanC_CW_Elapsed*), computes the remaining time and moves back to state DEFER.
3. A self interrupt happens indicating the backoff is finished. In this case, the process moves to state TRANSMIT.

**TRANSMIT**

Since the packet transmission is not instant, state transmit represents the time duration for packet transmission. When the process arrives at TRANSMIT state, it calls function *wlan_frame_transmit ()* to decide which types of packet to transmit. It can be a new packet such as *Data* or *RTS*, or it is a packet to respond such as *Ack* or *CTS*, or it is a packet that needs to be retransmitted. *wlan_frame_transmit ()* then calls function *wlan_prepare_frame_to_send ()* to add packet header and nav duration in that packet and use *op_pk_send ()* to start the packet transmission.

For state TRANSMIT, three possible interrupts could happen, and the actions and state flow for different interrupts are listed below:

1. A stream interrupt happens from *wlan_mac_intf* indicating a new packet is received from upper layer. In this case, the process inserts the packet into upper layer stream buffer for later access and stays in state TRANSMIT.
2. A statistic interrupt happens from *wlan_port_rx0* indicating channel changes. In this case, the process stays in state TRANSMIT.
3. A stream interrupt happens from *wlan_port_rx0* indicating packet is received at receiver. In this case, the process marks the received packet as bad packet by setting the flag *rcvd_bad_packet* true and stays in state TRANSMIT.
4. A self interrupt happens indicating the packet transmission is finished. In this case, if the transmitted packet is a *RTS* or *Data* packet that needs to be responded, the process moves to state WAIT_FOR_RESPONSE. Otherwise, if there is more packets waiting in the upper layer buffer, the process moves to state DEFER to initiate a new transmission, else the process moves to state IDLE.

**WAIT_FOR_RESPONSE**

When a node has sent a packet that needs to be responded, it schedules a self interrupt for *round trip time (RTT)* by calling *op_intrpt_schedule_self (RTT, WlanC_Frame_Timeout)* and moves to state WAIT_FOR_RESPONSE. Four interrupts could happen in this state, and the actions and state flow for different interrupts are listed below:

1. A statistic interrupt happens from *wlan_port_rx0* indicating channel changes. In this case, the process stays in state WAIT_FOR_RESPONSE.
2. A stream interrupt happens from *wlan_mac_intf* indicating a new packet is received from upper layer. In this case, the process inserts the packet into upper layer stream buffer for later access and stays in state WAIT_FOR_RESPONSE.
3. A self interrupt happens indicating the responding packet is timeout. In this case, the process increases the retry counter and move to state DEFER for retransmission.
4. A stream interrupt happens from *wlan_port_rx0* indicating a packet is received at receiver. In this case, the actions and state flow are described as follows:

```
cancel self interrupt;
if (received packet is the one it is waiting for)
        {
        if (no more packets waiting in the upper layer buffer)
                go to state IDLE;
        else
                go to state DEFER;
        }
else
        {
        retry_counter++;
        if (the packet is not destined to me)
```

*update nav if needs to and go to state DEFER for retransmission;*

*else if (the packet is a packet that needs to be responded such as RTS or Data)*

*set respond_packet_type and go to state DEFER for response packet;*

*}*

### 4.4 Chapter Summary

In this chapter, we briefly introduce fundamental knowledge for OPNET simulation including modeling mechanisms, wireless pipelines as well as details of 802.11 model. In Chapter 5, we will calculate the adjusted power for transmission power control according to the knowledge of propagation model in Chapter 2 and pipeline stage in this chapter[7]. We also collect a new statistic - *power consumption*, which is calculated based on the *packet transmission power* and *packet transmission time*. In Chapter 6, we will modify the 802.11 model to implement directional antennas at the transmitter. We firstly connect a directional antenna module to the radio transmitter module - *wlan_port_tx0* in the 802.11 node model. Then we modify the NAV mechanism in 802.11 process model to directional NAV. Finally, we add beamform function to allow a directional antenna to arbitrarily point its main lobe to a neighbor destination.

[7] Especially in *antenna gain* and *received power* stages

# Chapter 5: Transmission Power Control

As aforementioned, spatial reuse increases the network performance by allowing more traffic to happen simultaneously. In this chapter, we study how spatial reuse [5, 6] achieved by transmission power control increases the network performances such as network throughput and power consumption.

Power is precious resource in wireless ad hoc networks since most of the mobile terminals are powered by batteries. Therefore, how to reduce power consumption to prolong the battery life becomes one of the major issues in wireless ad hoc networks. A wireless terminal is always in one of three statuses: packet transmission, packet reception or idle. Table 5 shows the power consumption using 2.4 GHz DSSS Lucent IEEE 802.11 WaveLAN PC Card in ad hoc mode with channel bit rate of 11Mbps under these three status. We can see that transmission consumes more power than reception and idle waiting.

| Transmission | Reception | Idle Waiting |
|---|---|---|
| 1364 mW | 900 mW | 739 mW |

**Table 5: Power consumptions under three statuses**

One way to conserve power in wireless networks is *power saving* mentioned in Chapter 2 which saves the power consumption wasted on unnecessary *idle* waiting. An alternative to reduce power consumption is *transmission power control* (TPC) which saves the transmission power. It is usually a per-packet power control which adjusts the power level during *RTS/CTS* handshake. In addition to power saving, TPC can also increase the network capacity by improving network spatial reuse. As shown in Figure

36, assume the default transmission power achieves a transmission range R and there are two pairs of transmission nodes lying in the network which are node A, B and node C, D respectively. If all nodes in the network use transmission range R, then only one pair of transmission can happen at any time slot. However, if we reduce their transmission to r which is just the distance from source node to destination node, then two pairs of transmissions can happen at the same time. This gives us the hint that by simply controlling the transmission range, we can dramatically increase the network capacity.

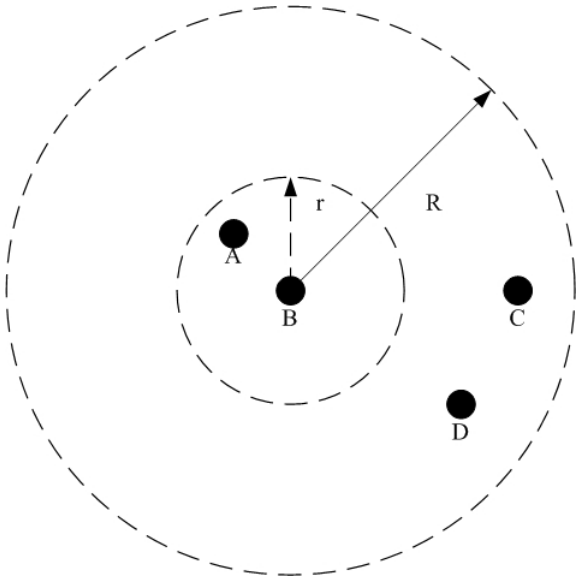



**Figure 36: Spatial reuse achieved by power control**

### 5.1 TPC protocols

In this section, we introduce two TPC schemes: basic TPC and spatial TPC. As we will see later, basic TPC can only save power consumptions whilst spatial TPC also exploits the benefits of spatial reuse to increase the network throughput.

**A. Basic TPC**

Before introducing TPC, we have following assumptions:

- The path is symmetric.
- The network topology is static.
- A maximum transmit power $P_{max}$ that a terminal can tune to is known to all the terminals in the network.
- All the terminals are capable of power adjust that is upon receiving a packet from terminal S with received power $P_r$, the terminal D can adjust its new transmit power to $P_t$ which is the a minimum power level to successfully reach terminal S. And the $P_t$ can be calculated as follows:

  $P_t = P_{max} - P_r + P_{reception} + \xi$,

  where $\xi$ is a power buffer to adapt the possible path change.

The handshake of basic TPC can be described as follows:

- If node S wants to send data packets to node D, it transmits a *RTS* at power level $P_{max}$.
- Upon receiving the *RTS* from node S, node D sends *CTS* back to node S at power level $P_{max}$.
- Upon receiving the *CTS* from node D, node S adjusts its power and sends D*ata* to node D at new power level $P_t$.
- Upon receiving the data from node S, node D adjusts its power and sends *Ack* back to node S at power level $P_t$.

We can see that since both of *RTS* and *CTS* are transmitted at the maximum power level, all the neighbors of both node S and D will keep silent during their transmission, basic

power control cannot benefit by spatial reuse to increase the network throughput. However, it reduces the power consumption since it transmits both *Data* and *Ack* packets at an adjusted minimum power level compared to 802.11. And the energy saved by basic power control can be calculated as follows:

$$J = \left( \frac{S_{Data} + S_{Ack}}{B_W} \right) \bullet \left( P_{\max} - P_t \right)$$

Where $S_{Data}$ and $S_{Ack}$ are the packet size of *Data* and *Ack*, and $B_W$ is the channel bandwidth.

**B. Spatial TPC and challenges**

To fully exploit the benefit of spatial reuse, we propose a novel TPC protocol named spatial TPC. Different from basic TPC, we transmit all the packets including *RTS, CTS, Data* and *Ack* at the minimum power level $P_t$. One challenge is that how to decide the $P_t$ for *RTS* when a node firstly initiates a transmission since there is no received power to reference. Our solution is to create a power reference table during neighbor discovery. It can be described as follows.

During neighbor discovery, each node sends *Hello* messages at the power level $P_{max}$. Upon receiving the *Hello* message, in addition to the neighbor node ID, we add two other parameters to the neighbor table which are the location information and the power level as shown in Table 6. When a node wants to initiate a transmission to a neighbor node, it looks up its neighbor table, finds the power level $P_t$, and transmits a *RTS* to its destination at power level $P_t$. In addition to neighbor discovery, this table can also be renewed when a node overhears packets destined to other nodes.

| Neighbor ID | Mac address |
|---|---|
| Location | $(d, \theta)$ |
| Power level | $P_t$ |

**Table 6: Neighbor information table**

The handshake of spatial TPC can be described as follows:

- If node S wants to send data packets to node D, it transmits a *RTS* at power level $P_t$.
- Upon receiving the *RTS* from node S, node D adjusts its power and sends *CTS* back to node S at new power level $P_t$.
- Upon receiving the *CTS* from node D, node S adjusts its power and sends *Data* to node D at power level $P_t$.
- Upon receiving the data from node S, node D adjusts its power and sends *Ack* back to node S at power level $P_t$.

