60 GHZ MAC AND NETWORK DESIGN

BY ZHUO CHEN

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Roy Yates

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

Jan, 2016

ABSTRACT OF THE DISSERTATION

60 GHz MAC and Network Design

by Zhuo Chen

Dissertation Director: Roy Yates

Recent technology advances are poised to enable low-cost, low-power communications in the 7 GHz of unlicensed spectrum at 60 GHz millimeter wave (mmW) frequencies. However, mmW systems that meet the Gb/s data rate demands of wireless multimedia applications must overcome severe propagation effects, including high path loss and high diffraction loss. Consequently, nodes in the network will have to use directional antennas. The narrow main beam widths of directional antennas introduce design challenges for Medium Access Control (MAC) protocols but, at the same time, provide opportunities for routing protocols to improve the network capacity through better spatial reuse. The small wavelength of a 60 GHz signal can help to achieve high directional antenna gain, but it also precludes diffraction around humans, furniture, and similarly-sized objects. These obstacles penalize a 60 GHz link budget by 20-30dB. Therefore, when people are in motion, 60 GHz network links go on and off frequently due to human body blockage; this introduces new design challenges for both routing and transport protocols.

In this dissertation, we propose solutions at the MAC and network layer to address the above challenges. In particular, we first propose an enhanced directional MAC (EDMAC) to resolve the unfairness and low channel utilization issues of deafness in directional MAC protocols for 60 GHz networks. We then study single path routing and find that shortest path routing often fails to exploit the high spatial reuse properties of directional antennas. We propose two

heuristic routing algorithms, namely HOP-FP and FP-HOP, which combine the fattest-path (FP) and minimum-hop (HOP) metrics, with and without the consideration of interference. We then employ multipath routing for 60 GHz networks to fully utilize the high spatial reuse property of directional antennas. We develop an online node-disjoint path discovery process to find multiple node-disjoint paths between the source and the destination without knowledge of the global topology. In addition, we model the characteristics of link outages that are induced by pedestrian blockage. Based on analytic models and MATLAB simulation results, we show that link blockages can be mitigated by multipath routing schemes with blockage timers for broken paths. We use the ns-2 simulator to validate all proposed protocols in this dissertation.

Acknowledgements

This thesis has only been possible due to the support and encouragement of a many people during the long years of my graduate study. Firstly, it is with the greatest respect and admiration that I acknowledge my advisors Professor Roy Yates and Professor Dipankar Raychaudhuri, without whose guidance and encouragement this thesis would not have been possible. They taught me how to do quality research and were always available when I needed any technical advice. The high standards they set through their professionalism and a rigorous approach towards research have contributed immensely towards my own professional growth. I hope to emulate these standards throughout my career. There is still a lot more I can learn from them, both professionally and personally. For this reason, I hope to continue interacting with them in the future.

Besides my advisor, I would like to thank the rest of my thesis committee: Professor Janne Lindqvist and Dr. Daniel Reininger, for their insightful comments and encouragement, but also for the hard question which incensed me to widen my research from various perspectives. My heartfelt thanks to Professor Yanyong Zhang and Professor Narayan Mandayam for their guidance as a member of my dissertation proposal committee.

I thank my fellow lab members from WINLAB for their helpful comments and suggestions on my research, paper drafts, and practice talks. Their friendly company, and insightful and/or entertaining discussions made for a great work environment.

My progress towards a PhD would not have been as smooth without the professionalism and incredible efficiency of the faculties and staffs at WINLAB. My sincere thanks to all of them.

Last but not the least, I would like to thank my family: my parents and wife for supporting me spiritually throughout writing this thesis and my life in general.

Dedication

I dedicate this thesis to my parents and my wife, without their patience and support, it would not been possible.

Table of Contents

Chapter 1

Introduction

1.1 Background

1.1.1 Evolution of 60 GHz radio technology

The origin of 60 GHz radio can be traced back to the work of J. C. Bose in the 1890s. In 1897, J.C. Bose described to the Royal Institution in London his research carried out in Calcutta at millimeter wavelengths. He used waveguides, horn antennas, dielectric lenses, polarizers and semiconductors at frequencies as high as 60 GHz. More than a century later, a massive unlicensed band was been allocated in the 60 GHz frequency band worldwide for civil usage, as shown in Fig. 1.1.

Driven by this abundance of unlicensed spectrum, advances in CMOS circuit design technology [1] [2] have led to a breakthrough in 60 GHz receiver RF front-ends, which is realized in 45nm digital CMOS with a noise figure of only 6dB. Its small area of only 150x150 μm^2, its low power consumption of 19mA at 1.1V supply voltage, and full-digital control make it highly suitable for phased-array systems [3]. Recently there have been several standards in the 60 GHz band that achieve Gbps throughput. These include IEEE 802.11ad [4], IEEE 802.15.3c [5], Wireless HD [6] and ECMATC48 [7].

1.1.2 Comparison with other unlicensed systems

60 GHz vs. WiFi

The remarkable success of WiFi derives from the opening of ISM bands at 2.4GHz and 5GHz. Nowadays, the 802.11-based WiFi equipment has been widely installed in homes, offices, schools and hot spots around the world. However, with the fast evolution and diversification of wireless multimedia communications, the WiFi bands (2-2.5 GHz and 5-5.8 GHz) have

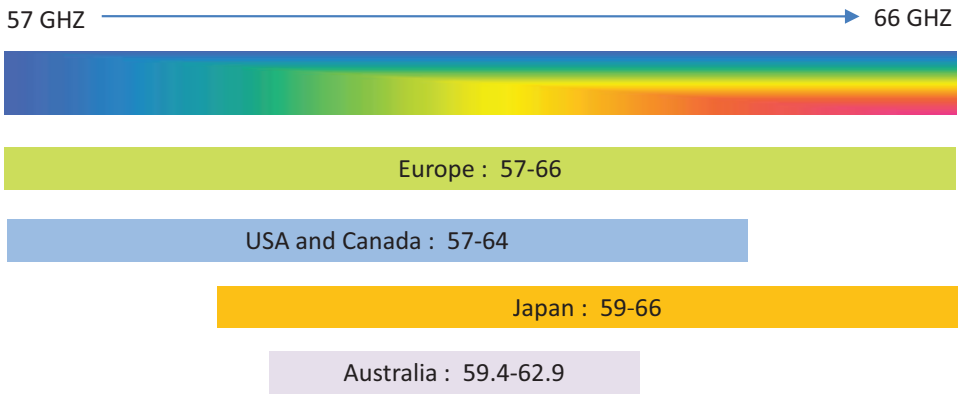


Figure 1.1: Global spectrum allocation for 60 GHz mmWave band

become congested. Although the application of MIMO (Multiple Input Multiple Output) technology can boost the capacity of 802.11n WLAN, there are practical constraints on the number of antennas that can be packed onto a small device.

In contrast, the total bandwidth at 60 GHz band is 7GHz (57-64 GHz), as shown in Fig. 1.1. Moreover, the bandwidth of each channel in 60 GHz is 2.16 GHz [4], which is much larger than the 40MHz channel bandwidth of WiFi. Thus, to achieve 1 Gbps rate, the spectral efficiency required by the 60 GHz is only 0.5 bps/Hz, making it an ideal candidate to support very high data rate applications using simple modulation. Furthermore, the short wavelength of 60GHz facilitates component miniaturization as well as the installation of multiple antennas on compact consumer electronic products.

60 GHz vs. UWB

Although Ultra-Wideband (UWB) technology leverages a broad channel bandwidth, its performance is severely limited due to the maximum power restrictions imposed by Federal Communications Commission (FCC) regulation. Specifically, the EIRP (Equivalent Isotropically Radiated Power) cannot exceed -41.3 dBm under FCC regulations.

In contrast, the spectrum for 60 GHz is less restricted in terms of power; the average EIRP can reach 40 dBm. This can not only overcome the higher propagation loss inherent to mmWave, but also boost the system throughput and transmission range.

Table 1.1: Global transmission power regulation for 60 GHz mmWave band

Region	**TX Power** [mW]	**EIRP** [dBm]	**Antenna Gain** [dBi]
USA	≤ 500	40 (ave); 43 (max)	–
Canada	≤ 500	40 (ave); 43 (max)	–
Japan	≤ 10	–	≤ 47
Australia	≤ 10	52 (max)	–
Europe	≤ 20	57 (max)	≤ 37

1.1.3 60 GHz vision and potential applications

The enormous bandwidth at 60 GHz makes it possible to realize gigabit applications such as uncompressed high definition (HD) video streaming, wireless gigabit Ethernet, super digital home (SDH) and super information portal (SIP) [8, 9]. In particular, the latest commercially available high definition television (HDTV) resolution is 1920×1080 with a refresh rate of 60 Hz. For RGB video with 8 bits/channel/pixel, the required data rate is approximately 3 Gbps [10]. In the future, the progressive scan resolution and refresh rates are expected to enhance the quality of next generation HDTV. This will easily scale the data rate to well beyond 5Gbps. In addition, SDH can provide wireless connectivity at several Gbps for a home/conference room environment, while SIP will enable low-latency read/write operations such that a DVD film can be downloaded within 10 seconds [8]. Examples of other short-range multimedia applications and their required data rates per user are listed in Table 1.2. The large variety of applications suggest that the 60 GHz infrastructure should support real-time traffic with various delay constraints as well as non-real-time traffic with heterogeneous QoS requirements. Flexible network solutions are required to accommodate the large number of communicating devices.

1.1.4 60 GHz radio properties

Free-space propagation loss between isotropic antennas scales as the square of wavelength [11]. Therefore, the propagation loss of 60 GHz is 21.6dB worse than 5 GHz. This makes it impossible for nodes to talk to each other using omni-directional antennas and transmission powers within FCC regulations. Therefore, 60 GHz network devices have to use directional antenna to communicate with each other. Moreover, for the same size of antenna effective

Table 1.2: Global transmission power regulation for 60 GHz mmWave band

Applicatino	**Rate** [Mbps]
Wireless Burglar Alarm	0.01
Indoor Remote Control	0.01
Wireless Embedded Systems in Cars	0.01
Road Pricing	0.1
Wireless Billing	0.1
Communication Between Home Appliances and Internet	0.1
Wireless Video phone	1.5
Wireless Surveillance Cameras	$4-10$
Hospital Bedside Application and Patient Monitor	10
Wireless Interactive Design	$20-40$
Wireless Ad Hoc Communications	$0.1-100$
High-Quality Video Conference	$10-100$
Gaming and Trading Terminal	$50-100$
Wireless IEEE 1394	$100-400$
Wireless High-Resolution Recording Camera	$150-270$
Wireless Virtual Reality Devices	450
WLAN Bridge Connecting Giga-Ethernet LANs	$100-1000$

aperture, antenna gains scale with the inverse of the square of wavelength. Therefore, if both transmitter and receiver use uniform antenna arrays with the same effective aperture size, the receiving power of 60 GHz outperforms 5 GHz by 21.6dB using the same transmission power. Furthermore, since the antenna wavelength of 60GHz is only 5mm, 60Ghz device can have antenna arrays containing more antennas on the device to achieve higher gain and narrower main lobe beamwidth than 2.4/5 GHz.

Although the small wavelength of 60 GHz can help to achieve high directional antenna gain, the small wavelength of 60 GHz electromagnetic wave cannot diffract around humans and furniture. This indicates human body and obstacles penalize the 60 GHz link budget by 20-30dB. In short, the Line-of-Sight (LoS) path between transmitter and receiver can be easily blocked.

1.2 Research challenge

The main challenges of 60 GHz research are how to design Medium Access Control (MAC) protocols and routing protocols that work well with the unique properties of 60 GHz networks,

such as directional antennas, LoS blockages and no message broadcasting due to low quasi-omnidirectional antenna gain.

1.2.1 Medium Access Control

There are five challenges in the design of a good 60GHz MAC protocol. First is link pairing problem in which transmitter and receiver are required to point at each other for successful communication. Second is how to resolve the short-term unfairness and low channel utilization caused by a directional CSMA/CA-based protocol. Third is interference management that tries to avoid collisions at each receiver. Fourth is resolving link blockage. This includes detecting link blockage and differentiating link blockage from the deafness problem. Last is maximizing the spatial reuse of the network.

1.2.2 Routing

There are two challenges in the design of a good 60 GHz routing protocol. First, there is no omni-broadcasting in 60 GHz due to low quasi-omni antenna gains. Therefore, nodes have to send to all directions one-by-one instead of doing a single broadcast. This brings a lot of overhead. As a consequence, routing protocols in Mobile Ad Hoc Networks (MANETS) that exploit omni-directional antennas [12–15] may not be suitable for 60 GHz since they will bring too much overhead due to broadcast sweeps. Second, a 60 GHz routing protocol should fully utilize the high spatial reuse property of directional antennas. Currently, when there are multiple flows in the network, minimum hop routing protocols in MANETs may not always fully utilize the high spatial reuse property of directional antennas. Moreover, high spatial reuse can enable multipath routing on non-interfering node-disjoint path. However, TCP with multipath routing often performs worse than with the single path routing [16].

1.2.3 Link outages by pedestrian blockage

A human body and similarly sized obstacles penalize the 60 GHz link budget by 20-30dB. Therefore, when people are moving, links in 60 GHz networks go on and off frequently due to

human body blockage. This introduces design challenges for both routing and transport protocols. Although outages caused by human blockage in 60 GHz directional antenna networks have some similarities with outages caused by node mobility in omni-directional antenna networks, there are also some differences. When mobility breaks a link in a traditional MANET, the link connection may never come back or it may come back only after a long period. This outage period is difficult to estimate when nodes in the network move randomly using the Random Waypoint model [17]. In comparison, a link that is broken by the blockage of a pedestrian will recover after a short period, since the person will move out of the LoS path a few seconds after creating the blockage. Therefore, a 60 GHz network is likely to have different solutions than a MANET.