The differences between 802.11, basic TPC and spatial TPC is shown in Figure 37.

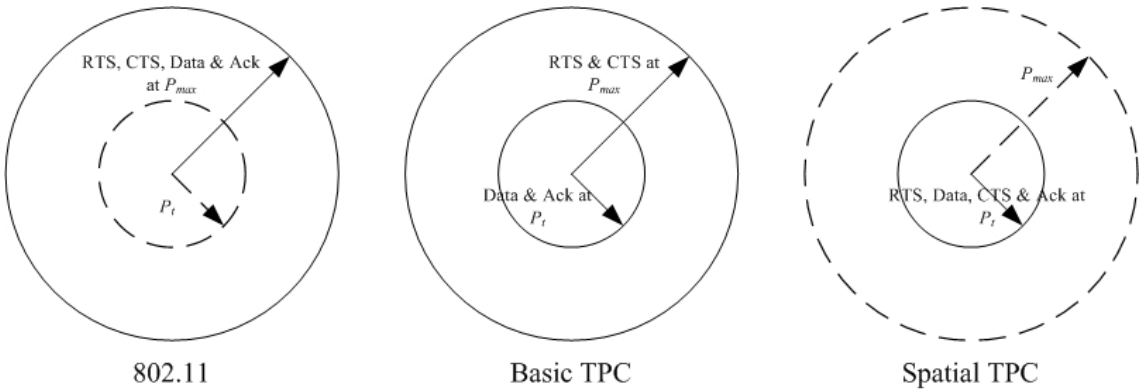



**Figure 37: Transmission ranges for three power control schemes**

However, for spatial TPC, different power levels generate different transmission ranges. The variable transmission ranges cause heterogeneous hidden terminal problems. The hidden terminal problem happens because the transmission of lower power cannot be detected by potential transmitters at higher power. As shown in Figure 38, node A is out of the interference range of node C, as a result, it can neither hear the *CTS* from node C nor sense the channel busy. If node A sends *RTS* or *DATA* packet to its destination node B when node C is receiving *Data* packet from node D, collision happens at node C. So we say a transmission at lower power is vulnerable to nearby transmission at higher power for variable radius network.

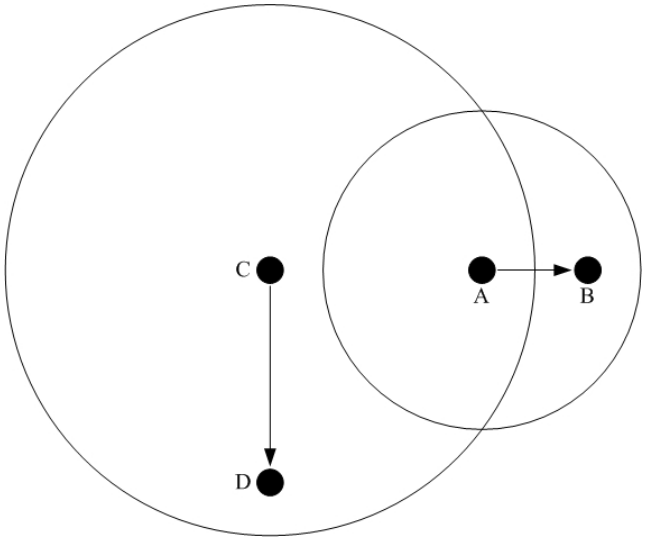



**Figure 38: New hidden problems caused by variable transmission ranges**

### 5.2 Simulation results

We now evaluate the performance of two power control schemes and compare them with the IEEE 802.11. Data packets are 512 bytes and the channel data rate is 11 MHz. The fixed maximum power $P_{max}$ is 1 mW which has a transmission range around 300m.

**A. Linear topology**

We first simulate a four-linear topology to highlight the advantage of spatial TPC. Four nodes reside linearly in the network and distances between nodes and traffic patterns are shown in Figure 39.

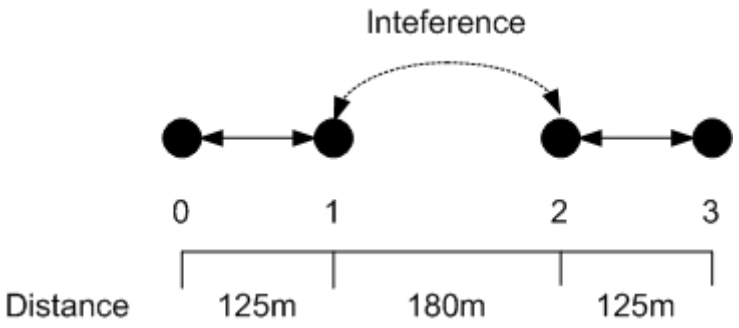



**Figure 39: Linear topology**

Simulation results for both network throughput and power consumption are shown in Figure 40. For 802.11, since all the packets are transmitted at $P_{max}$, it consumes more power than basic and spatial TPC. Meanwhile, only one communication happens at any time, so the throughput is about 2.2e+06 bits/sec. For basic TPC, since *Data* and *Ack* packets are transmitted using $P_t$, it consumes less power than 802.11. However, when one pair of nodes is in communication, the other pair of nodes overhears the *RTS* and *CTS* packets, so they will keep silence according to the NAV duration. As a result, only one transmission can happen at any time, and the throughput of basic TPC is the same as 802.11. However, for spatial TPC, all the packets are transmitted at $P_t$ and the two pair of nodes cannot hear packets from each other. As a result, spatial TPC doubles the throughput and consumes the least power among the three schemes.

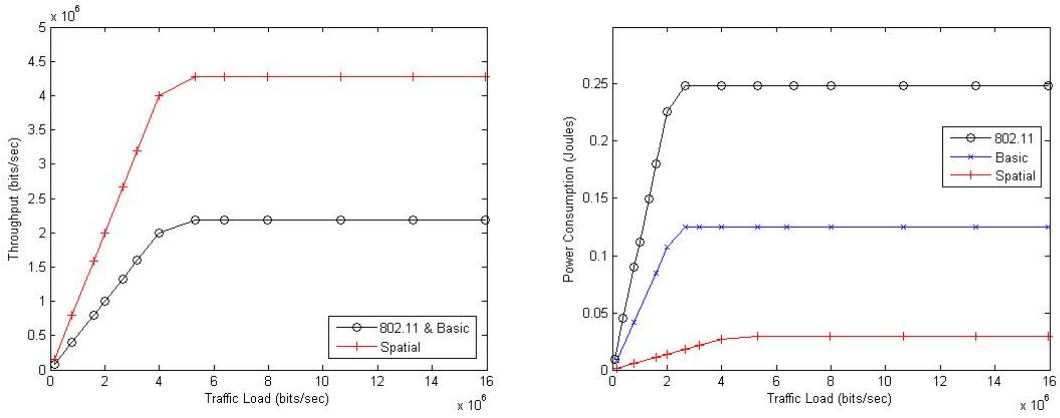



**Figure 40: Throughput (left) and power consumption (right) for linear topology**

## B. Grid topology

We now study the performance under more realistic network topologies. First, we consider a grid topology as shown in Figure 41, where 16 nodes are placed within a spare area of length 375 m with adjacent columns and rows separated by 125m. Each node selects a random adjacent node as its destination to highlight the benefit of spatial reuse.

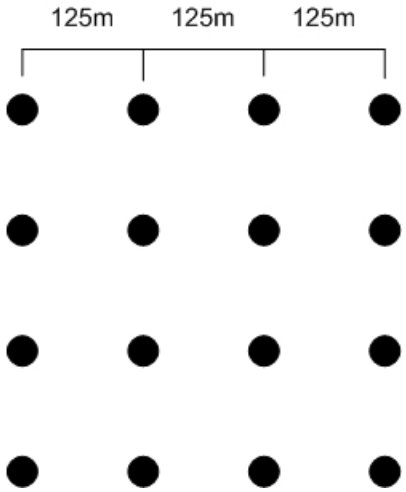



**Figure 41: Grid topology**

The simulation results are demonstrated in Figure 42. We see both throughput and power consumption of spatial TPC dramatically outperform the one of 802.11 and basic TPC. The throughput of spatial TPC is almost tripled compared to 802.11 and basic TPC. This performance enhancement is due to more transmissions achieved by transmission power control.

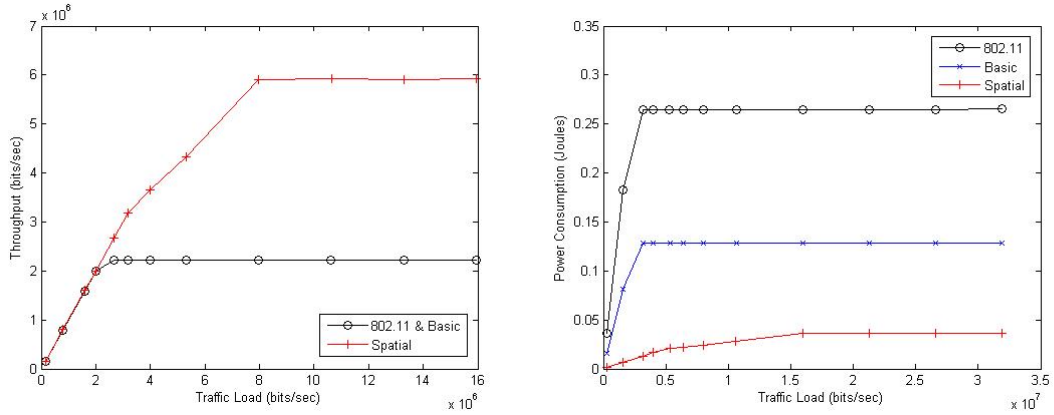



**Figure 42: Throughput (left) and delay (right) for grid topology**

### C. Random topology

Now we study the performance under a random topology as shown in Figure 43, where 16 nodes are randomly placed within a spare area of length 375 m. Each node selects a random adjacent node as its destination to highlight the benefit of spatial reuse.

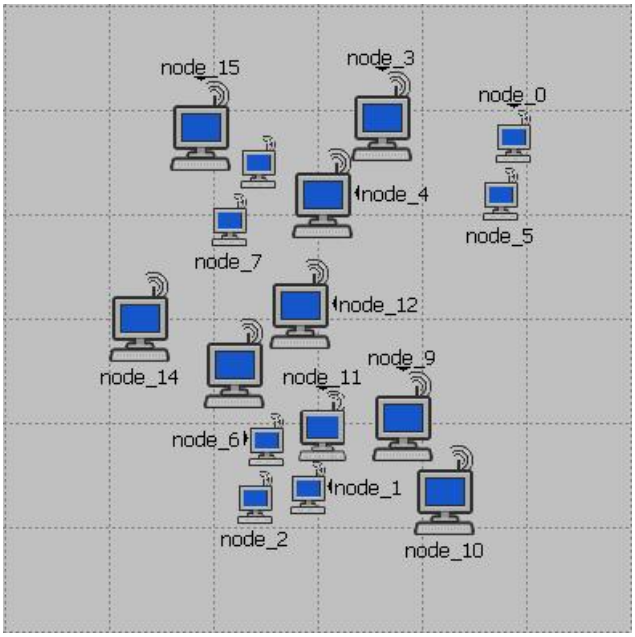



**Figure 43: Random topology**

The simulation results are demonstrated in Figure 44. Again, we see both throughput and power consumption of spatial TPC dramatically outperforms the one of 802.11 and basic TPC. Note that different from linear topology, there are new hidden problems caused by variable transmission ranges in this topology. However, the benefit of spatial reuse outweighs the hidden problem.

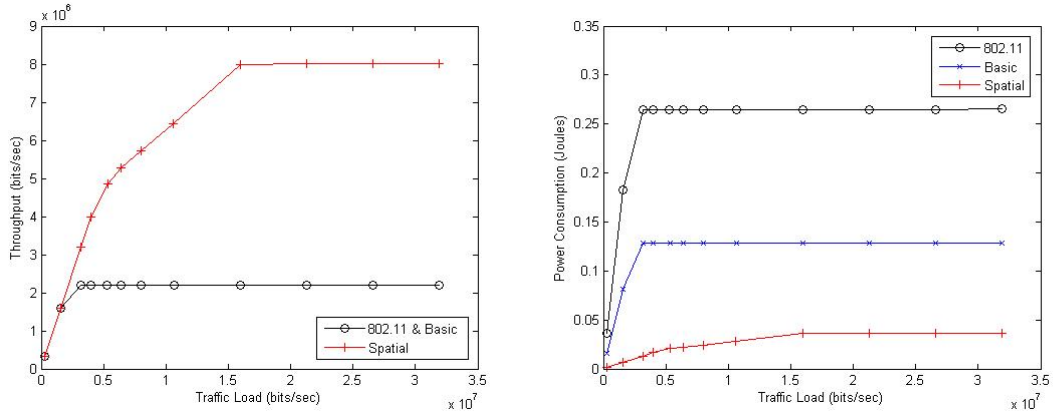



**Figure 44: Throughput (left) and delay (right) for random topology**

### 5.3 Related work

Narayanaswamy et al. proposes a common power (COMPOW) protocol [42] and implements it on a real wireless test-bed. In COMPOW, all the nodes in the network use a common power that can achieve the same level of network connectivity as the highest power level. Jung et al. proposes a power control MAC (PCM) [43] to solve the hidden problem in the basic power control MAC. In PCM, a node also uses maximum transmit power for *RTS/CTS* packet, however, it periodically increases the transmit power during Data transmissions to inform the on-going transmission to all adjacent nodes.

Monks et al. proposes the power controlled multiple access (PCMA) protocol [44] based on busy tone. In PCMA, a node measures the noise and interference level during its data packet reception and calculates the additional interference it can tolerate. If a node wants to transmit a data packet, it collects the busy tone sent by all nearby receivers and calculates the maximum power it is allowed to transmit according to the maximum tolerate interference power and busy tone information. With the aid of busy tone, PCMA solves hidden problem caused by variable transmission range and fully exploits the benefit of spatial reuse.

Muqattash et al. proposes a power control MAC (POWMAC) for single-channel single-transceiver ad hoc networks [45]. POWMAC uses *CTS* packet and a newly defined *decide-to-send (DTS)* packet to adjust power level according to the surrounding noise level. In POWMAC, an *access window (AW)* is also used to allow for a number of *RTS/CTS* exchanges to happen before several concurrent data transmission, so that multiple transmissions can happen simultaneously after a series of *RTS/CTS* exchanges.

Different power control schemes for directional antennas are also studied. Nasipuri et al. [46] studies the power consumption and throughput enhancement for directional MAC protocols. Arora et al. [47] proposes a directional MAC protocol with power control (DMAP) based on separated channel for control packets and data packets. The transmission power for data packet is calculated based on the adjacent interferences. Huang et al. [48] proposes a power control MAC protocol using directional antennas for both transmission and reception. Nodes make rendezvous with destination node for both direction and power level information in the reservation period and simultaneous transmissions happen after the reservation period. [40, 49]

### 5.4 Chapter Summary

In this chapter, we propose the *spatial TPC* based on the *basic TPC* with the intention of fully exploiting the benefits of spatial reuse. We identify the new hidden problem for spatial TPC due to variable transmission ranges. Through simulation comparisons, we find spatial reuse achieves highest throughput and lowest power consumption compared to 802.11 and basic TPC.

# Chapter 6: Directional MAC

In this chapter, we firstly discuss how the utilisation of directional antennas affects the MAC layer decision. Then we present spatial directional MAC based on the basic directional MAC, followed by simulation results and performance comparison under various network topologies.

## 6.1 Impact of directional antennas

When implementing directional antennas, the first issue is in which scenarios we want to implement directional antennas. Basically there are three scenarios that a node uses its antennas namely: *transmission*, *reception* and listening to the channel when it is *idle*. When we implement directional antennas at the receiver, one fatal problem for reception is called *deafness problem* which was firstly proposed by Choudhury [50]. The receiver is only aware of the transmission in the direction where it points to and transmissions from all other directions cannot be heard by the receiver. Due to deafness problem, directional reception becomes unsuitable for random access protocols, since a node will never know in which direction that a random transmission will happen.

The solution of Choudhury's work is to only use directional antennas for reception and once the reception is finished, the receiver changes the antenna pattern back to omni-directional antennas immediately to monitor the channel changes from all directions. Compared to only using directional antennas at transmitters, this solution achieves a longer transmission range due to the directional reception. However, from the spatial reuse aspect, directional transmission has no difference from directional transmission and reception. So in this thesis, we focus on studying how to implement directional

antennas at a transmitter. Next we introduce both advantages and challenges of implementing directional antennas at transmitters.

**A. Advantages**

As mentioned in Section 2.1, directional antennas can focus most of its energy in an intended direction and eliminate interferences with transmissions in other directions. This feature gives us opportunities to increase the network performance by exploiting the spatial reuse of directional antennas. As shown in Figure 45, four nodes are within the transmission range of each other. When using omni-directional antennas, only one transmission can happen at any given time. However, if directional antennas are used, two transmissions can happen simultaneously, since the two transmissions do not interfere with each other. As a result, directional antennas can achieve a higher network throughput compared to omni-directional antennas.

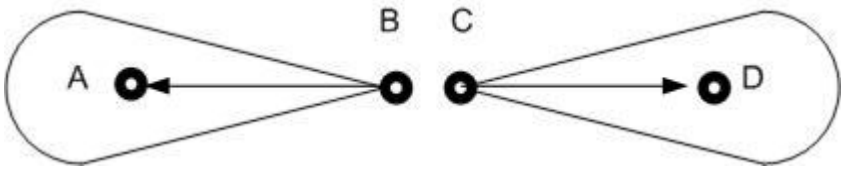



**Figure 45: Spatial Reuse**

Another advantage of using directional antennas is longer transmission ranges and low power consumptions. By focusing the power in one direction, the antenna can achieve a high quality signal with a longer transmission range. This means it is possible to deliver a packet in fewer hops than omni-directional antennas. As shown in Figure 46, node A can reach node E in two hops rather than 4 hops if directional antennas are used. As a result, the higher network connectivity will dramatically improve routing performances.

Meanwhile, directional antennas consume less power than omni-directional antennas to reach a node with same distance. The bigger the directional antenna's gain is, the more power it is saved by the directional antenna.

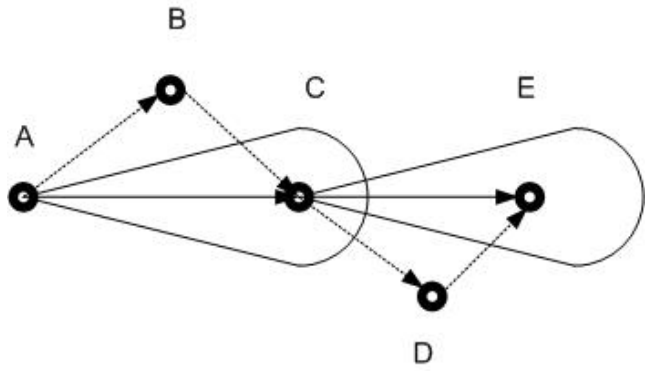



**Figure 46: Longer Transmission Range**

**B. Challenges**

One challenge for directional transmission is broadcasting, since directional antennas change the broadcasting nature of omni-directional antennas. When a direction antenna has a packet to broadcast, it has to switch its antenna consistently 360 degree named sweeping or scanning. Clearly, the shortcomings for sweeping are large overheads and longer delays.

Another challenge for directional antennas is called the new hidden problem. It is due to the unawareness of transmissions using directional antennas. As shown in Figure 47, node S is communicating with node D using directional antennas. If node B does not broadcast any packet such as *RTS* or *CTS*, then node C would never know the transmission at node B. So both physical and virtual CS at node C indicates that the channel is free and if node C initiates a transmission to node B, collisions will happen.