1.3 Main contributions and dissertation outline

This dissertation focuses on MAC and network layer design for 60 GHz networks. At the MAC layer, we study the performance of popular CSMA/CA-based directional MAC protocol (DMAC) that is used in 2.4/5 GHz directional networks. We show that the exponential backoff scheme in DMAC causes short-term unfairness and low channel utilization problems. We propose an enhanced DMAC protocol (EDMAC) that does not use an exponential backoff mechanism. Instead, EDMAC employs a low control overhead protocol that enables receivers to adaptively tune senders' contention window sizes. NS-2 simulation results are given to demonstrate that EDMAC compares favorably to DMAC, achieving similar capacity and lower delay jitter in single hop networks, and significantly higher capacity in multihop network scenarios under UDP and TCP traffic.

At the routing layer, we first study the performance of single path routing in 60 GHz networks. We find that the shortest path routing protocol does not always fully utilize the high spatial reuse properties of directional antennas. We propose two heuristic routing algorithms, namely HOP-FP and FP-HOP, that combine the fattest-path and minimum-hop metrics, with and without the consideration of interference. With the consideration of interference, both algorithms outperforms the shortest path algorithm on all topologies.

We then employ multipath routing for 60 GHz networks to fully utilize the high spatial reuse

property of directional antennas. We develop an online node-disjoint path discovery process to find multiple node-disjoint paths between the source and the destination without knowledge of the global topology. From the ns-2 evaluation of different route request (RREQ) transmitting and forwarding schemes, we find that using the "Neighbor Sweep" or the "Neighbor Unicast" transmitting scheme along with the "Not Longer" forwarding scheme is an efficient way to discover a large number of node-disjoint paths in the network.

For the link outage problem, we first study the characteristics of pedestrian blockages. Based on the analytic models and MATLAB simulation results, we propose a two part solution for the link blockage problem; We introduce a blockage timer for the broken path and we employ a the multipath routing scheme. NS-2 simulation results show that when there are fewer people in the network, e.g. 10 people, the proposed two schemes can improve the TCP throughput compared to single path routing.

The remainder of the dissertation is organized as follows. Chapter 2 describes the physical layer model used in the dissertation, including antenna model, propagation model and humane blockage model. Chapter 3 is dedicated to the design of the MAC layer to support the 60 GHz networks. In chapter 4, we study the performance of the proposed EDMAC for TCP traffic in various scenarios, and show that EDMAC outperforms DMAC in all scenarios. Chapter 5 focuses on improving performance of single path routing for the network. Chapter 6 studies multipath routing and a dynamic node-disjoint paths routing protocol is proposed to improve the throughput of TCP flows. Chapter 7 focuses on improving the performance of the network when pedestrians in the network introduce frequent link outages. In the last chapter, we conclude this dissertation.

Chapter 2

Physical Layer Abstract Model

In this chapter, we describe physical layer models used in the dissertation, including antenna model, propagation model and humane blockage model.

2.1 Antenna model

2.1.1 Antenna gain

We consider an antenna located in the origin of a spherical coordinate system. The angle from the x-axis in the xy-plane is $\phi \in [0, 2\pi]$, and the angle from z-axis is $\theta \in [0, \pi]$. The radiation intensity $u(\theta, \phi)$ of the antenna in a given direction (θ, ϕ) is defined as the radiated power per unit solid angle. For lossless antennas, the antenna gain $g(\theta, \phi)$ is defined as the ratio between the radiation intensity in a given direction and the radiation intensity that would be obtained if the same power was radiated isotropically

$$g(\theta, \phi) = \frac{u(\theta, \phi)}{\frac{1}{4\pi} \int_0^{2\pi} \int_0^{\pi} u(\theta, \phi) \sin\theta \, d\theta \, d\phi} \tag{2.1}$$

2.1.2 Antenna array

We study two practical antenna models which are Uniform Linear Array (ULA) and the Uniform Circular Array (UCA). Uniform linear arrays place the antenna elements along z-axis symmetrically of origin, with a distance d between adjacent elements as shown in Figure 2.1(a). All N elements have identical amplitudes but each element i ($i = 1, ..., N$) has a progressive phase shift β to the preceding one, i.e., the phase of i-th element β_i, $\beta_i = \beta_{i-1} + \beta$. Without loss of generality, we regard the yz-plane($\phi = \pm\frac{\pi}{2}$). If we define $\psi(\theta) = kd\cos\theta + \beta$, , where $k = 2\pi/\lambda$ is the angular wave number, λ is the wavelength at 60 GHz, $\lambda = 5$ millimeter, the

radiation intensity for this form of the array fulfills [18]

$$u(\theta) \propto \left(\frac{1}{N} (\frac{\sin(N\psi(\theta)/2)}{\sin(N\psi(\theta)/2)}) \right)^2 \tag{2.2}$$

The directions in which the radiation intensity achieves its maximum are called boresight directions θ_o. With the ULA, the maximum radiation intensity is achieved for

$$\theta_o = \pm \arccos(-\frac{\beta}{kd}) \tag{2.3}$$

Thus, we can set phase of the antenna elements to $\beta = -kd\cos\theta_o$, if we intend to maximize the antenna gain toward certain directions θ_o. This insight enables us to express $\psi(\theta)$ as

$$\psi(\theta) = kd(\cos\theta - \cos\theta_o) \tag{2.4}$$

By substituting Equation (2.2) and Equation (2.4) into Equation (2.1), the antenna gain in the direction θ when the peak of the main beam in direction θ_o, can be computed and simplified by

$$g(\theta) = \frac{2\left(\frac{\sin(Nkd(\cos\theta - \cos\theta_o))}{\sin(kd(\cos\theta - \cos\theta_o)} \right)^2}{\int_0^\pi \left(\frac{\sin(Nkd(\cos\theta - \cos\theta_o))}{\sin(kd(\cos\theta - \cos\theta_o)} \right)^2 \sin\theta \, d\theta} \tag{2.5}$$

In Uniform Circular Array as shown in Figure 2.1(b), the N isotropic elements are equally arranged on a circle ring of the radius R in the xy plane, where the spacing between two neighboring elements is again given by $d = 2R\sin(\pi/N)$. If we assume the phase excitation of the nth element is α_n and the angular position of nth element on xy plane is ϕ_n, The radiation intensity fulfills [18]

$$u(\theta, \phi) \propto \left| \sum_{n=1}^{N} e^{jkR[\sin\theta\cos(\phi - \phi_n) + \alpha_n]} \right|^2 \tag{2.6}$$

To direct the peak of the main beam in (θ_o, ϕ_o) direction, the excitation of the nth element can be chosen to be

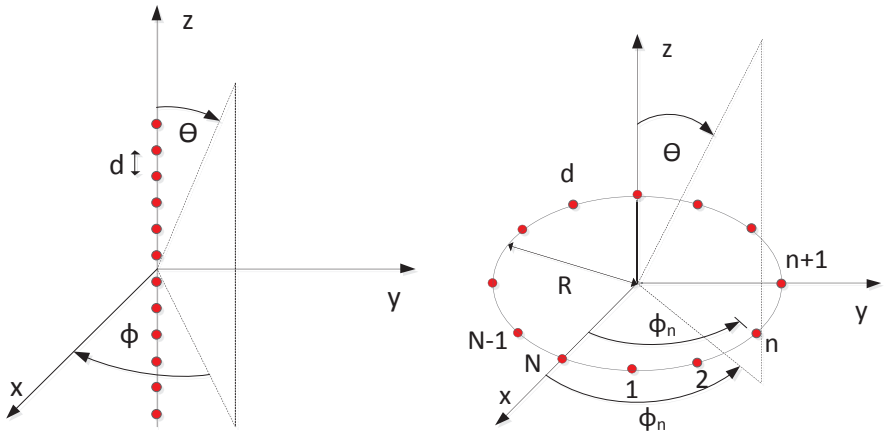
$$\alpha_n = -kR\sin\theta_o\cos(\phi_o - \phi_n) \tag{2.7}$$

Thus the radiation intensity can be written as

$$u(\theta, \phi) \propto \left| \sum_{n=1}^{N} e^{jkR[\sin\theta\cos(\phi - \phi_n) - kR\sin\theta_o\cos(\phi_o - \phi_n)]} \right|^2 \tag{2.8}$$

Table 2.1: Link parameters

Parameter	value
Transmit power P_t	10dB
Path loss at 1m PL_0	-68dB
Path loss exponent n	2
Implementation loss I_L	5dB
Body blockage loss B_L	30dB



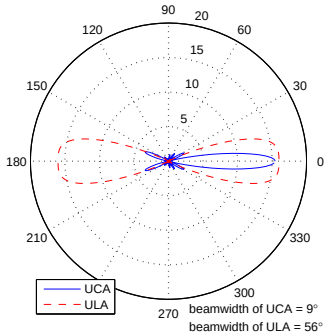
(a) Uniform Linear Array (b) Uniform Circular Array

Figure 2.1: Illustration of ULA and UCA. Definition of angles ϕ and θ.

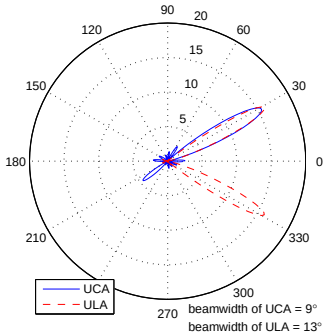
Therefore, the antenna gain in the direction (θ, ϕ) when the peak of the main beam in direction (θ_o, ϕ_o), can be computed

$$g(\theta,\phi) = 4\pi \frac{\left|\sum_{n=1}^{N} e^{jkR[\sin\theta\cos(\phi-\phi_n)-kR\sin\theta_o\cos(\phi_o-\phi_n)]}\right|^2}{\int_0^{2\pi}\int_0^{\pi}\left|\sum_{n=1}^{N} e^{jkR[\sin\theta\cos(\phi-\phi_n)-kR\sin\theta_o\cos(\phi_o-\phi_n)]}\right|^2 \sin\theta\, d\theta\, d\phi} \tag{2.9}$$

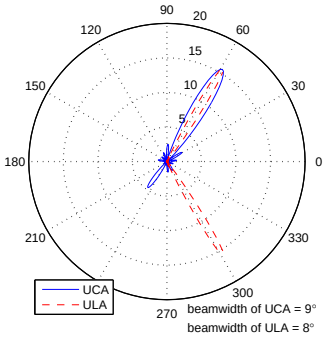
In this dissertation, we only consider two dimensional network, i.e., the azimuthal cophasal pattern which lies in xy plane, $\theta = \theta_o = \pi/2$. The gain patterns of a UCA and ULA with 16 antenna elements are shown in Figure 2.2. In this dissertation, we use UCA antenna since it has same main lobe width and small side lobe gain for all directions.



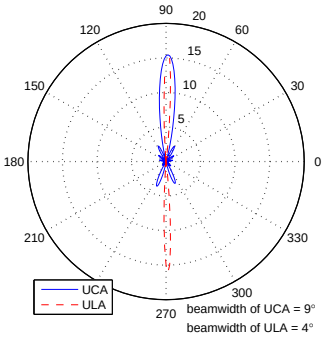
(a) $\theta_0 = 0°, \phi_0 = 90°$ for ULA, $\theta_0 = 90°, \phi_0 = 0°$ for UCA



(b) $\theta_0 = 30°, \phi_0 = 90°$ for ULA, $\theta_0 = 90°, \phi_0 = 30°$ for UCA



(c) $\theta_0 = 60°, \phi_0 = 90°$ for ULA, $\theta_0 = 90°, \phi_0 = 60°$ for UCA



(d) $\theta_0 = 90°, \phi_0 = 90°$ for ULA, $\theta_0 = 90°, \phi_0 = 90°$ for UCA

Figure 2.2: Gain patterns of UCA and ULA for antenna array with 16 elements.

2.2 Link propagation model

The Line of Sight (LOS) is the case if there is no obstacle between transmitter and receiver. The received power(in dB) at the receiver side in LOS case is represented as,

$$P_r(d) = P_t + G_t + G_r - 10n\log(d) - PL_0 \tag{2.10}$$

where P_t is the transmitting power, G_t and G_r are the antenna gains of the transmitter and receiver as in Equation 2.9, n is the path loss exponent which equals 2, d is the distance from the transmitter and receiver, PL_0 is the reference path loss when $d = 1m$. If there are n_p number of people between the transmitter and the receiver, the receive power will be further decreased by $n_p B_L$, where B_L is the loss of the signal when block by a person which is about 20dB in 60 GHz. The typical value of above parameters are listed in Table 2.1, as referred from

IEEE standard 802.11ad.

Chapter 3

Enhanced Directional MAC for 60 GHz Networks

3.1 Introduction

Because of high propagation losses in 60 GHz channels, nodes cannot use omni-directional antennas to communicate with each other due to the insufficient antenna gain. The transmitter and the receiver have to use directional antennas and point their main beams toward each other to achieve high data rate transmission. The link pairing problem is how to coordinate the transmitter and the receiver point at each other when transmitting packets. Here we define "point at each other" to mean both the transmitter and the receiver aim their main lobes toward each other. The transmitter can point the main beam at the receiver if we assume that an initialization process enables each node to learn the direction of all nodes in the network. The key issue is how to let the receiver know the transmitter is transmitting to it in order to point the main beam back toward the transmitter. While some existing techniques such as Direction of Arrival (DoA) or Angle of Arrival (AoA) estimation can help the receiver learn the directional of transmitter. However, these techniques are too complex in terms of computation. The first challenge of 60 GHz network MAC protocol is to choose a simple solution for the link pairing problem using phased array antennas.

The second challenge is the carrier sensing mechanism, which is the key feature of CS-MA/CA based protocol in omni-directional antenna networks, does not work well in 60 GHz networks. The wavelength at 60 GHz is only 5 millimeters, which is only one twelfth of that at 5 GHz. Thus, for the same area size antenna array, we can put more antenna elements at 60 GHz comparing to 5 GHz. Directional antenna arrays on 60 GHz devices can achieve narrower main lobe beamwidth and smaller side lobe gain comparing to 5 GHz. This feature along with the high propagation loss mentioned in the previous paragraph make the interference range very small when nodes are transmitting. The benefit of this property is that the network can

achieve high spatial reuse, but the disadvantage is that the node cannot detect other ongoing transmissions due to low qusi-omni directional gain when the node is idle. Therefore, a node seeking to transmit will always sense the channel idle and decrease the backoff timer. We refer to this effect as *directional deafness*. Because of directional deafness, nodes that have packets destined to the same node are hidden terminals of each other. It is harder to solve the hidden terminal problem in directional network since nodes cannot broadcast extra control information like RTS and CTS in omni-directional network. We propose a solution to resolve this problem after we finish a detail study about issues in using the CSMA/CA based protocol in section3.3.