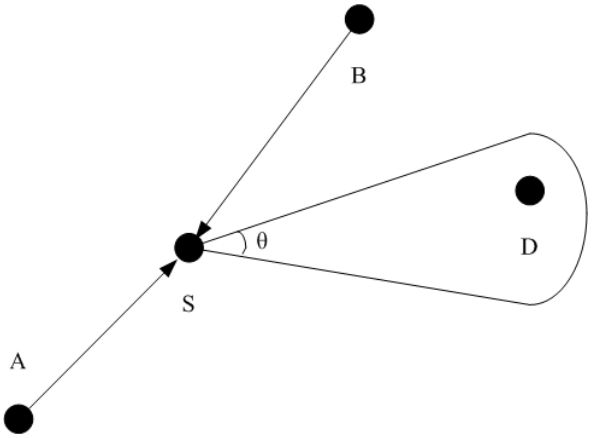



**Figure 47: New hidden node problem**

### 6.2 Directional MAC

We now introduce directional NAV (DNAV) and basic directional MAC (DMAC) that are well studied by previous work [51-56].

**A. DNAV** [57]

Before introducing the basic directional MAC protocol, we firstly present the modification of NAV which will be used in basic directional MAC protocols. As mentioned before, each node in the network maintains a NAV. If a node overhears a packet with NAV larger than its own NAV, it updates its NAV duration and does not access the channel until the NAV duration expires. However, for directional antennas, when a node overhears a transmission from one direction, it can still initiate transmission in other directions to fully exploit the spatial reuse achieved by directional antennas.

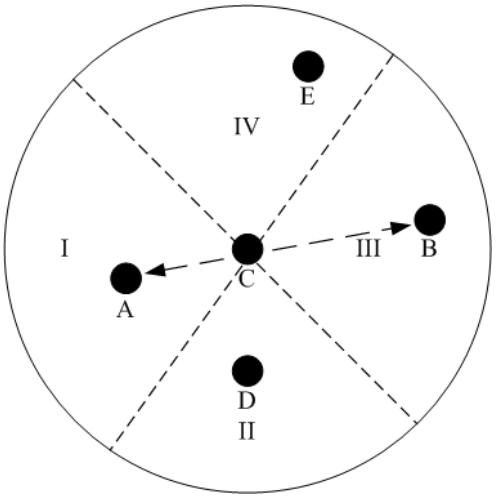



**Figure 48: DNAV**

For DNAV, a node's NAV is divided into several sub-NAV according to the beamwidth of directional antennas. As shown in Figure 48, the beamwidth of directional antennas at node C is $\frac{\pi}{2}$, so the NAV of node C is divided into 4 sub-NAVs. Assuming node A and B is communicating with each other, upon receipt of packets from node A and B, node C updates its NAV durations in directions of I and III. However its NAV durations in directions II and IV are kept unchanged, so that transmissions with node D or E will not be blocked at node C.

**B. DMAC** [50]

The DMAC can be described as follows. When a node receives a packet from its upper layer, it firstly reads the MAC address that this packet is destined to and looks up direction information of the destination node in its neighboring table to decide which direction it should beamform its antenna to. The node physically senses the channel and checks the NAV duration in its DNAV. If both PCS and VCS indicate that the channel is free for longer than SIFS time, it transmits a *RTS* using the selected beam to its destination node and waits for the *CTS* from detonation node. Otherwise if the channel

is busy, it waits until the channel is free and then enters into backoff stage. When backoff is finished, it transmits a *RTS* using the selected beam to its destination node and waits for the *CTS* from detonation node. Upon receiving *RTS* from the source node, the destination node beamforms its antenna to the source node, waits for a SIFS time and then transmits *CTS* back to the source node. When the source node receives the *CTS* from the destination node, it initiates the *Data* and *Ack* handshake with its destination node. If any of responding packet such as *CTS* and *Ack* is missed, the source node doubles its contention window and initiates the transmission again.

### 6.3 Four schemes and Simulation results

Now we introduce four schemes with the intention of overcoming the new hidden problems in DMAC discussed in Section 5.2.1. For all the four schemes, a node receives packets omni-directionally and listens to the channel omni-directionally when it is idle. However, it transmits packets using directional antennas. To fully explore the spatial reuse of directional antennas, we transmit *Data* and *Ack* packets using directional antennas. However, we would like one of *RTS* and *CTS* to be transmitted omni-directionally to inform transmission nodes' neighbor the on-going transmission.

These four schemes are directional *RTS* / *CTS*, directional *RTS* / omni-directional *CTS*, omni-directional *RTS* / directional *CTS* and omni-directional *RTS*/*CTS* respectively[8]. For DD and DO schemes, we assume the source node knows about the location of destination nodes by using GPS or the knowledge obtained during neighbor discovery period, so that source node can beamform its antenna to the right direction for *RTS* transmission.

[8] We call the four schemes DD, DO, OD and OO respectively and DD scheme is same as DMAC mentioned above.

To study the performance of four schemes, we have eight scenarios with different topologies and traffic patterns as shown in Figure 49. For all the scenarios, two adjacent rows or columns are separated by 125 meters. The channel data rate is 11 Mbps and packet size is 512 bytes[9]. The beamwidth and gain for the directional antenna are $\frac{\pi}{4}$ and 10dB respectively. Transmission power for omni-directional antennas and directional antennas are 0.1 mW and 0.01 mW respectively, and reception power threshold is 3.16E – 13 W, so the transmission range is about 175 metres.

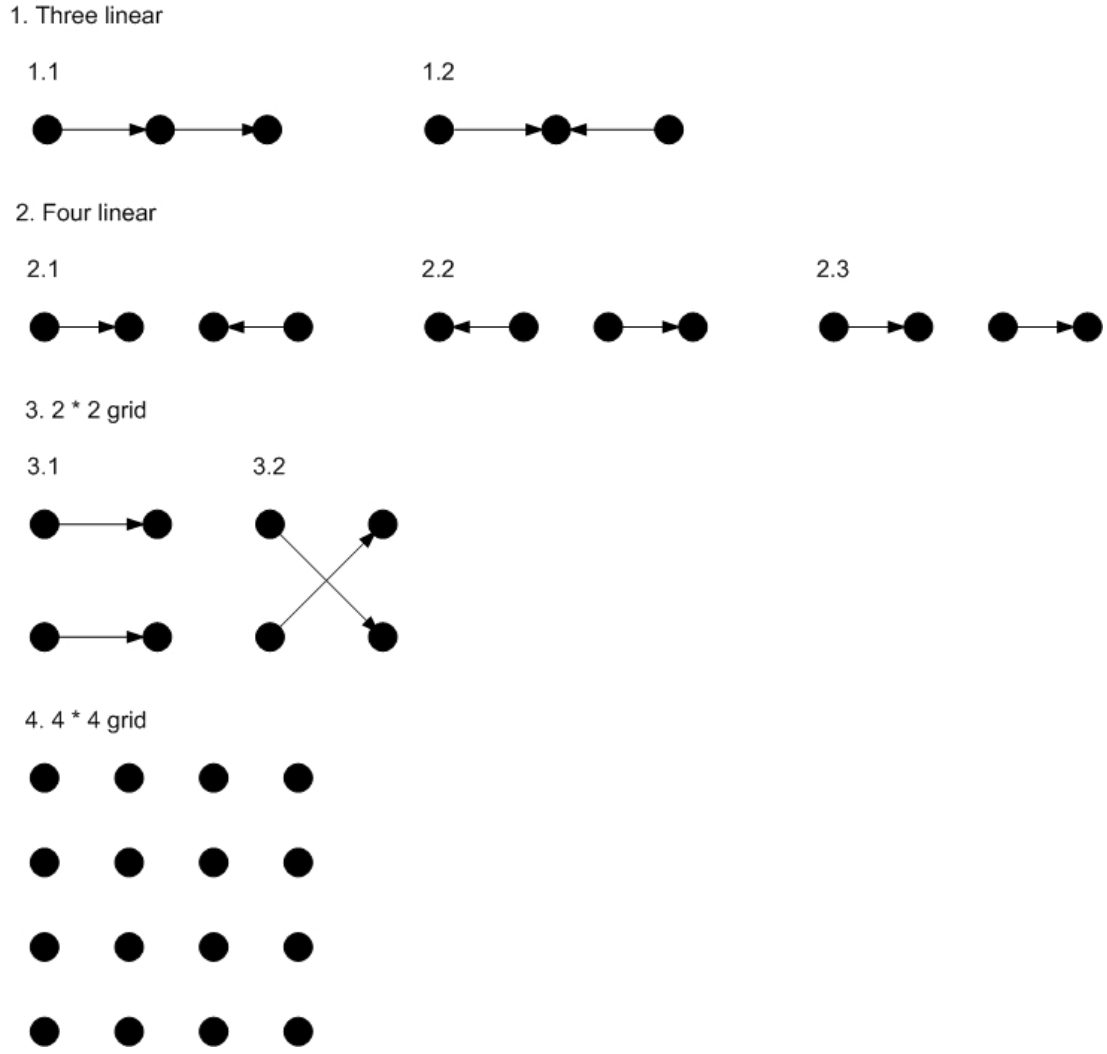



**Figure 49: Simulation topologies**

[9] Fragmentation is not considered.

**1. Three linear**

For the two three-linear topologies, since two traffics intersect at node 1 and according to SISO rules in Section 2.1, node 1 cannot take part in two communications at the same time. As a result, we cannot take advantage of the spatial reuse for directional antennas. Instead, the performance of directional antennas decreases due to the new hidden problem.

- **Topology 1.1**

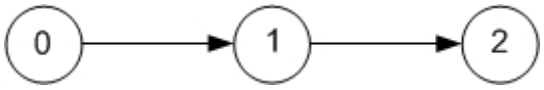



**Figure 50: Topology 1.1**

For topology 1.1 shown in Figure 50, node 1 acts as both a transmitter and a receiver. If node 0 first initiates a transmission with node 1, then node 1 will not try to start a transmission with node 2 and there is no risk of confliction 2 or 3. However, if node 1 first initiates a transmission with node 2, due to the new hidden problem, node 0 may not know whether there is a transmission between node 1 and node 2. So in topology 1.1, the omni-directional *RTS* is recommended, so that node 1 can inform the transmission to node 0 to avoid conflictions.

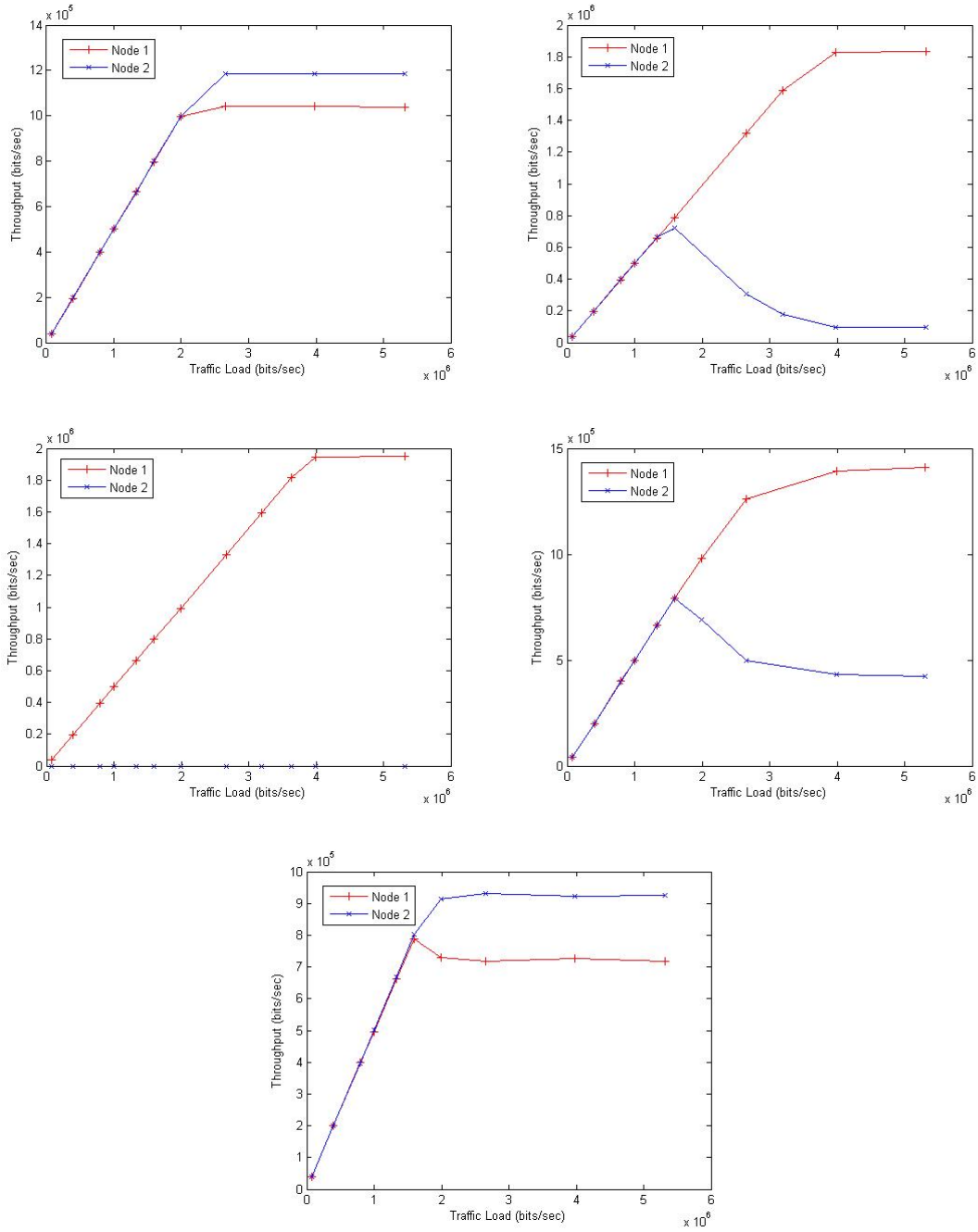



**Figure 51: Throughput and Fairness**

**(802.11 top left, DD top right, DO middle left, OD middle right and OO bottom)**

The throughput for node 1 and 2 under different schemes are shown in Figure 51. We can see that since the original 802.11 transmits all its packets omni-directionally, it achieves the highest throughput compared to the four schemes. Meanwhile, network throughput is evenly distributed to node 1 and 2 in original 802.11. However, for DD and DO modes, since node 1 cannot inform its transmission to node 0, the transmission

between node 1 and 2 is frequently interrupted by node 0, hence, most of the network throughput is distributed to node 1.

- **Topology 1.2**

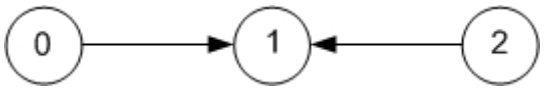



**Figure 52: Topology 1.2**

This topology and traffic pattern is known as the hidden problem. Node 1 can only successfully receive one packet either from node 0 or node 2 at a time. If node 0 and node 2 transmit packet simultaneously, both of packets collide at node 1 and none of them is successfully received. In this topology omni-directional *CTS* at node 1 is recommended to inform and block one node's transmission if it is already in transmission with the other node. As shown in Figure 53, throughput of DD and OD modes that use directional *CTS* is smaller than those which use omni-directional *CTS* due to collisions at node 1.

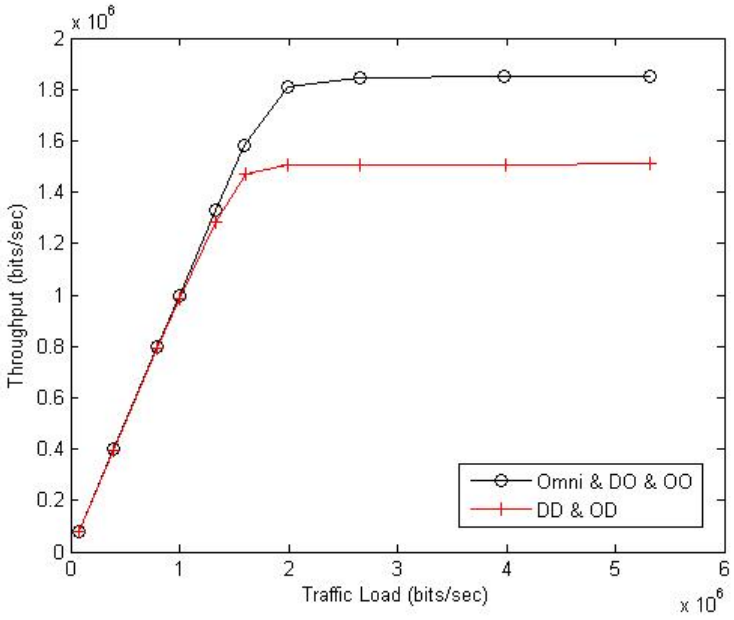



**Figure 53: Throughput under topology 1.2**

**2. Four linear**

For all the three four- linear topologies, there are two transmission pairs which are node 0, 1 and node 2, 3 respectively. Ideally, we expect that the network throughput will be doubled compared to topology 1 due to the benefit of spatial reuse. In the four-linear topology, since node 1 and 2 are within the transmission range of each other, node 1 and 2 are prone to interfere with each other. Whether the two transmissions could happen simultaneously depends on the traffic pattern and MAC schemes.

- **Traffic 2.1**

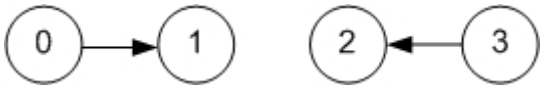



**Figure 54: Topology 2.1**

For traffic pattern 2.1, if omni-directional transmissions of *CTS* are used, one pair transmission could be interfered by the other pair's *CTS* packet. For example, when node 2 is transmitting *CTS* back to node 3, node 1 is receiving *RTS* or *Data* packets from node 0. Then packets from node 0 and node 2 collide at node 1, as a result, node 1 cannot successfully receive packet from node 0 and node 0 has to restart its transmission. So directional *CTS* is suggested for this topology.

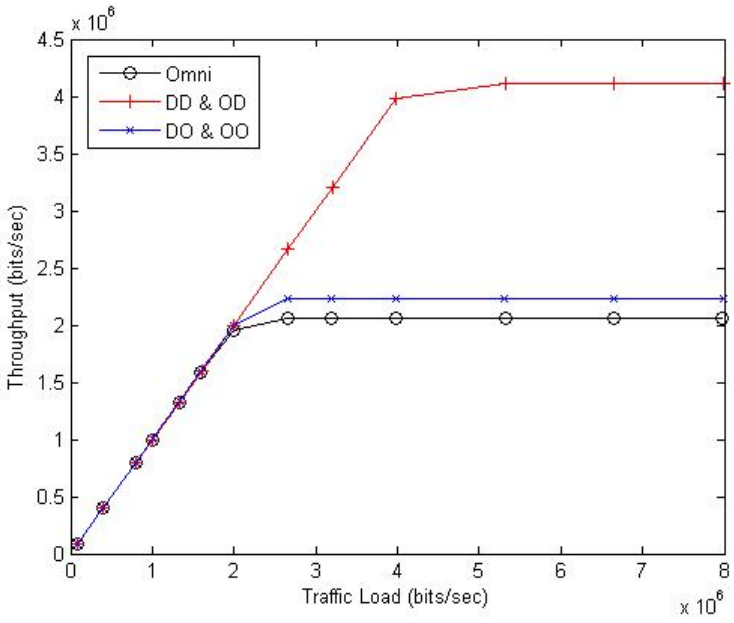



**Figure 55: Throughput under topology 2.1**

The simulation result for different antenna modes under different traffic loads is shown in Figure 55. We can see that DD and OD modes nearly double the number of throughput compared to other modes. This high performance is due to the benefit of spatial reuse achieved by directional *CTS*. But the throughput of OD and OO modes only slightly outperforms original 802.11 due to the interferences of omni-directional packets between node 1 and 2.