3.2 Related works

To resolve the link pairing and directional deafness problems, MAC protocols for 60 GHz networks include both industry standards such as IEEE 802.11ad [4] and IEEE 802.15.3c [19], and academic research [20] [21]. We classify them as centralized or distributed.

In the centralized protocols [20] [4] [19], a network coordinator polls the traffic demand from each node, and pushes a link pairing schedule back to the nodes. Thus, all nodes know which node to transmit to or receive from in each slot and do not suffer from link paring and deafness problems. A benefit is that the coordinator can use its knowledge of the scheduling demands to achieves high spatial reuse. In [20], the author proposes Frame-based Scheduling directional MAC (FDMAC), which uses a low-complexity greedy coloring algorithm to compute near-optimal schedules with respect to the total transmission time. But the drawback of this protocol is a lack of robustness. In particular, 60 GHz links are easily blocked by objects such as walls and human bodies. If the link between the coordinator and a network node is broken, the node will not know in which slot to transmit or receive. This drawback pushes us to find a robust distributed MAC protocol for 60 GHz network.

There are both TMDA and CSMA based distributed MAC protocols. In a TDMA based protocol, the link pairing and deafness problem can be solved by a synchronous time-slotted reservation based protocol [21]. The key idea in [21] is to use memory to achieve implicit coordination among the mesh nodes despite deafness. Each node persists in using the transmit or receive slots that have been successfully used for a given neighbor, and puts slots over which it

has been unsuccessful on a per-neighbor blacklist. Thus, a node's transmit and receive history with each of its active neighbors provides feedback that is used for implicit coordination, and persistent use of a given slot for transmitting to a given neighbor leads to an approximate TDM schedule. The drawback of these kind of TDMA based protocols is that they need assistant synchronization devices like GPS or a complex synchronization protocol to make sure the time slots are synchronized. This drawback motivates us to find an easy implemented unsynchronized distributed protocols.

CSMA-based directional MAC protocols have been well studied in 2.4/5 GHz but not for 60 GHz. The existing solutions for $2.4/5$ GHz networks focus on adding global control information to solve the deafness problem like using omni-directional RTS/CTS in [22], using circular RTS/CTS in [23] or using a global control channel like ToneDMAC [24]. The key idea is the receiver tells its intended sender it will be deaf in next certain period [22] [23] or that it is not deaf now [24]. The intended sender will stop or reset its backoff timer when receiving these messages to eliminate deafness problem. However, these protocols all have some shortcomings in 60 GHz. First, it is hard to add global control information in a 60 GHz network as in [24] [22] because of the extremely low gain in quasi-omni pattern, the receiver has to use directional antenna to send a deafness notification. Second, adding control messages can bring considerable protocol overhead. For examples, in [23], the receiver has to transmit a deafness message to all directions one by one. Although the authors in [25] give an enhanced solution in which a receiver only sends deafness notifications to nodes that it recently received a packet from, the control overhead will increase dramatically with the number of intended senders. Moreover, nodes that are not in the receiver deafness notification list or miss the notification still suffer from the deafness problem, which decreases the channel utilization. These shortcomings highlight the need to look for a directional CSMA MAC protocol that avoids broadcasting broadcasting deafness notification.

3.3 Link paring and deafness problem

In 60 GHz networks with phased array antennas and CSMA based protocol, a transmitter and a receiver can point at each other by exchanging beamforming training packet, as in the IEEE

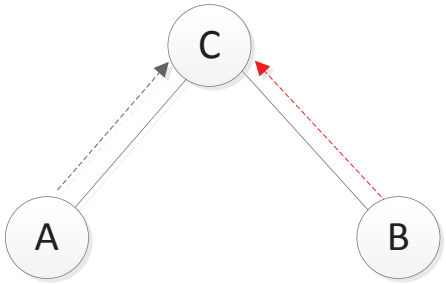


Figure 3.1: A basic scenario in 60 GHz network: both node A and node B send packets to node C via one hop.

802.11 standard [4]. Therefore, we model the beamforming training process as a directional RTS/CTS handshake before sending the data packet. In particular, we assume that a node has learned the directions of all its neighbors, and that when a node is in idle state, it will employ a quasi-omni antenna pattern. As defined in the IEEE 802.11 standard [4], a quasi-omni antenna pattern is an antenna pattern that covers all the directional antenna patterns generated by the phased array although gains for all directions are not exactly identical. This permits a node to keep sensing incoming packets from all directions. As shown in Fig. 3.2, when node A wants to send a packet to node C, node A will switch the main beam toward node C and send an RTS packet first. If node C is in idle, it will receive the RTS packet and switch the main beam toward node A and send a CTS packet back. After node A receives the CTS packet, it will send a data packet to node C, and node C will send an ACK packet to node A. After transmitting the ACK packet, the antenna array of node C will go back to quasi-omni pattern while the antenna array of node A will go back to quasi-omni pattern also after successfully receiving the ACK packet from node C. If node A fails to receive a CTS or ACK packet from node C, it will go back to quasi-omni mode after the timeout.

Deafness becomes an issue when both nodes A and B have packets to send to node C as shown in Fig. 3.1. When nodes A and C point at each other and start a transmission, node B, unable to sense that transmission due to low quasi-omni antenna gain, may transmit an RTS packet to node C after its backoff timer expires as shown in Fig. 3.2. Because the main lobe gain of a directional antenna is usually much larger than the side lobe gain, the packet from node B cannot create enough interference to cause a collision at node C and will be dropped. The basic DMAC use the exponential backoff mechanism as in legacy 802.11 that

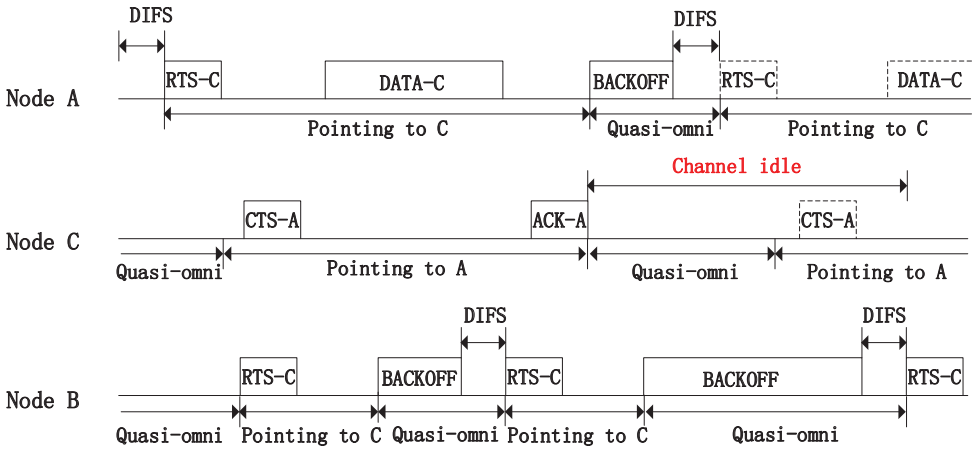


Figure 3.2: Unfairness and Low channel utilization of basic DMAC.

a node will double its contention window size with each failed retransmission until it reaches the maximum window size. The exponential backoff mechanism efficiently resolves collisions in an omni-directional antenna network since collision is essentially the only reason that a transmission fails. But in a directional antenna network, deafness becomes the main reason for a transmission failure [25]. Moreover, after a transmission failure, a node such as B does not know whether the failure is due to a collision or the deafness of the receiver. Therefore, when a transmission fails due to deafness, doubling the contention window will not help to increase the success probability of a node's next retransmission. Instead, it just increases delay. The example in Fig. 3.2 demonstrates this effect. If the initial contention window of node B is small, its backoff timer has high probability to expire while node A is still transmitting. In this case, node B will have a transmission failure and will double its contention window. However, node A does not increases its contention window after finishing its current transmission. This creates unfairness as it gives increased priority to node A to transmit over node B. Thus a successful transmitter, say node A, is able to hog the channel and transmit a number of packets as fast as it can before other nodes, say node B can grab the channel again. Therefore, the delay of packets is small when a node hogs the channel and is bigger when other nodes hog the channel. Thus, the variance of delay, often refereed to as jitter, is larger. The channel utilization also decreases because after node A releases the channel, node B is still in an unnecessarily large backoff period. This long backoff period will decrease the channel utilization of the network.

The unfairness and low channel utilization problems of 60 GHz network using a carrier sensing protocol can be resolved if we let nodes that send to the same receiver use the same contention window size W, independent of past transmission success or failure. In the next section, we will study the performance of such a CSMA protocol that all nodes with same contention window size in 60 GHz.

3.4 Capacity analysis of CSMA protocol that uses fixed contention window

In this section, we study the capacity of CSMA/CA protocol in which all nodes sending to the same receiver employ the same contention window size. In our model, a finite number, n, of transmitters are located around one receiver at different directions. We assume that the beamwidth of each sender's main lobe is quite narrow. Specifically, if the receiver points at a transmitter, then there are no other transmitters in the main lobe of the receiver. We also assume the side lobe gain of the directional antenna pattern and the quasi-omni antenna pattern are sufficiently small that no transmitter can sense the activity of the other $n-1$ transmitters. Thus, all n transmitters operate as hidden terminals of each other. We are not interested in the case that transmitters can sense other nodes' transmissions, since such a network will become equivalent to a traditional CSMA network. We are also not interested in the case that nodes are in the same main lobe of the receiver, since nodes in the same main lobe of the receiver can receive the CTS packet from the receiver which alleviates the hidden terminal problems as in traditional CSMA networks. We also assume that the network operates in a saturation mode; in other words, each of the n transmitters always has a packet ready to transmit.

It is not straightforward to analyze the capacity of the network using existing models. First, the system is asynchronous since no node can sense the other nodes' transmissions. Thus, the capacity analysis models for IEEE 802.11 MAC in [26] and [27] based on the assumption of synchronized slots do not work well. Second, the backoff procedure is discrete. A backoff timer is initialized to several number of slots. The backoff timer deceases by one slot, if the node senses the channel idle during that slot. Thus the capacity analysis model for p-persistent CSMA in [28] also does not work well. In this section, we propose a discrete-time model similar to [27] and a continuous model similar to [28], with modifications for the node directional

deafness and capture properties, to approximately model the network.

3.4.1 Discrete-time model

In the discrete-time model, we assume all nodes are synchronized to simplify the analysis and give us a coarse throughput estimate. As in the 802.11 MAC protocol, time is slotted and the length of each slot is 3 us in our model. A packet transmission generally requires multiple slots. A slot can be an idle slot, a period associated with successful transmission, or a period associated with collision. Although nodes have a fixed contention window size W and choose a backoff time that is uniformly distributed over $[1, W]$, we adopt a memoryless approximate model as in [27] that nodes choose a backoff time sample from a geometric distribution with parameter p. This model will have a smaller error if the length of a slot is shorter, W is bigger and the number of transmitters in the network n is bigger. In order to match the expected backoff time under uniform backoff, the memoryless approximated model assumes that the transmission probability for each node in any time slot is [27]

$$p = \frac{2}{W+1} \tag{3.1}$$

If we define P_{i} as the probability that a time slot is idle, P_{s} as the probability that there is one successful transmission, and P_{c} as the collision probability that at least two RTS packets overlap at the receiver, we can obtain the following equations

$$P_{\mathrm{i}} = (1-p)^n, \tag{3.2}$$

$$P_{\mathrm{s}} = np(1-p)^{n-1}(1-p)^{n-1}, \tag{3.3}$$

$$P_{\mathrm{c}} = 1 - P_{\mathrm{i}} - P_{\mathrm{s}}. \tag{3.4}$$

Note that the P_{s} in Eq. (3.3) has an additional factor $(1-p)^{n-1}$ from that in the omni-directional model in [27]. Since nodes cannot sense other nodes' transmissions, a successful transmission occurs if and only if one node contends in the slot, and the other $n-1$ nodes keep silent, not only in the current slot, but also in the next slot because we assume the time to transmit RTS is two slots instead of its real value of 1.42 slots. This is a conservative assumption that the longer RTS is, the higher collision probability is and the lower the throughput is.

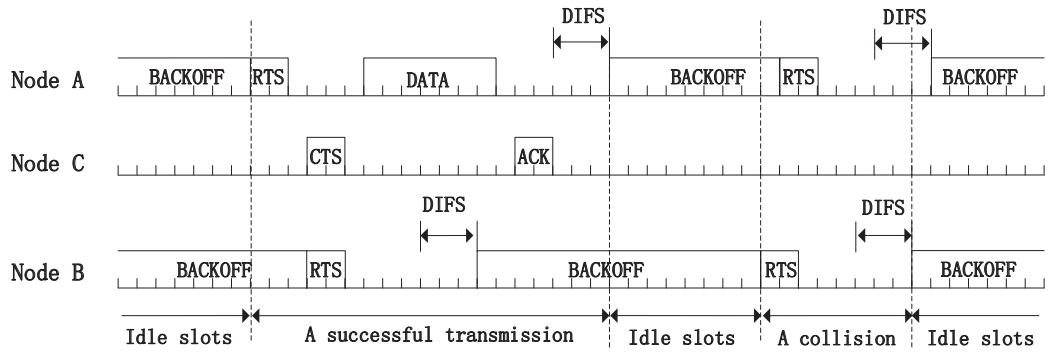


Figure 3.3: Illustration of discrete-time model.