- **Topology 2.2**

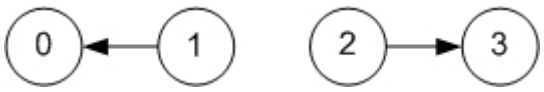



**Figure 56: Topology 2.2**

For traffic pattern 2.2, both node 1 and 2 are transmission initiators. Directional *RTS* is suggested, since omni-directional *RTS* generates interference between node 1 and node 2. The simulation results for different modes under different traffic load are shown in

Figure 57. We can see that DD and DO modes nearly double the number of throughput compared to other modes.

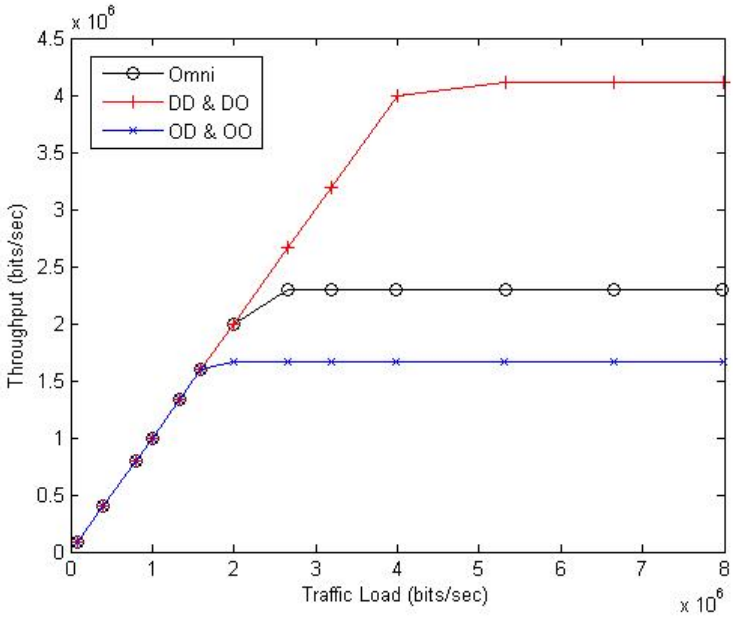



**Figure 57: Throughput under topology 2.2**

- **Topology 2.3**

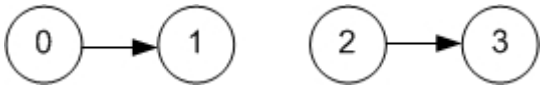



**Figure 58: Topology 2.3**

For traffic pattern 2.3, node 1 is a receiver and node 2 is a transmission initiator. If *RTS* is transmitted omni-directionally, then node 1 will be interfered with the *RTS* from node 2. And if *CTS* is transmitted omni-directionally, then node 2 is interfered with the *CTS* from node 1. However, if we transmit all packets using directional antennas, the two transmission pair does not interfere with each other at all. The simulation results in Figure 59 show that the DD mode outperforms all other three modes as well as 802.11.

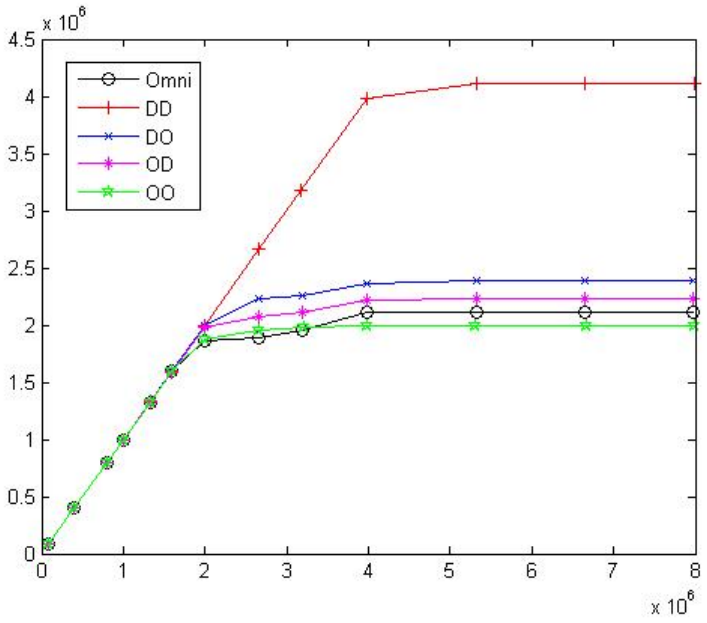



**Figure 59: Throughput under topology 2.3**

### 3. 2*2 grid

Different from four-linear topology, all four nodes in 2*2 grid topology are within transmission range of each other. As a result, packet collision is more serious than four-linear topology.

- **Topology 3.1**

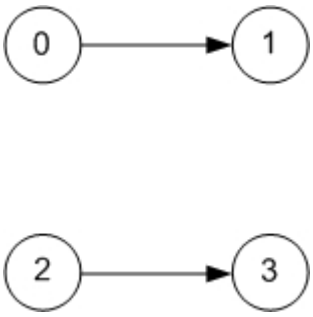



**Figure 60: Topology 3.1**

When using DD, since the two transmission pair 0-1 and 2-3 do not capture each other at all, the two transmissions can happen simultaneously. Hence, the throughput of DD is

nearly doubled compared to omni-directional antennas as shown in Figure 61. However, the throughput for DO, OD and OO are even slightly smaller that omni-directional antennas. This throughput decrease is due to collisions between one transmission of a pair of nodes and the omni-directional control packets *RTS* or *CTS* from the other pair of nodes. For example node 2 has received the omni-directional *RTS* from node 0, it does not need to defer its transmission to node 3, since node 3 lies in a different direction with node 0. However, if node 2 is transmitting a *Data* packet to node 3 while node 1 is transmitting an omni-directional *CTS* to node 0, then the *CTS* from node 1 will collide with the *Data* packet at node 3. So, for DO, OD and OO modes, we say the omni-directional control message is not useful but harmful in this topology.

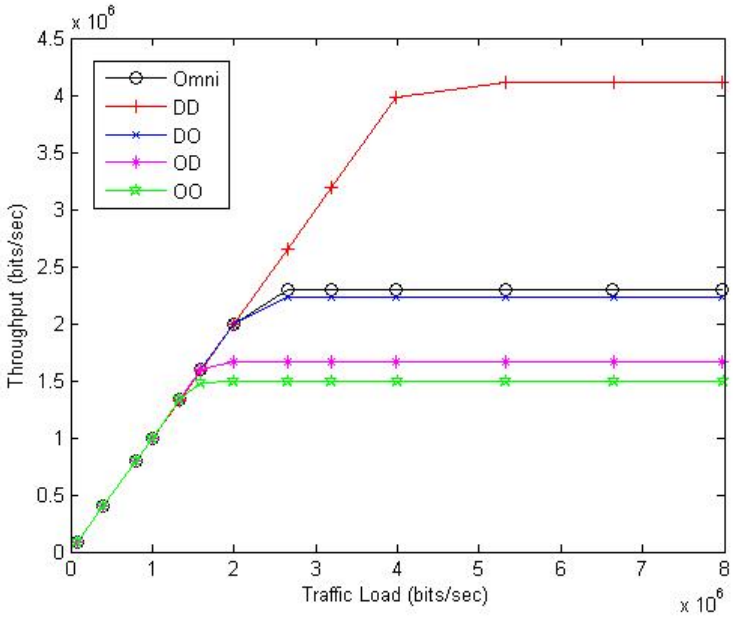



**Figure 61: Throughput under topology 3.1**

- **Topology 3.2**

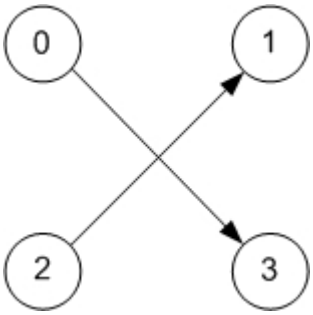



**Figure 62: Topology 3.2**

This topology is similar to topology 3.1, DD performs nearly twice better than omni-directional transmissions. And omni-directional control messages decrease network throughput by colliding with data packets from the other pair of nodes. Meanwhile, the throughput is slightly smaller than topology 3.1 due to higher interferences between two transmission pairs.

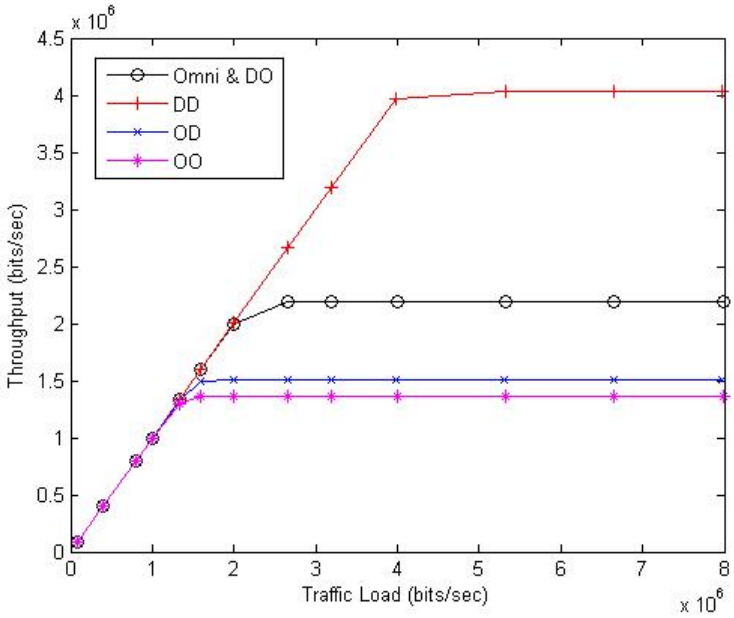



**Figure 63: Throughput under topology 3.2**

**4. 4*4 grid**

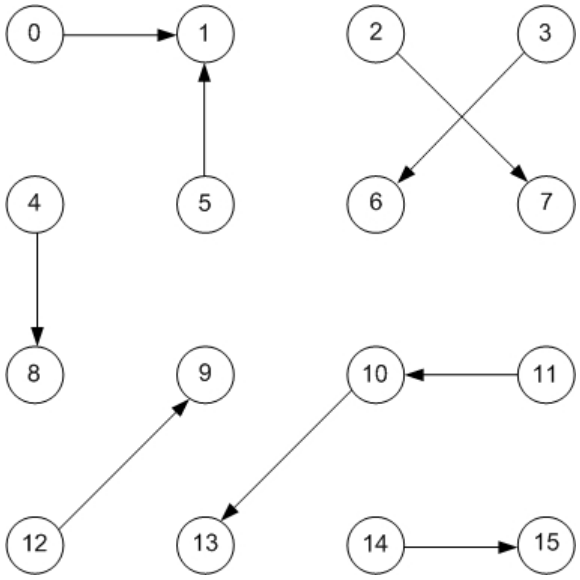



**Figure 64: Topology 4**

We now test the performance of four schemes in a 4*4 grid topology as shown in Figure 64 containing different traffic patterns. From simulation results shown in Figure 65, we can see that the throughput of DD mode dramatically surpasses other schemes and 802.11. This performance is attributed to less capture and collisions caused by the omni-directional packets.