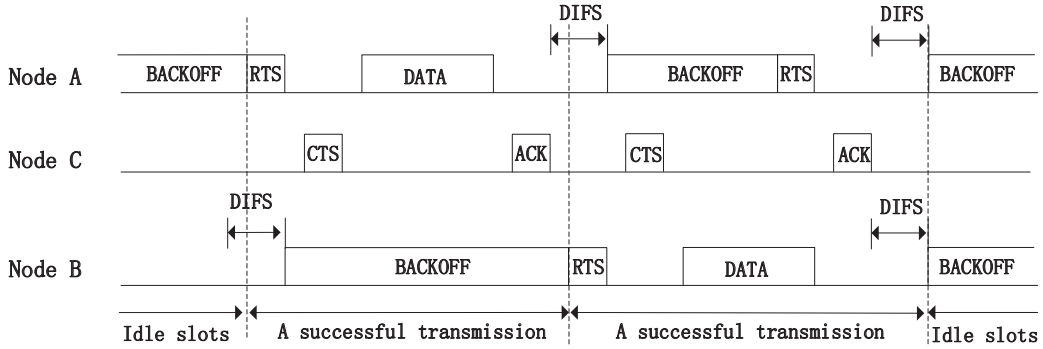


Figure 3.4: Illustration that the $(i+1)$th successful transmission can immediately start after the ith transmission finished.

We define T_{transmit} as the time duration of a successful transmission, T_{Coll} as the time duration wasted by a packet collision, T_{Idle} as the time duration of an empty slot time T_{slot} and T_{data} as the time duration needed to transmit a data payload. From [27] we know the throughput of the network is

$$S = \frac{P_{\text{s}} T_{\text{data}}}{P_{\text{s}} T_{\text{transmit}} + P_{\text{c}} T_{\text{Coll}} + P_{\text{i}} T_{\text{Idle}}} \tag{3.5}$$

To compute T_{data}, we assume that data packet lengths are fixed to simplify the analysis; we can later extend the evaluation when packet lengths have other distributions as in [26]. T_{transmit} includes the time to transmit RTS, CTS, data and ACK packet plus SIFS between them as shown in Fig. 3.3. Thus,

$$T_{\text{transmit}} = T_{\text{DIFS}} + T_{\text{RTS}} + T_{\text{SIFS}} + T_{\text{CTS}} + T_{\text{SIFS}} + T_{\text{data}} + T_{\text{SIFS}} + T_{\text{ACK}}. \tag{3.6}$$

Note that the transmission period T_{transmit} is followed by T_{DIFS} as in the legacy IEEE 802.11

standard because there will be nodes that are able to hear and interfere with each other occasionally and thus we need traditional CSMA contention resolution. This also indicates that our protocol is fully compatible with legacy IEEE 802.11 standard. Note that T_{transmit} in Eq. (3.6) that contains T_{DIFS} is a conservative estimate of T_{transmit} since in the 60 GHz network, nodes cannot sense other nodes' transmissions and always sense the channel to be idle due to the directional deafness. Thus, they may start the $(i+1)$th transmission immediately after the ith transmission finished instead of waiting for T_{DIFS} as in omnidirectional model in [27] as shown in Fig. 3.4. Note that, Eq. (3.6) is more accurate when n is smaller.

The collision period is

$$T_{\text{Coll}} = T_{\text{DIFS}} + T_{\text{RTS}} + T_{\text{CTS}}. \tag{3.7}$$

Note that similar to T_{transmit} in Eq. (3.6), the collision period T_{Coll} in Eq. (3.7)is also a conservative estimate and is more accurate when n is smaller.

Applying (3.1)-(3.4), (3.6) and (3.7) to (3.5), we can obtain the throughputs of the network as a function of the contention window size W using the discrete-time model.

3.4.2 Continuous time model

As mentioned earlier, a successful transmission in a 60 GHz network can only occur when a transmitter starts sending the RTS to an idle receiver and no other nodes start sending during that RTS period. We assume that each backlogged packet repeatedly attempts to retransmit at randomly selected times separated by independent, exponentially distributed random delays τ, with $E[\tau] = 1/\lambda$. In our model, the average interval between two transmissions is $(W+1)T_{\text{slot}}/2$. Therefore, we have

$$\frac{1}{\lambda} = \frac{(W+1)T_{\text{slot}}}{2} \tag{3.8}$$

Since all n nodes have backlogged packets, the time until the first transmission starts is an exponentially distributed random variable with rate $n\lambda$.

Similar to [28], we define a busy period as the time between two consecutive idle periods, in which there is at least one transmission to the receiver. As shown in Fig. 3.5, a busy period always starts with an RTS packet and it might be a period with successful data transmission, or possibly a period with RTS packet collisions. The throughput at the receiver, as discussed by

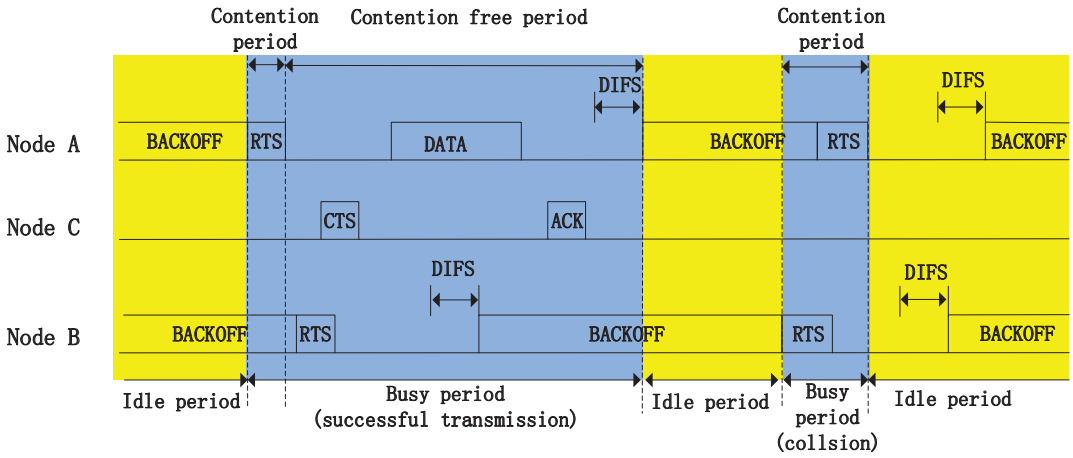


Figure 3.5: Illustration of idle period and busy period that is consisted of RTS contention period and contention free period in the continues model.

Kleinrock and Tobagi in [28], can be expressed as

$$S = \frac{\bar{U}}{\bar{B} + \bar{I}}, \tag{3.9}$$

where $\bar{U}$, $\bar{B}$, and $\bar{I}$ are the average utilization time for a data packet transmission, the average busy time, and the average idle time of the channel, respectively, in each cycle.

As shown in Fig. 3.5, an RTS packet originating from any transmitter is successfully received if no other RTS packets are sent in the RTS period denoted T_{RTS}. This is because the receiver will switch the main lobe to the transmitter after it receives the RTS packet, and any other RTS packets coming from a side lobe direction will be neglected. When the RTS packet is successfully received at the receiver, it will send the directional CTS packet to the transmitter and then wait for the data packet. We argue that when the RTS packet is successfully received, data packet reception will be guaranteed because we assume there is only one transmitter in the main lobe direction of the receiver. Therefore, the busy period that shaded in blue in Fig. 3.5 consists of RTS contention periods and contention free periods.

The average duration of the RTS contention period $E[T_{\text{c}}]$ is given by Lemma 1,

Lemma 1. *The average time duration of a contention period is*

$$E[T_c] = \frac{(e^{n\lambda T_{RTS}} - 1)}{n\lambda}. \tag{3.10}$$

Proof. If we assume the characteristic function of T_{c} is $L_c(s) = E[e^{-sT_{\text{c}}}]$. We expand the expectation using the law of iterated expectations by conditioning it on two categories of events.

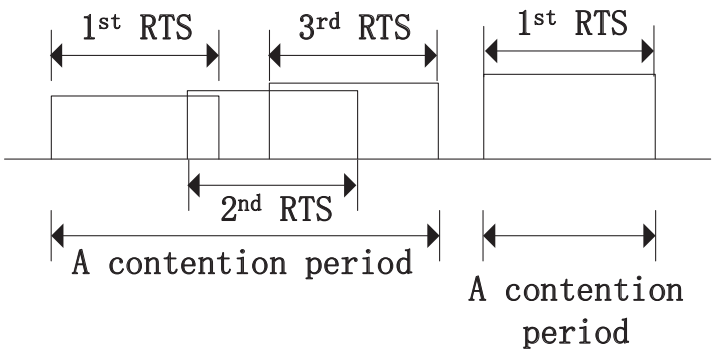


Figure 3.6: Example of a contention period.

Without loss of generality, we set the first RTS packet arrival time to 0 and assume the arrival rate of the system $n\lambda$ is a constant value. In the first category, there is no arrival of other RTS packets during $[0, T_{\text{RTS}}]$. This happens with probability $e^{-n\lambda T_{\text{RTS}}}$ and if this happens $T_{\text{c}} = T_{\text{RTS}}$ as shown in Fig. 3.7(a). In the second category, we assume there is no arrival of other RTS packets during $[0, t]$ $(t < T_{\text{RTS}})$ and then during $(t, t + dt]$, there is one RTS packet arriving as shown in Fig. 3.7(b). These two events jointly happen with probability $e^{-n\lambda t}(n\lambda dt)$ because they are independent. But if this joint event occurs, we can simply regard the newly arrived RTS packet as the first packet since from now on further contention can happen only during its transmission time $(t, t + T_{\text{RTS}})$. In other words, from time t on, the contention period is extended by a time whose length has the same distribution of T_{c}. The entire contention period conditioned on the category two events is therefore $t + T_{\text{c}}$ as shown in Fig. 3.7(a). We can write

$$L_c(s) = E[e^{-sT_c}] \tag{3.11}$$

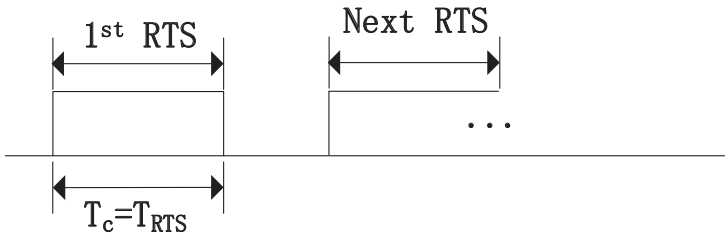
$$= e^{-n\lambda T_{\text{RTS}}} E[e^{-sT_{\text{RTS}}}] + \int_0^{T_{\text{RTS}}} e^{-n\lambda t}(n\lambda dt) E[e^{-s(t+T_{\text{c}})}] \tag{3.12}$$

$$= e^{-n\lambda T_{\text{RTS}}} e^{-sT_{\text{RTS}}} + \int_0^{T_{\text{RTS}}} e^{-n\lambda t}(n\lambda) e^{-st} E[e^{-sT_{\text{c}}}] dt \tag{3.13}$$

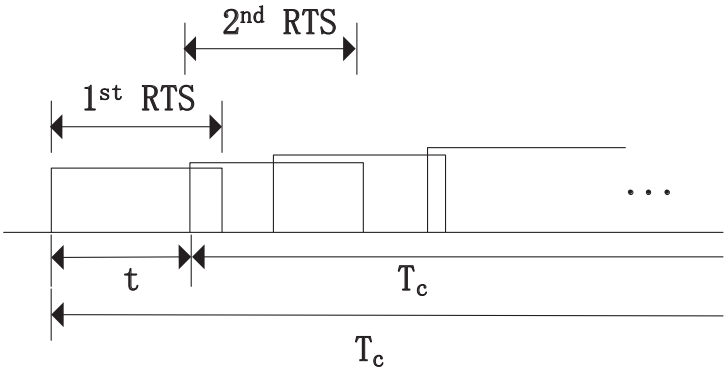
$$= e^{-(n\lambda+s)T_{\text{RTS}}} + L_c(s) \int_0^{T_{\text{RTS}}} n\lambda e^{-(n\lambda+s)t} dt \tag{3.14}$$

Solve for $L_c(s)$ in (3.11), we get

$$L_c(s) = \frac{n\lambda + s}{n\lambda + se^{(n\lambda+s)T_{\text{RTS}}}} \tag{3.15}$$



(a) First category: no arrival of other RTS packets during $[0, T_{\text{RTS}}]$



(b) Second category: the second RTS packet arrival during $(t, t + dt]$

Figure 3.7: Average contention period definition and calculation.

The average time duration of a contention period can be computed as

$$E[T_{\text{c}}] = -dL_c(0)ds|_{s=0} = \frac{(e^{n\lambda T_{\text{RTS}}} - 1)}{n\lambda}. \tag{3.16}$$

□

The probability P_{s} that a transmitter successfully sends a packet is the probability

$$P_{\text{s}} = e^{-n\lambda T_{\text{RTS}}} \tag{3.17}$$

that there is no RTS packet arrival during that T_{RTS} period. With probability P_{s}, the contention period will be followed by a contention-free period of duration

$$T_{\text{s}} = T_{\text{transmit}} - T_{\text{RTS}} = T_{\text{DIFS}} + T_{\text{SIFS}} + T_{\text{CTS}} + T_{\text{SIFS}} + T_{\text{data}} + T_{\text{SIFS}} + T_{\text{ACK}} \tag{3.18}$$

which is the time needed to successfully transmit a data packet excluding T_{RTS} which is already counted in the contention period. Otherwise, with probability $1 - P_{\text{s}}$, no contention-free period follows the contention period. Therefore, the average duration of a contention free period is

$$E[T_{\text{cf}}] = P_{\text{s}}T_{\text{s}} \tag{3.19}$$

Table 3.1: Link parameters

Parameter	value
Transmit power P_t	10dB
Path loss at 1m PL_0	-68dB
Implementation loss I_L	5dB
Carrier Sense Threshold	-48dBm
Header duration	4μs
Slot time	3μs
SIFS	3μs
DIFS	9μs
Data rate	1155Mbps
RTS	4.5μs
CTS, ACK	4.3μs

and the average duration of busy period is

$$\bar{B} = E[T_{\text{c}}] + E[T_{\text{cf}}]. \tag{3.20}$$

The average utilization time is the product of the probability of a successful transmission and the data packet transmission time

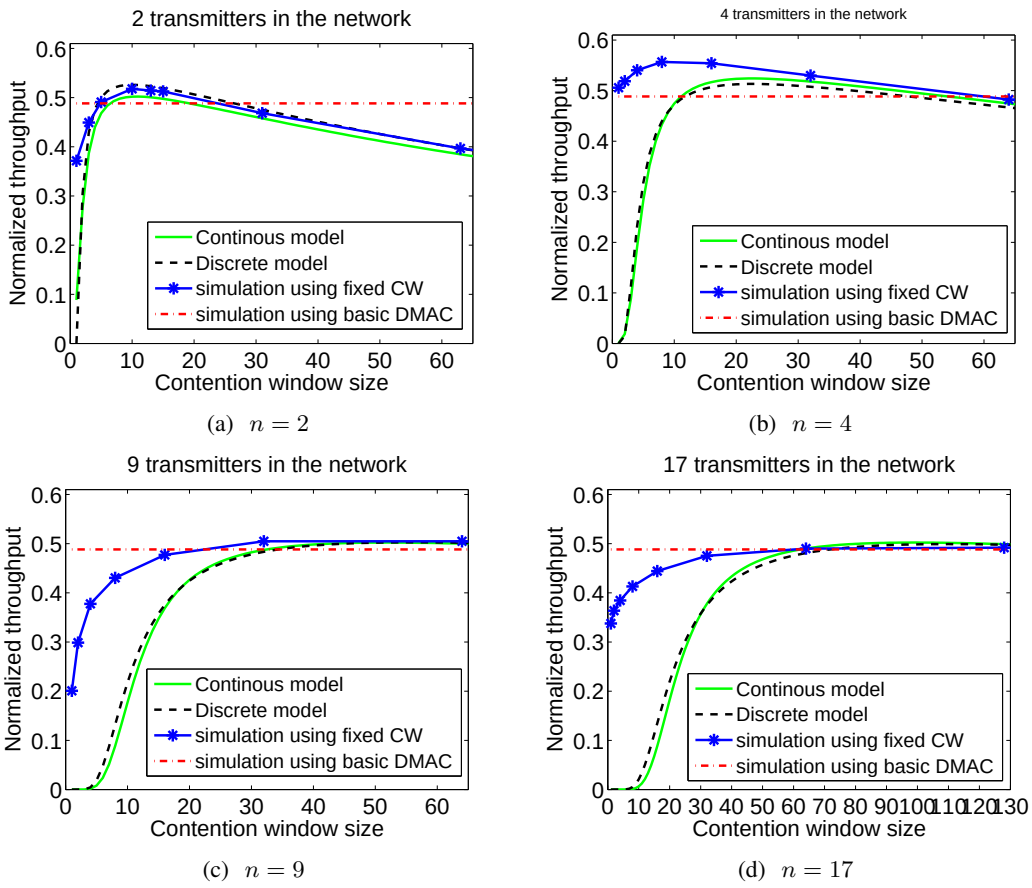
$$\bar{U} = P_{\text{s}} T_{\text{data}}. \tag{3.21}$$

The average duration of the subsequent idle period (highlighted in yellow in Fig. 3.5) is the average inter-arrival time of RTS packets from all nodes given by

$$\bar{I} = \frac{1}{n\lambda}. \tag{3.22}$$

Substituting (3.20)-(3.22) into (3.9), we obtain the channel throughput of the protocol in the discussed network model

$$S = \frac{P_{\text{s}} T_{\text{data}}}{P_{\text{s}}(T_{\text{transmit}} - T_{\text{RTS}}) + E[T_{\text{c}}] + \frac{1}{n\lambda}}. \tag{3.23}$$



(a) $n = 2$ (b) $n = 4$

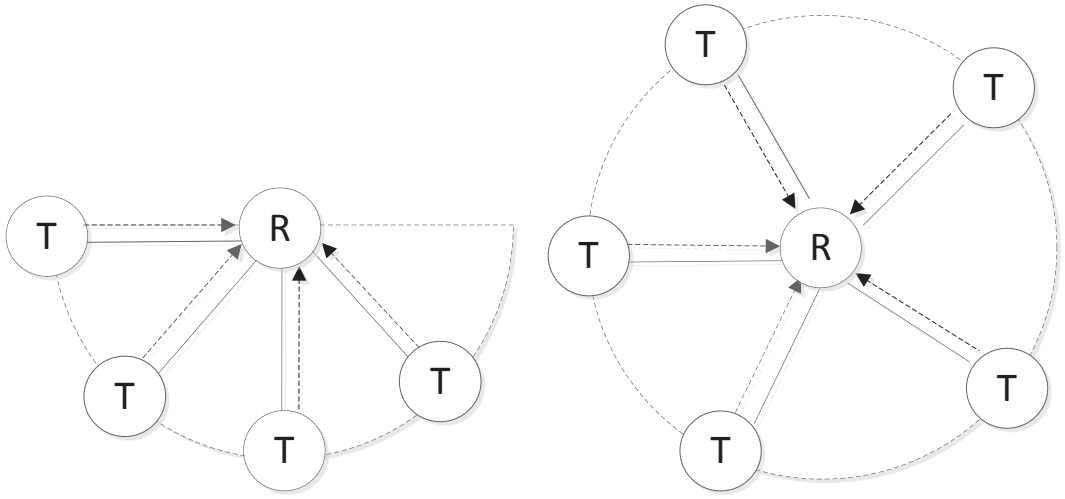
(c) $n = 9$ (d) $n = 17$

Figure 3.8: Normalized throughput of a saturate network with size of contention window increase, with n transmitters uniformly located on a circle when n is odd or semi-circle when n is even centered at the receiver with radius 12 meters.

3.4.3 Simulation validation

We use ns-2 simulator to validate above analytical models. In the simulation, the payload length is 8000 bytes and the system parameters are shown in Table 3.1. All nodes are equipped with uniform circular antenna array with 16 antenna elements and a main lobe beamwidth of 9°. All nodes are located on the xy plan. The receiver is located at the point with coordinate (r, r) and a finite number, n, of transmitters are uniformly located on a circle if n is odd or on a semi-circle if n is even centered at receiver with radius r as shown in Fig. 3.9. Thus, the coordinate of the ith node

$$(x_i, y_i) = \begin{cases} r(1+\cos(\frac{\pi+\pi*(i-1)}{n}), r(1+\sin(\frac{\pi+\pi*(i-1)}{n})), & \text{if } n \text{ is even} \\ r(1+\cos(\frac{\pi+2\pi*(i-1)}{n}), r(1+\sin(\frac{\pi+2\pi*(i-1)}{n})), & \text{if } n \text{ is odd} \end{cases} \tag{3.24}$$



(a) $n = 4$ (b) $n = 5$

Figure 3.9: Single hop scenario: n transmitters are uniformly located on a circle when n is odd or semi-circle when n is even centered at the receiver.

This topology guarantees that no three nodes are in a line and no transmitter can sense other nodes' transmissions. This topology also guarantees that all RTS packets arriving at the receiver have the same power $P(r)$ after attenuation to eliminated the near/far effect as discussed in [29]. We choose a bigger r, for example $r = 12$ meters in the simulation, to eliminate the capture effect as discussed in [30]. Otherwise, if r is too small, for example, less than 10 meters, the receiving power of a RTS packet is much higher than the noise level N_l. Therefore, if the second RTS packet arrives at the receiver while the receiver is receiving the first RTS packet, the Signal to Interference Ratio (SINR) of the first packet, i.e., $P(r)/(P(r) + N_l)$, will be bigger than 0.63 which is the SINR threshold of successfully decoding a RTS packet. This means the first RTS will be successfully received even there is an overlap between two RTS packets. This contradicts with our assumption in our analysis models that a collision will happen if there is an overlap between two RTS packets.

The saturated throughput of the network when $n = 2, 4, 9, 17$ calculated by the discrete-time (bold dashed line) and continues models (bold solid line) are shown in Fig. 3.8. The saturated throughput measured by ns-2 simulations using fixed contention window size is marked using asterisk as shown in Fig. 3.8. In Fig. 3.8, the gap between analytical model and the simulation results is bigger when contention window size is small. This is because the memoryless assumption of the backoff timer is not accurate when contention window size is small. We also

Table 3.2: Neighbor table of node A

ID	Direction	W	link activity
B	0	1	Not Active
C	45	8	Not Active

plot the simulation results using DMAC using dash-dot line. We can observe that the maximum network throughput using fixed contention window size is bigger than using DMAC especially when n is small like $n = 2, 4$. This is because the optimal contention window size that achieves the maximum throughput when $n = 2$ is much smaller than the average contention window size used in DMAC. Smaller contention window size means the interval between two transmission is small and thus the system has higher throughput. This motivates us to design an enhanced DMAC protocol which let all nodes use same optimal contention window size W when sending packets to the same receiver in the next section.

3.5 Enhanced directional MAC for 60 GHz networks

In this section, we first propose EDMAC that not only let all nodes use the same contention window size W when sending packets to the same receiver, but also an algorithm that can help a node to quickly find the optimal W that its neighbors should use when sending packets to it. In EDMAC, each node maintains a neighbor table with a record for every neighbor. The first column of the table is the name of the neighbor. The second column is the direction that the neighbor is located. This "direction" data is dependent on the node's underlying antenna technology. For example, in a beamforming antenna array, direction is represented by a set of combining weights. The third column is the contention window size the node should use when it transmits a packet to its neighbors. The last column indicates whether the node has received packets from the neighbor recently. As an example, Table 3.2 shows the neighbor table of node A. The second record in the table indicates node A can transmit to or receive from node C at $45°$ direction. When node A transmits a packet to node C, it will choose $W = 8$ as its fixed contention window size. The link from node C to node A is inactive since node A has not received a packet from node C recently.

To enable all neighbors to use the same contention window when sending a packet to the same receiver, the receiver will put the contention window size information in the Acknowledgement (ACK) packet after it receives a data packet from a neighbor. After receiving the ACK packet, the neighbor will update the value of contention window size in its neighbor table.

In the previous section, both analytical and simulation results show that there is an optimal contention window size to achieve the maximum throughput at the receiver when all transmitters have backlogged packets to transmit. To find the optimal contention window size which maximizes the throughput (3.23), we divide both numerator and denominator of (3.23) by P_s. We have

$$S = \frac{T_{\text{data}}}{(T_{\text{transmit}} - T_{\text{RTS}}) + \frac{E[T_{\text{c}}]}{P_{\text{s}}} + \frac{1}{n\lambda P_{\text{s}}}} = \frac{T_{\text{data}}}{(T_{\text{transmit}} - T_{\text{RTS}}) + D(\lambda)}. \tag{3.25}$$

We can see that the maximum throughput will be achieved when

$$D(\lambda) = \frac{E[T_{\text{c}}]}{P_{\text{s}}} + \frac{1}{n\lambda P_{\text{s}}} \tag{3.26}$$

is minimized in the denominator of Eq. (3.23). If we substitute P_s and T_{c} using Eq. (3.17) and (3.10) to Eq. (3.26), Eq. (3.26) becomes

$$D(\lambda) = \frac{e^{2n\lambda T_{\text{RTS}}}}{n\lambda}. \tag{3.27}$$

By setting the derivative of Eq. (3.27)

$$D'(\lambda) = \frac{e^{2n\lambda T_{\text{RTS}}}(2n\lambda T_{\text{RTS}} - 1)}{n\lambda^2} = 0, \tag{3.28}$$

we find that the maximum throughput is achieved when $n\lambda T_{\text{RTS}} = 0.5$. From (3.8) we have

$$\lambda = \frac{2}{(W+1) * T_{\text{slot}}}. \tag{3.29}$$

Therefore, we can get the optimal contention window

$$W = \frac{4nT_{\text{RTS}}}{T_{\text{slot}}} - 1 = 5.887n - 1, \tag{3.30}$$

which is only related to the number of active neighbors n.

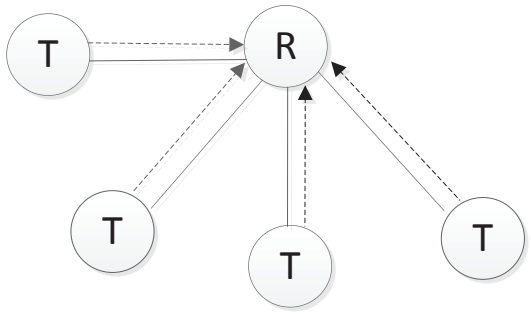


Figure 3.10: A simple scenario in 60 GHz network: four transmitters are located 12 meters from the receiver and start sending Constant Bit Rate (CBR) traffic to the receiver with same rate using UDP protocol at time 0 second.

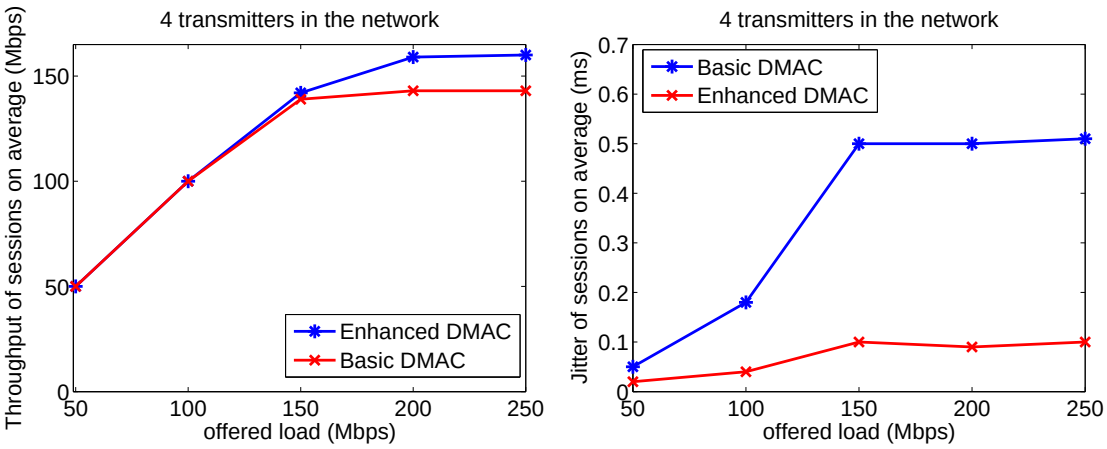
3.6 Performance evaluation for EDMAC

In this section, we validate proposed EDMAC in previous section using ns-2 simulator for more scenarios including multi-hop scenarios. Simulation parameters are shown in Table 3.1. The size of the packet queue in the MAC layer at each node is 50 packets. We measure the performance, namely throughput, delay and jitter, of each flow as defined in Section 3.3.