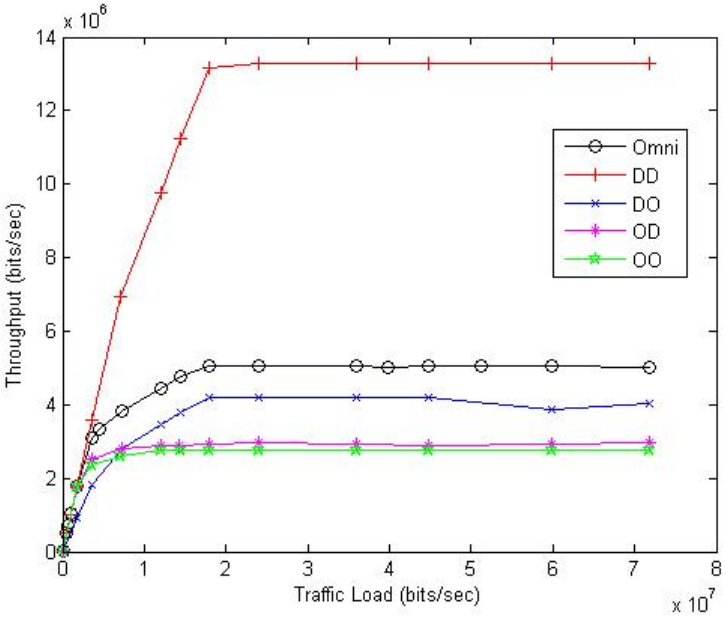



**Figure 65: Throughput under topology 4**

To sum up, for topology 1, since two transmissions intersect with each other, throughput cannot be enhanced by directional spatial reuse. Instead, throughput decreases due to the unawareness of transmission caused by directional transmission of control packets. However, for topology 2 and 3, directional antennas can dramatically increase the network throughput by the benefit of spatial reuse. Meanwhile, the network throughput for different schemes is more sensitive to the network topology and traffic pattern.

### 6.4 Related work

Nasipuri et al. [51] assumes nodes in the network have no knowledge of neighbor nodes' location information and each node is equipped with M directional antennas with an angle of $\frac{2\pi}{M}$. The directional MAC works in the following ways. Idle node listens to channel omni-directionally. The source node and destination node firstly build up *RTS/CTS* handshake using omni-directional transmissions. During the *RTS/CTS*

handshake, two nodes get the location information of each other. Then they use directional antennas for *Data-Ack* exchange. Though this protocol is not dependent on the assumption that all the nodes have location information for neighboring nodes, the omni-directional RTS and CTS reduce the benefit of spatial reuse.

Korakis et al. [52] proposes a Circular-DMAC protocol, which only uses directional transmissions to exploit the benefit of directional antennas. In this paper, if there is no location information for the destination node in the cache table, a node scans 360 degrees using a consistent directional transmission to get the location information of neighbors as well as the destination node. As a result the protocol can exploit the benefits of directional antennas when assure the accurate location information for neighborhood. However, this protocol incurs high control overhead and delay due to the circular scanning.

Y. B. Yo et al. [53] uses omni-directional CTS to address the new hidden problem. Two schemes are proposed which are using only directional *RTS* and using both directional *RTS* and omni-directional *RTS*. The two schemes are also under the assumption that the location information for neighboring nodes can be obtained by GPS. And simulation studies under different topologies show that the two schemes outperform 802.11 with omni-directional antennas by exploiting the benefit of spatial reuse.

Dual Busy Tone Multiple Access with Directional Antennas (DBTMA/DA) is a variation of DBTMA by using directional antennas proposed by Haas et al. Two busy tones are used besides the directional RTS/CTS: a transmission busy tone and a reception busy tone. Each tone is assigned a unique frequency in the control channel.

Any node which hears a transmission or reception tone defers its receiving or transmitting. As a result, both hidden and exposed terminal problems are solved.

Choudhury et al. proposes a variation to basic DMAC named multihop RTS MAC (MMAC) which fully exploits the longer transmission range of directional antennas. MMAC uses multihop RTS to establish one hop connections for data packet between source and destination nodes.

Choudhury et al. also proposes Tone-based directional MAC (ToneDMAC) to address the deafness problem. ToneDMAC performs *DRTS/DCTS* directly without trying to inform its neighbors. It transmits a tone after its data communication to inform its neighbors it has been engaged in the recent past. There are two sub channels for ToneDMAC: a data channel for *RTS*, *CTS*, *Data* and *Ack* messages and a narrow control channel for tones. Each tone is unique and can be distinguished by a tone frequency and a duration that are hash functions of the node's identifier.

Bao et al. [54] proposes distributed receiver-oriented multiple access (ROMA) using multi-beam adaptive array (MBAA), which can form multiple beams for transmission or reception simultaneously. All nodes are synchronised and divided into transmitters and receivers. ROMA achieves collision free using two-hop topology information.

The directional transmission and reception algorithm (DTRA) [55] is proposed to use pure directional transmission and reception using directional antennas. Similar to ROMA, all nodes are synchronised, and time in DTRA is divided into 3 sub-frames,

which are used for neighbor discovery, reservation and data transmission respectively. Power control is also achieved in DTRA [56].

### 6.5 Chapter Summary

In this chapter, we firstly present the impact of implementing directional antennas and existing work of directional MAC protocols. We then propose four schemes that adopt different transmission strategies of *RTS/CTS* based on basic DMAC with the intention of overcoming new hidden node problem of directional antennas. By extensive simulation under different topologies and traffic patterns, we find that even facing new hidden node problem, DD scheme still outperforms other schemes due to fully exploitation of spatial reuse with directional antennas.

# Chapter 7: Conclusions

## 7.1 Conclusions

The research activities have been documented in several chapters and they are summarised and concluded as follows.

- In Chapter 2, we firstly introduced physical issues of wireless medium that impact on the decisions on MAC layer such as directional antennas and propagation model. Then we presented background knowledge of neighbor discovery, MAC protocols as well as broadcasting algorithms in wireless ad hoc networks.

- In Chapter 3, we studied the functionality details of 802.11 DCF including DCF timing, backoff, carrier sensing as well as *RTS/CTS* handshake. We also argued the effectiveness of *RTS/CTS* handshake under different topologies. Both numerical and simulation results indicated that *RTS/CTS* should be turned off when the data packet size is small.

- In Chapter 4, we introduced basic knowledge for OPNET simulation including modeling mechanisms, wireless pipelines as well as details of 802.11 model in OPNET, which lay a fundamental work for the following simulations in Chapter 5 and 6.

- In Chapter 5, we exploited the benefit of spatial reuse in wireless ad hoc networks which is achieved by *TPC*. We identified the new hidden problem caused by variable transmission ranges and showed the performance enhancement of spatial TPC by extensive simulations.

- In Chapter 6, we identified the new hidden problem caused by deafness of *RTS/CTS* messages, and tested the performance of four different schemes under different network topologies and traffic patterns. Simulation results showed that, among the four schemes, DD mode outperformed other three schemes by fully exploiting the benefit of spatial reuse.

In conclusion, by extensive simulation studies under different topologies and traffic patterns, we found that, even facing new hidden problems caused either by variable transmission ranges in TPC or deafness of *RTS/CTS* messages in directional MAC protocols, performances of spatial TPC and directional MAC still dramatically surpassed the original 802.11 by fully exploiting the benefit of spatial reuse in wireless ad hoc networks.

### 7.2 Future Work

There are several directions in which our results can be extended and they are listed as follows:

- In Chapter 5, we proposed spatial TPC to enhance the network capacity. However, the spatial TPC faces new hidden node problems due to variable transmission ranges. We plan to overcome the new hidden node problem by modifying the spatial TPC. In the new spatial TPC, a node transmits *RTS* or *CTS* at maximum power level. In addition to the NAV duration, we also encapsulate the power level of succeeding *Data* packet in the *RTS* or *CTS* packet. When a neighbor node receives the *RTS* or *CTS* from the source node, it defers its transmission only if its intended transmission

interferes with the source node. In this way, the new spatial TPC can overcome the new hidden node problem without losing the benefits of spatial reuse.

- In this thesis, we attempted to increase the network capacity by exploiting the spatial reuse achieved by directional antennas. Another advantage of directional antennas is the longer transmission range compared to omni-directional antennas. Based on the developed DMAC model, we plan to implement directional antennas in DSR routing protocol for a crossed-layer design. The longer transmission range of directional antennas can increase the network connectivity so that a node can reach a destination node in fewer hops. As a result, packet delivery ratio and delay in DSR with directional antennas are expected to outperform the one with omni-directional antennas.

- As mentioned in Chapter 2, broadcasting is a fundamental operation in wireless ad hoc networks. However, it suffers serious broadcasting redundancies that degrade the network performance. We plan to design a broadcasting algorithm for both omni-directional and directional antennas. In the new designed algorithm, each broadcasting node maintains a table of two hop neighbor information and it selects a minimum subset of its one hop neighbors that can cover all the two hop neighbors to relay the broadcast message. In this way, it reduces broadcast redundancies while does not lose the delivery reliability.