3.6.1 Single hop scenario

We first study a scenario shown in Fig. 3.10 with four transmitters, each located 12 meters from the receiver, sending Constant Bit Rate (CBR) traffic to the receiver using UDP starting at time 0. The throughput and jitter of each flow with increasing traffic rate are shown in Fig. 3.11. We can see from Fig. 3.11 that, before the network gets saturated, both protocols perform well and the jitter is smaller when using the proposed optimal algorithm. The network using DMAC gets saturated first and has the smallest throughput. We see in 3.11(b) that the jitter is much higher because of the unfairness issue.

Table 3.3 shows the performance of the saturated network with various numbers of transmitters in the network. Using EDMAC, the jitter decreases 75% comparing with basic DMAC by solving the unfairness issue as discussed in Section 3.3.



(a) Throughput. (b) Jitter.

Figure 3.11: Performance of each flow with traffic rate increase when four transmitters starts sending Constant Bit Rate (CBR) traffic to the receiver with same data rate using UDP protocol at time 0 second. The blue stars and red crosses mark the performance of the flow using EDMAC and basic DMAC respectfully.

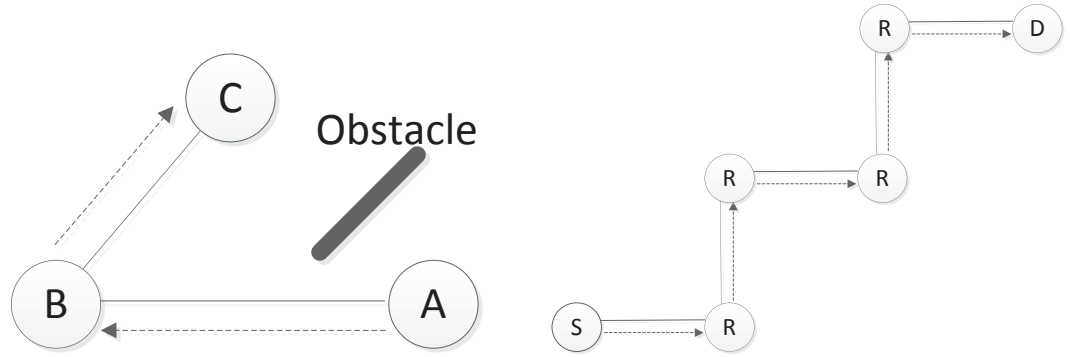
Table 3.3: Performance of n transmitters in the network with various schemes

n	Scheme	Throughput(Mbs)	Jitter(ms)
2	EDMAC	644	0.09
	DMAC	564	0.21
9	EDMAC	583	0.3
	DMAC	566	1.3
17	EDMAC	568	0.55
	DMAC	569	1.9

3.6.2 Multi-hop scenarios

In a 60 GHz network, line-of-sight links can be easily blocked by obstacles such as the human body. Consider the network shown in Fig. 3.12(a), if the link between the source node A and the destination node C is blocked, node A has to find at least one relay node, for example node B, to forward its packets to node C . We study a simple scenario that node A starts sending Constant Bit Rate (CBR) traffic to node C via node B using UDP protocol. The performance of the flow with increasing offered load is shown in Fig. 3.13.

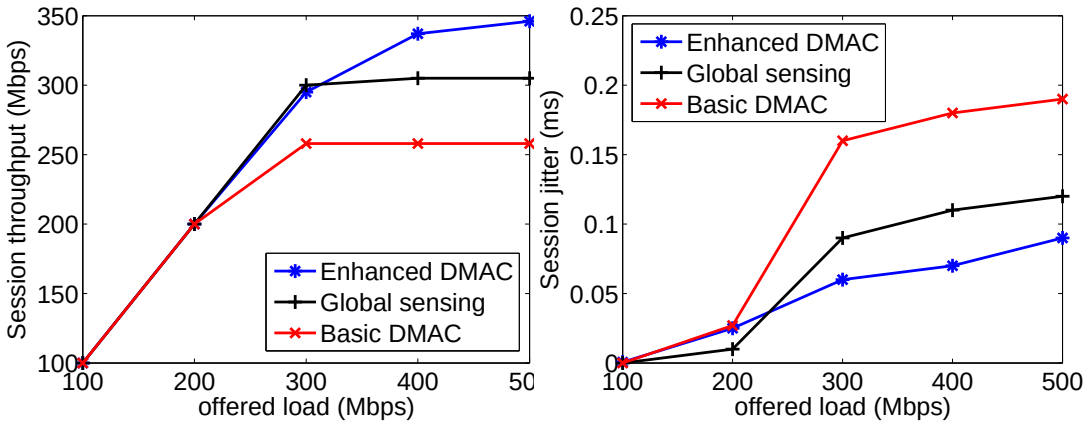
To better study the problem, we compare against an impractical "global sensing" protocol. The "global sensing" protocol is the same as DMAC, except that nodes can sense other nodes'



(a) Two hops: node A sends packets to node C via node B.

(b) h hops: node S sends packets to node D via nodes R.

Figure 3.12: Ad hoc scenarios



(a) Throughput.

(b) Jitter.

Figure 3.13: Performance of the flow with increasing offered load in the scenario that node A starts sending CBR traffic to node C via node B using UDP.

transmissions by using omni-directional antenna with unrealistically large transmission powers. We see that the throughput of network using DMAC is lower than using global sensing protocol due to low channel utilization caused by directional deafness. As shown in Fig. 3.14, when node B is forwarding a packet to node C, node A cannot detect it and sends a new packet to node B. The packet will be discarded by node B since it is pointing at node C. Therefore, node A has to double its contention window and choose a new backoff timer. Since node B will hog the channel for a while, the backoff timer of node A is likely to expire before node B releases the channel. Therefore, after the failed retransmission, node A has to double its contention window size again and chooses a larger backoff timer. After releasing the channel, node B will be idle for a long time since node A is still in a large backoff period. This long idle period will

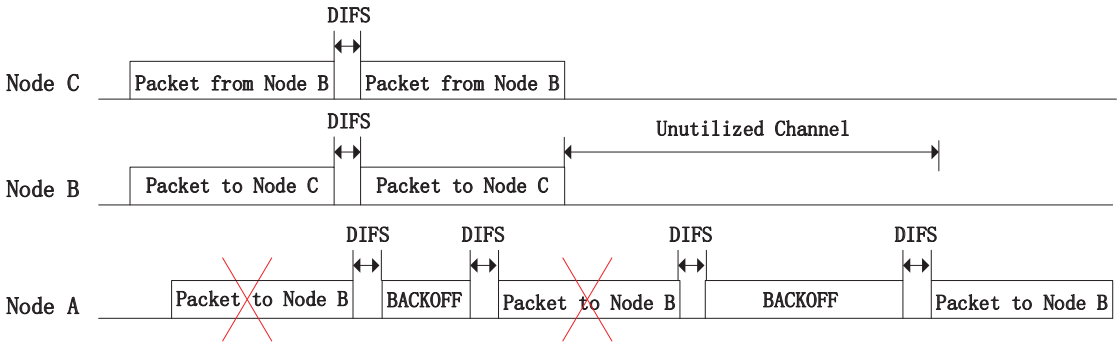


Figure 3.14: Low channel utilization of CSMA/CA with mandatory backoff scheme.

significantly decrease the capacity of the network. On the other hand, with the help of global carrier sensing, node A will pause its backoff timer and avoid sending its next packet while node B is transmitting. Thus, global carrier sensing reduces the number of retransmission, decreases the average backoff time and increases the capacity of the network. The network performs best using the EDMAC since nodes can choose an appropriate contention window that even outperforms the impractical global sensing protocol. It is also worth noting that, ToneMAC which is best performing DMAC protocol in 2.4/5 GHz [24] performs below the global sensing protocol due to the control overhead.

Sometimes due to blockages, the source node cannot reach the destination node via one relay node. Therefore, we also study a h hops scenario that node S sends packets to node D via relay nodes R. To avoid interferences between adjacent hops, we employ a topology that no three nodes are located in a straight line. In Figure 3.15, we plot the throughput of the UDP flow as a function of h and see that the EDMAC always outperforms basic DMAC by at least 25%. Note that the throughput decreases slowly when h increases which is similar with UDP flow over multiple hops using omni-directional antenna as studied in [31]. Using omni-directional and CSMA/CA protocol, the throughput of the flow over many hops in chain topology is only 14% of the throughput over one hop. However, using directional antenna and EDMAC protocol, the throughput of the flow over many hops in chain topology can achieve 25% of the throughput over one hop. This demonstrates that directional antenna can achieve higher spatial reuse that omni-directional antenna network.

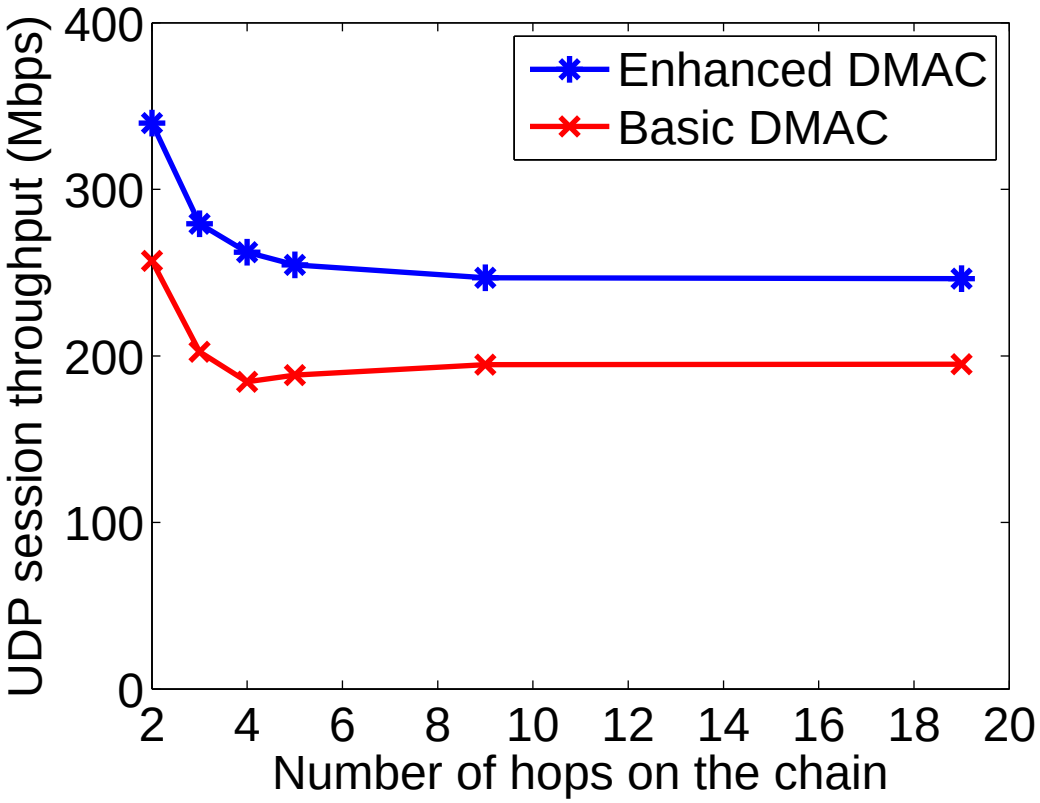


Figure 3.15: The throughput of the UDP flow while number of hops increases.

3.7 Conclusion

This chapter has proposed an EDMAC to resolve the unfairness and low channel utilization issues of deafness in directional MAC protocols for 60 GHz networks. In the EDMAC, packets destined to the same receiver from different sources use a common optimized contention window size. The experimental results show that the EDMAC outperforms basic DMAC especially in multi hop scenarios. Moreover, our EDMAC can be used not only in 60 GHz network, but also for 2.4/5 GHz.

Chapter 4

Performance Evaluation of EDMAC for TCP Traffic

4.1 Introduction

Many measurements indicate TCP now accounts for 85 to 95 percent of the wide-area Internet traffic [32]. Different from UDP traffic, TCP traffic has flow control and congestion control. It is not clear if the flow control and congestion control can work well in 60 GHz networks with EDMAC and DMAC protocol. Authors in [33] studied the performance of TCP flow with omni-directional antenna in 2.4/5 GHz networks, and pointed that unlike wired networks where buffer overflow dominates packet losses, most packet drops experienced by TCP are due to link layer contention, incurred by hidden terminals. However, unlike omni-directional antenna networks, deafness becomes the main reason for a packet drop in 60 GHz networks with directional antennas. A packet loss should not be the first sign of a network overload. Therefore, we should increase the MAC layer retransmission times to make sure a packet will not be dropped due to deafness problem. Moreover, in 60 GHz networks, when a transmitter and a receiver are pointing at each other and transmitting a packet, interference from the side lobe cannot corrupt the packet due to the high main lobe gain. Therefore, 60 GHz networks perform similar as wired networks but different than wireless networks using omni-directional antenna. In this chapter, we evaluate the performance of TCP flows in 60 GHz networks for various scenarios.

4.2 Performance of a TCP flow over multi-hop networks

In this section, we study the performance of a TCP flow whose source and destination are placed at each end of a chain with h hops. To avoid interferences between adjacent hops, we design a topology that no three nodes are located in a line as shown in Figure 3.12(b), i.e., node

i is located at coordinate $(\lceil (i/2.0) * 12 \rceil, \lfloor (i/2.0) * 12 \rfloor)$. We set the receive window (RWND) and the slow start threshold to 100 packets. This indicates that the congestion window will be doubled every round trip time (RTT) until it reaches 100. Each data packet of the TCP flow has 8000 bytes. The buffer of each node is 500, such that no buffer over flow happens.

In Figure 4.1, we plot the throughput of the TCP flow as a function of h. We use $T_e(h)$ and $T_d(h)$ to denoted the throughput using EDMAC and DMAC respectively. We can see that both $T_e(h)$ and $T_d(h)$ are monotonic decreasing when h increases. However, $T_e(h)$ is always greater than $T_d(h)$, indicating that EDMAC preforms better than DMAC which uses the exponential backoff scheme. Since a node cannot transmit and receive a packet at the same time, the maximum throughput of the TCP flow over two hops is only half of the throughput over one hop. However, the simulation results show that $T_e(19)$ is only 36% of $T_e(1)$ and $T_d(19)$ is only 27% of $T_d(1)$. This is because contention based MAC protocols are not optimal scheduling algorithms that can maximize the spatial reuse. However, EDMAC can achieve higher throughput than DMAC in this multi-hop chain topology. It is interesting to see that both $T_e(h)$ and $T_d(h)$ become flat when h becomes bigger which is similar as the UDP traffic in previous chapter. we will explain the reason as follows.

We know from IETF RFC 6349 standard that the throughput of a TCP flow is

$$T = \text{RWND} \times 8/\text{RTT}, \tag{4.1}$$

where RTT is the average Round Trip Time (RTT) of the TCP flow. In our simulation scenario, since RWND is fixed to 100 packets we focus on analyzing why the RTT of the TCP flow does not increase much when the chain becomes longer. We measure the packet transmission time of a packet at MAC layer, which is the time difference between the time that the packet becomes the head of the interface queue of the sender and the time that the sender receives the ACK for the packet from the receiver. From the set of packet transmission times over link (i, j), we can estimated the average packet transmission time $t_{i,j}$ of link (i, j). We also record the length of MAC layer interface queue of each node every 1ms and obtain the average number of packets in the interface queue of node i denoted by l_i. When a packet arrives at a node, it has to wait until all packets already in the queue are transmitted. Thus, we can calculate RTT by

$$\text{RTT} = \sum_{i=0}^{h-1} (t_{i,i+1} l_i + t_{i+1,i} l_{i+1}) \tag{4.2}$$

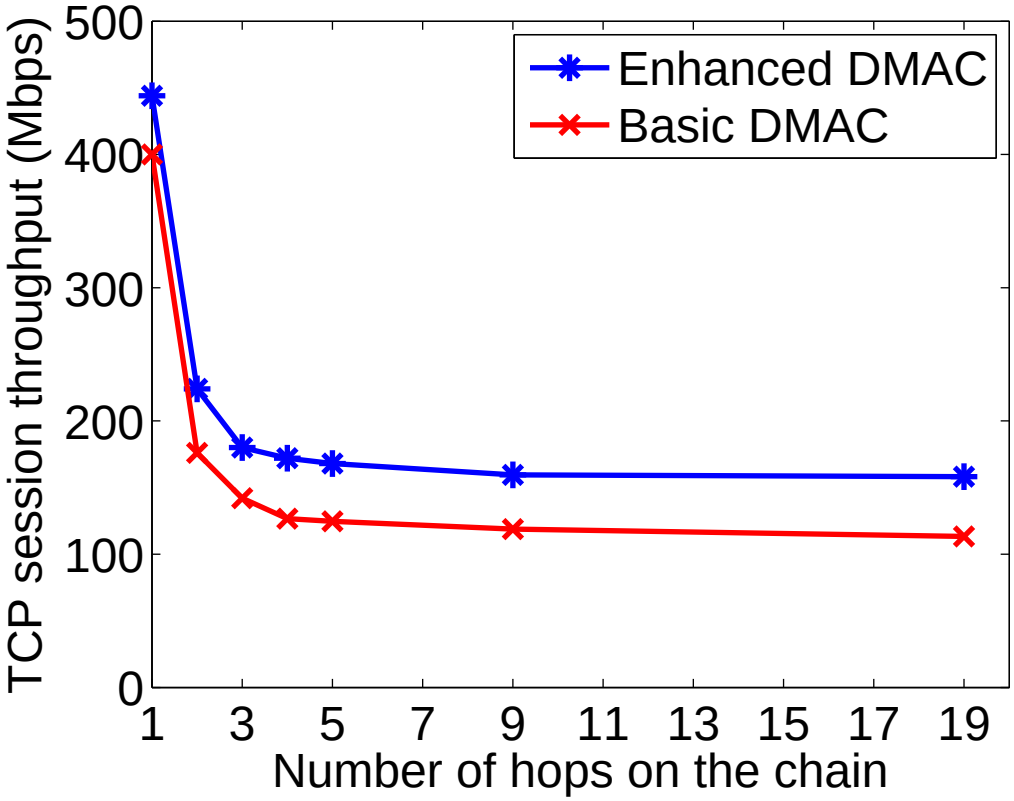
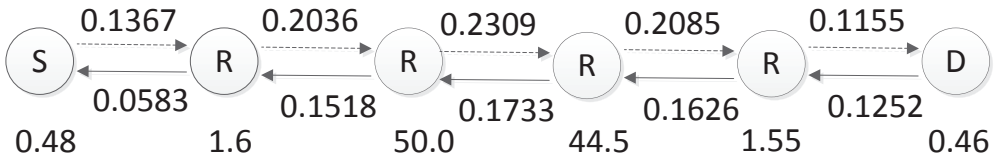
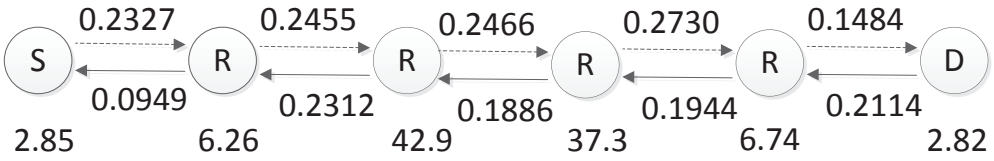


Figure 4.1: The throughput of the TCP flow while number of hops increases.

The average packet transmission time of each link and the average queue length of each node are shown in Fig. 4.2 and 4.3 for 6 and 10 nodes on a chain using EDMAC and DMAC protocols.

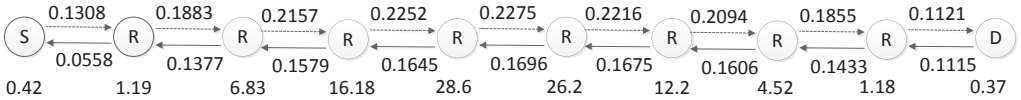


(a) Using EDMAC protocol.

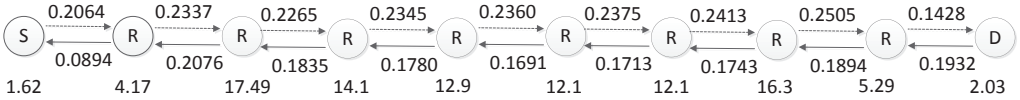


(b) Using DMAC protocol.

Figure 4.2: Average packet transmission time of each link and the average queue length of each node for 6 nodes on a chain.



(a) Using EDMAC protocol.



(b) Using DMAC protocol.

Figure 4.3: Average packet transmission time of each link and the average queue length of each node for 10 nodes on a chain.

From Fig. 4.2 and 4.3, we first observe that $t_{i,i+1}$ and $t_{i+1,i}$ on the same chain do not vary much with different i when $0 < i < h - 1$, especially when the chain is long. These hold because nodes in the middle of the chain are identical in terms of topology and always have queued packet to transmit. In the TCP flow control mechanism, the source only sends a new data packet out when it receives a TCP ack packet from the destination. Therefore, the sum of

number of TCP data and ack packets queued in the network is equal to RWND. Thus,

$$\psi = \sum_{i=1}^{h+1} l_i = 100 = \text{RWND} \tag{4.3}$$

Note that the value of ψ obtained from simulation results is a little bit smaller 100. This is because in $ns-2$ simulator, there is a 25us processing delay between the time when the MAC layer sends a packet up to routing layer and the time when the routing layer receives the packet. Moreover, there is also a 25us processing delay between the time when the routing layer sends a packet down to the MAC layer and the time when the MAC interface queue receives the packet. Therefore, the gap T_{gap} between the time when a node receives a packet from MAC layer and the time when the node inserts the packet into the MAC interface queue is 50μs. Thus, if we sample the length of the interface queue during the gap, the queue length is one packet smaller than the actual number of packets buffered. The probability ρ that a packet is not counted in the interface queen of all nodes is $\frac{2T_{gap}h}{RTT}$. Therefore, the number of packets monitored in the MAC interface queue is

$$\psi' = (1 - \frac{2T_{gap}h}{RTT}) \times \text{RWND}. \tag{4.4}$$

Fig. 4.4 shows that value of ψ obtained from simulation results is very close to ψ' got from Eq. (4.4,) and thus verifies our analysis.

From the above observations, we can induce that further increasing the number of nodes in the chain does not increase the average transmission time between nodes much. It only changes the distribution of l_i at each node, and the sum of l_i remains the same as RWND is fixed. Therefore, the RTT of a TCP flow does not increase much when the chain becomes longer and so does the throughput.

In [33], the authors studied the performance of TCP flows over multi-hops networks using omni-directional antenna. In their study, they mentioned that there exists an optimal RWND equal to $h/4$ that achieves maximum throughput for a flow in chain topology. Decreasing or increasing the RWND will decrease the throughput. The reason is that when RWND is smaller, the network capacity is not fully utilized, but when RWND is bigger, collisions will happen and a packet drop will trigger slow start process which decreases the congestion window size. However, this is not true in 60 GHz as shown in Fig. 4.5. This is because once the

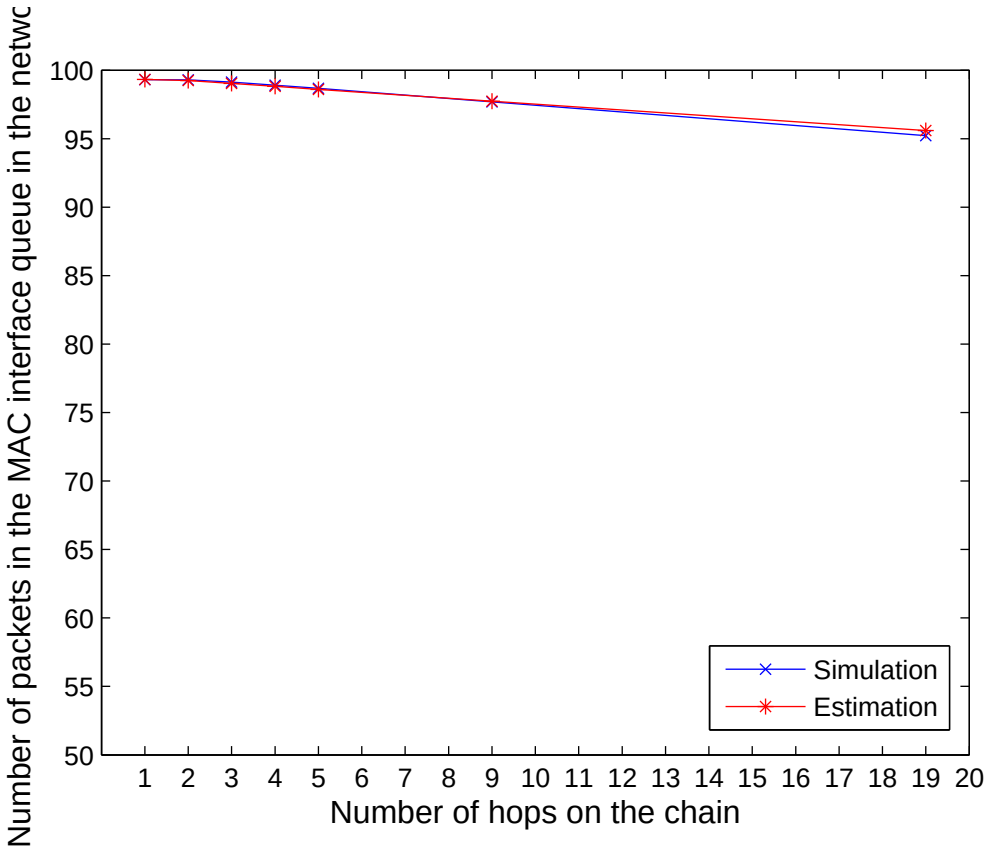


Figure 4.4: Number of packets measured in the MAC interface queue of nodes in the network.

transmitter and receiver point at each other, no collision will happen due to high main lobe gain of directional antennas. Therefore, increasing the RWND will not decrease the throughput as long as the buffer at each node does not overflow.