## References


1. *Information technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* ANSI/IEEE Std 802.11, 1999 Edition (R2003), 2003: p. i-513.

2. Gupta, P. and P.R. Kumar, *The capacity of wireless networks.* Information Theory, IEEE Transactions on, 2000. **46**(2): p. 388-404.

3. Yi, S., Y. Pei, and S. Kalyanaraman, *On the capacity improvement of ad hoc wireless networks using directional antennas.* Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, 2003: p. 108-116.

4. Li, J., et al., *Capacity of Ad Hoc wireless networks.* Proceedings of the 7th annual international conference on Mobile computing and networking, 2001: p. 61-69.

5. Nelson, R. and L. Kleinrock, *Spatial TDMA: A Collision-Free Multihop Channel Access Protocol.* Communications, IEEE Transactions on [legacy, pre-1988], 1985. **33**(9): p. 934-944.

6. Kleinrock, L. and J. Silvester, *Spatial reuse in multihop packet radio networks.* Proceedings of the IEEE, 1987. **75**(1): p. 156-167.

7. Ye, F., S. Yi, and B. Sikdar, *Improving Spatial Reuse of IEEE 802.11 Based Ad Hoc Networks.* Proceedings of IEEE GLOBECOM, 2003.

8. Xu, K., M. Gerla, and S. Bae, *Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks.* Ad Hoc Networks, 2003. **1**(1): p. 107-123.

9. Ma, H.A., E.; Roy, S.,. *Analysis and Simulation Model of Physical Carrier Sensing in IEEE 802.11 Mesh Networks,*. in *OPNETWORK Conference 2006*. Aug. 2006. Washington DC.

10. Basagni, S., et al. *A distance routing effect algorithm for mobility (DREAM)*. in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*. 1998.

11. Ko, Y.B. and N.H. Vaidya, *Location-Aided Routing (LAR) in mobile ad hoc networks.* Wireless Networks, 2000. **6**(4): p. 307-321.

12. Jiang, M., J. Li, and Y.C. Tay, *Cluster Based Routing Protocol (CBRP).* draft-ietf-manet-cbrp-spec-01. txt, Internet Draft, IETF, Aug, 1999.

13. Xiaoyan, H., X. Kaixin, and M. Gerla, *Scalable routing protocols for mobile ad hoc networks.* Network, IEEE, 2002. **16**(4): p. 11-21.

14. Royer, E.M. and T. Chai-Keong, *A review of current routing protocols for ad hoc mobile wireless networks.* Personal Communications, IEEE [see also IEEE Wireless Communications], 1999. **6**(2): p. 46-55.

15. Hu, L., *Topology control for multihop packet radio networks.* Communications, IEEE Transactions on, 1993. **41**(10): p. 1474-1481.

16. Zhuochuan, H., et al. *Topology control for ad hoc networks with directional antennas*. in *Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on*. 2002.

17. Capkun, S., M. Hamdi, and J.P. Hubaux. *GPS-free Positioning in Mobile Ad Hoc Networks*. in *Cluster Computing*. 2002.

18. Niculescu, D. and N. Badri. *Ad hoc positioning system (APS) using AOA*. in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*. 2003.

19. Vasudevan, S., J. Kurose, and D. Towsley, *On neighbor discovery in wireless networks with directional antennas.* Proceedings of INFOCOM, 2005.

20. McGlynn, M.J. and S.A. Borbash, *Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks.* Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, 2001: p. 137-145.

21. Tobagi, F. and L. Kleinrock, *Packet Switching in Radio Channels: Part III--Polling and (Dynamic) Split-Channel Reservation Multiple Access.* Communications, IEEE Transactions on [legacy, pre - 1988], 1976. **24**(8): p. 832-845.

22. Tobagi, F. and L. Kleinrock, *Packet Switching in Radio Channels: Part IV--Stability Considerations and Dynamic Control in Carrier Sense Multiple Access.* Communications, IEEE Transactions on [legacy, pre - 1988], 1977. **25**(10): p. 1103-1119.

23. Ramanathan, S. *A unified framework and algorithm for (T/F/C)DMA channel assignment in wireless networks*. in *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*. 1997.

24. Garcia-Luna-Aceves, J.J. and C.L. Fullmer, *Floor acquisition multiple access (FAMA) in single-channel wireless networks.* Mobile Networks and Applications, 1999. **4**(3): p. 157-174.

25. Kleinrock, L. and F. Tobagi, *Packet Switching in Radio Channels: Part I--Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics.* Communications, IEEE Transactions on [legacy, pre - 1988], 1975. **23**(12): p. 1400-1416.

26. Tobagi, F. and L. Kleinrock, *Packet Switching in Radio Channels: Part II--The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution.* Communications, IEEE Transactions on [legacy, pre - 1988], 1975. **23**(12): p. 1417-1433.

27. Haas, Z.J. and J. Deng, *Dual busy tone multiple access (DBTMA)-a multiple access controlscheme for ad hoc networks.* Communications, IEEE Transactions on, 2002. **50**(6): p. 975-985.

28. Karn, P., *MACA-a new channel access method for packet radio.* ARRL/CRRL Amateur Radio 9th Computer Networking Conference, 1990. **140**.

29. Bharghavan, V., et al., *MACAW: a media access protocol for wireless LAN's.* Proceedings of the conference on Communications architectures, protocols and applications, 1994: p. 212-225.

30. Zhu, C. and M.S. Corson, *A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks.* Wireless Networks, 2001. **7**(4): p. 371-384.

31. Garcia-Luna-Aceves, J.J. and A. Tzamaloukas, *Reversing the collision-avoidance handshake in wireless networks.* Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, 1999: p. 120-131.

32. Perkins, C.E. and E.M. Royer, *Ad hoc On-Demand Distance Vector (AODV) Routing.* Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999. **100**.

33. Johnson, D.B., D.A. Maltz, and J. Broch, *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks.* Ad Hoc Networking, 2001. **1**: p. 139-172.

34. Tseng, Y.C., et al., *The Broadcast Storm Problem in a Mobile Ad Hoc Network.* Wireless Networks, 2002. **8**(2): p. 153-167.

35. Tseng, Y.C., S.Y. Ni, and E.Y. Shih, *Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network.* Computers, IEEE Transactions on, 2003. **52**(5): p. 545-557.

36. Haas, Z.J., J.Y. Halpern, and L. Li. *Gossip-based ad hoc routing*. in *Networking, IEEE/ACM Transactions on*. 2006.

37. Li, X.Y., K. Moaveninejad, and O. Frieder, *Regional Gossip Routing for Wireless Ad Hoc Networks.* Mobile Networks and Applications, 2005. **10**(1): p. 61-77.

38. Dai, F. and J. Wu, *Efficient broadcasting in ad hoc wireless networks using directional antennas.* Parallel and Distributed Systems, IEEE Transactions on, 2006. **17**(4): p. 335-347.

39. Chunyu, H., H. Yifei, and J. Hou. *On mitigating the broadcast storm problem with directional antennas*. in *Communications, 2003. ICC '03. IEEE International Conference on*. 2003.

40. Hou, T.C. and V. Li, *Transmission Range Control in Multihop Packet Radio Networks.* Communications, IEEE Transactions on [legacy, pre-1988], 1986. **34**(1): p. 38-44.

41. *Opnet documentation*.

42. Narayanaswamy, S., et al., *Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol.* European Wireless Conference, 2002. **2002**.

43. Jung, E.S. and N.H. Vaidya, *A Power Control MAC Protocol for Ad Hoc Networks.* Wireless Networks, 2005. **11**(1): p. 55-66.

44. Monks, J.P., V. Bharghavan, and W.M.W. Hwu, *A power controlled multiple access protocol for wireless packetnetworks.* INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2001. **1**.

45. Muqattash, A. and M. Krunz, *POWMAC: a single-channel power-control protocol for throughput enhancement in wireless ad hoc networks.* Selected Areas in Communications, IEEE Journal on, 2005. **23**(5): p. 1067-1084.

46. Nasipuri, A., K. Li, and U.R. Sappidi, *Power consumption and throughput in mobile ad hoc networks using directional antennas.* Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on, 2002: p. 620-626.

47. Arora, A., M. Krunz, and A. Muqattash, *Directional medium access protocol (DMAP) with power control for wireless ad hoc networks.* Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE. **5**.

48. Huang, Z., Z. Zhang, and B. Ryu, *Power control for directional antenna-based mobile ad hoc networks.* International Conference On Communications And Mobile Computing, 2006: p. 917-922.

49. Gupta, P. and P.R. Kumar. *Critical power for asymptotic connectivity*. in *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*. 1998.

50. Choudhury, R.R., et al., *On designing MAC protocols for wireless networks using directional antennas.* Mobile Computing, IEEE Transactions on, 2006. **5**(5): p. 477-491.

51. Nasipuri, A., et al., *A MAC protocol for mobile ad hoc networks using directionalantennas.* Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE, 2000. **3**.

52. Korakis, T., G. Jakllari, and L. Tassiulas, *A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks.* Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, 2003: p. 98-107.

53. Young-Bae, K., V. Shankarkumar, and N.H. Vaidya. *Medium access control protocols using directional antennas in ad hoc networks*. in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. 2000.

54. Bao, L. and J.J. Garcia-Luna-Aceves. *Transmission scheduling in ad hoc networks with directional antennas*. in *Proceedings of the 8th annual international conference on Mobile computing and networking*. 2002.

55. Zhang, Z., *Pure directional transmission and reception algorithms in wireless ad hoc networks with directional antennas.* Communications, 2005. ICC 2005. 2005 IEEE International Conference on, 2005. **5**.

56. Dai, H., K.-W. Ng, and M.-Y. Wu. *An Overview of MAC Protocols with Directional Antennas in Wireless ad hoc Networks*. in *Wireless and Mobile Communications, 2006. ICWMC '06. International Conference on*. 2006.

57. Takai, M., R. Bagrodia, and A. Ren, *Directional virtual carrier sensing for directional antennas in mobile ad hoc networks.* Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, 2002: p. 183-193.