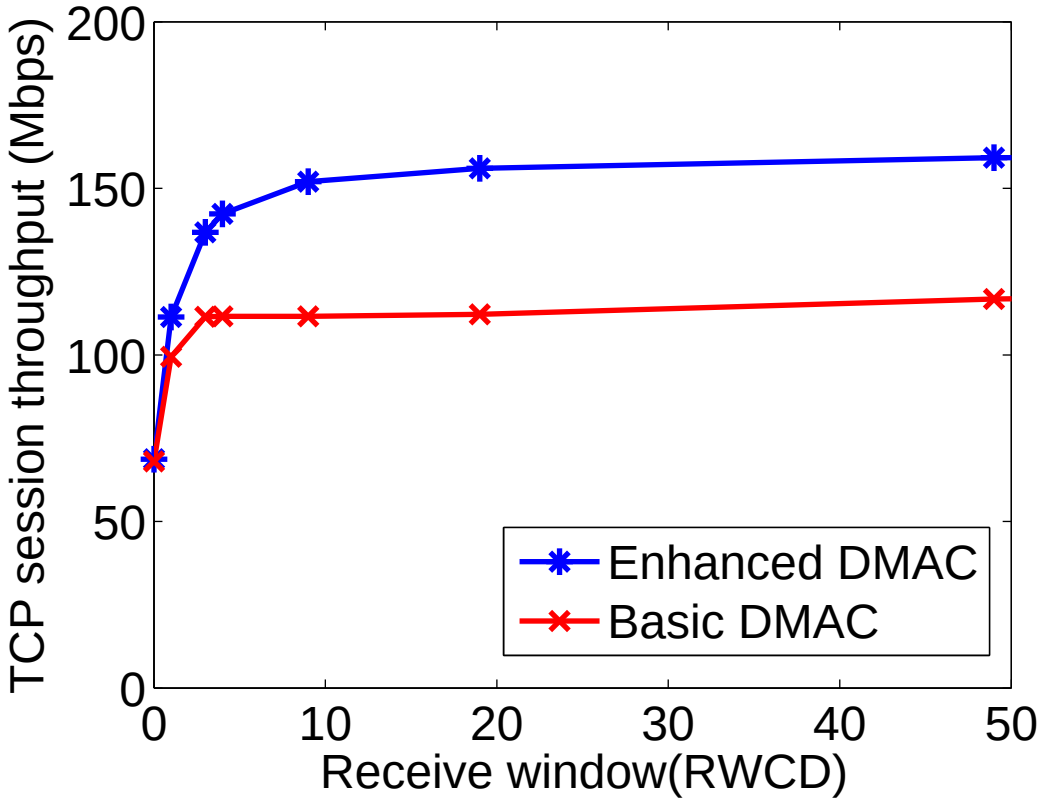


Figure 4.5: The throughput of a TCP flow as a function of received window size.

4.3 Performance of multiple TCP flows in networks

In this section, we study the performance of multiple co-existing TCP flows in many topologies. First, we study the Scenario 1 as a baseline that both source nodes S1 and S2 have a TCP flow designated to node D as shown in Fig. 4.6(a). We then study Scenario 2 that the link between nodes S1 and D in Scenario 1 is blocked, node S1 has to choose a relay node R to forward packets to node D as shown in Fig. 4.6(b). We also study Scenario 3 that the link between nodes S2 and D in Scenario 2 is blocked and node S2 chooses the same relay node R to relay its packets. Table 4.1 summarizes the throughput of TCP flows 1 and 2 respectively. We find

Table 4.1: Throughput of multiple TCP flows

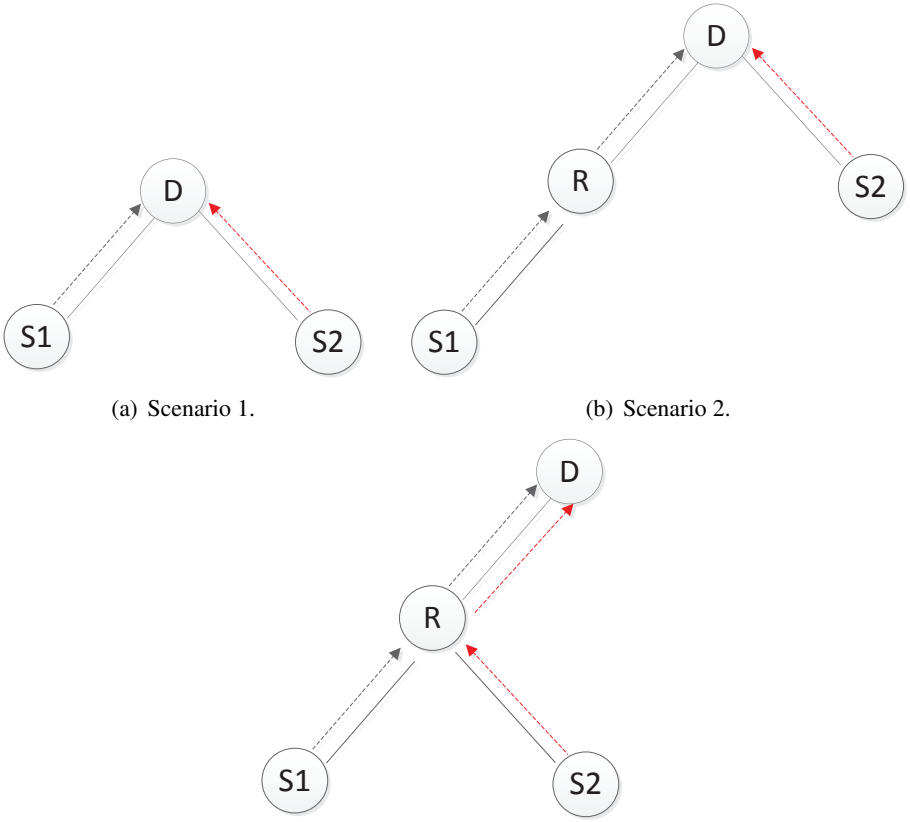
		EDMAC Throughput(Mbps)		DMAC Throughput(Mbps)	
Case	Queue type	S1	S2	S1	S2
1	FIFO	231	231	202	202
2	FIFO	170	227	87	280
	DRR	180	204	116	240
3	FIFO	112	112	89	59

Table 4.2: Throughput of multiple TCP flows

		EDMAC Throughput(Mbps)		DMAC Throughput(Mbps)	
Case	Queue type	S1,S2,S3	S4	S1,S2,S3	S4
4	FIFO	72,72,72	100	60,60,60	231
	DRR	59,59,59	176	51,51,51	156
		S1,S2,S3,S4	S5,S6	S1,S2,S3,S4	S5,S6
5	FIFO	47,47,47,47	47,47	38,38,38,38	40,40
	DRR	37,37,37,37	75,75	34,34,34,34	59,59

that for Scenario 1 and Scenario 3, the throughput of two flows are almost the same since they are symmetric identical in topology. However, in Scenario 2, flow 2 has higher throughput than flow 1 since flow 2 has fewer hops than flow 1. If nodes use a single First-come-first-serve (FIFO) queue, the throughput difference between flow 1 and flow 2 is bigger than using Deficit Round Robin (DRR) [34] queue. This is because DRR can increase the fairness between flows. Moreover, the throughput difference is smaller using EDMAC than using DMAC, which demonstrates again that EDMAC performs better than DMAC in terms of fairness.

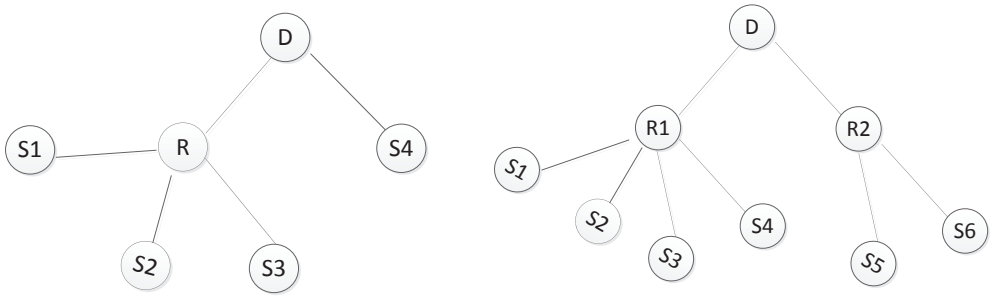
Fig. 4.7 shows two complicated scenarios with more TCP flows. The throughput of each flow is shown in Table 4.2. We notice that using DRR queue, the flow rate can be predicted based on the topology. For example, as shown in Fig. 4.7(a), flows S1, S2 and S3 share the same relay nodes, therefore, they have the same throughput. At the same time, node R and node S4 share bandwidth of node D, therefore, the sum of throughput of S1−S3 is closed to the throughput of S4. Same result can be seen for Scenario 5 shown in Fig. 4.7(b) that the sum of throughput of S1-S4 is closed to the sum of throughput of S5 and S6.



(a) Scenario 1. (b) Scenario 2.

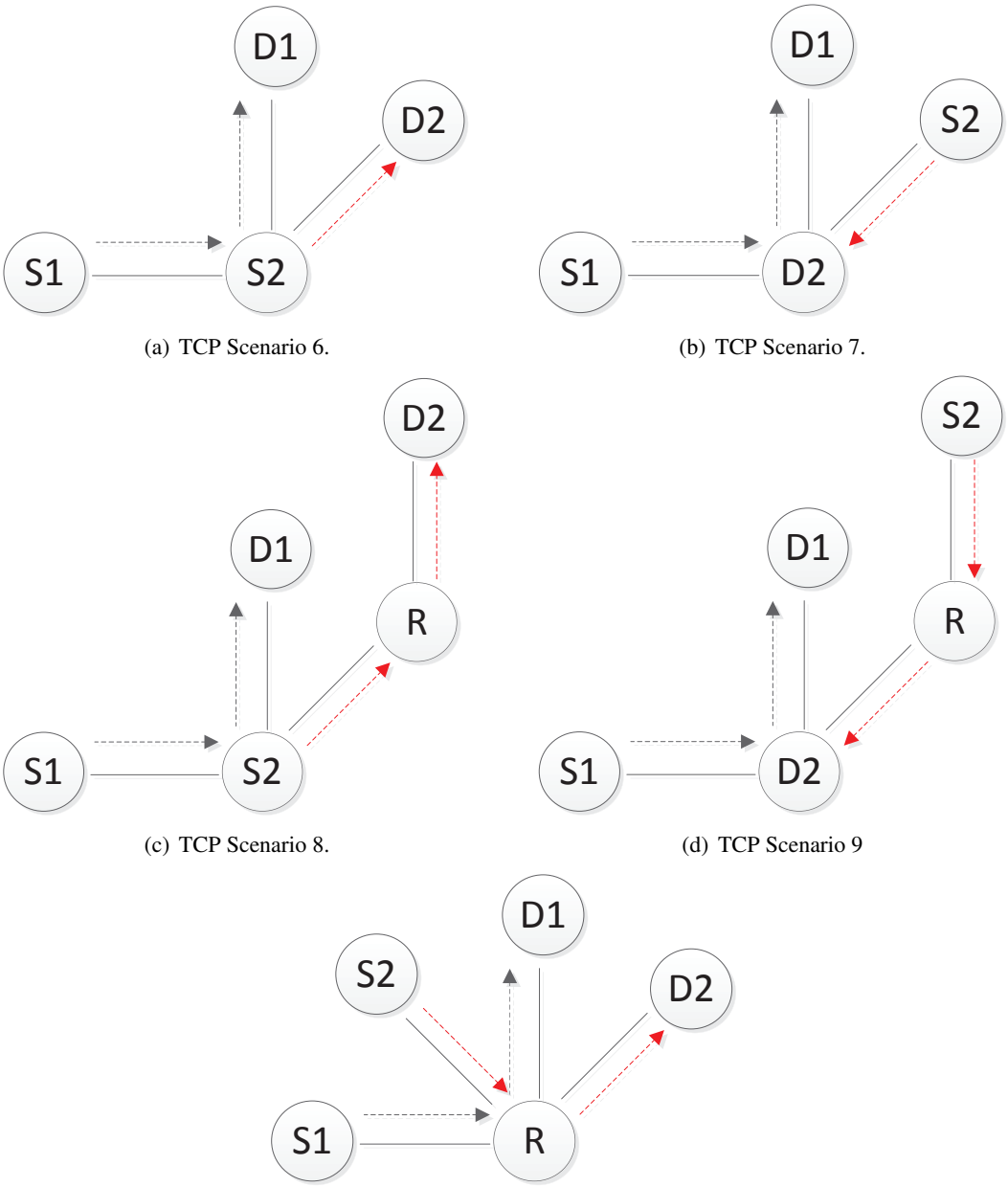
(c) Scenario 3.

Figure 4.6: Topologies for multiple TCP flows in the the network.



(a) TCP Scenario 4. (b) TCP Scenario 5.

Figure 4.7: Topologies for multiple TCP flows in the the network.



(a) TCP Scenario 6.

(b) TCP Scenario 7.

(c) TCP Scenario 8.

(d) TCP Scenario 9

(e) TCP Scenario 10

Figure 4.8: Topologies for multiple TCP flows in the the network.

Table 4.3: Throughput of multiple TCP flows

		EDMAC Throughput(Mbps)		DMAC Throughput(Mbps)	
Case	Queue type	S1	S2	S1	S2
6	DRR	152	152	81	230
	RR	154	151	72	249
7	DRR	135	184	67	262
	RR	153	153	77	239
8	DRR	138	138	120	98
	RR	139	139	116	101
9	DRR	118	147	84	129
	RR	137	137	116	100
10	DRR	97	97	101	83
	RR	99	99	100	85

In simulation scenarios $1-5$, there is only one destination node in the network. In Fig. 6.14, we study some basic scenarios that two flows cross each other and are destined to different nodes. The throughput of each scenario is shown in Table 4.3. We can see that two flows have equal throughput if nodes use DRR queue and EDMAC except Scenario 7 and Scenario 9. This is because links share the bandwidth of a node equally. In Scenario 7 and Scenario 9, since TCP ack packets are much smaller than TCP data packets, flow 2 has higher throughput than flow 1 if nodes use DRR queue. We can see that two flows have the same throughput if nodes use basic Round Robin (RR) queue. Based on these observations, we can estimate the bandwidth of a link, $e(i,j)$, using number of active links crossing nodes i and j. If we assume the total bandwidth associated with a node is 1 and n_i is number of active links associated with node i, the maximum bandwidth of each link associated with node i is $1/n_i$. Thus, we have

$$e(i,j) = \min(\frac{1}{n_i}, \frac{1}{n_j}). \tag{4.5}$$

For example, in Scenario 6 the bandwidth of each link associated with node S2 is $1/3$ and in Scenario 10 the bandwidth of each link associated with node R is $1/4$. The throughput of the flow along path p, $T(p)$, is determined by the link on the path which has the minimum bandwidth. If we denote a h hops path $p = (v_1, v_2, ..., v_{h+1})$, where v_i is the ith node on the

path, we have

$$T(p) = \min_{i=1,2,..,h} e(v_i, v_{i+1})$$

4.4 Conclusion

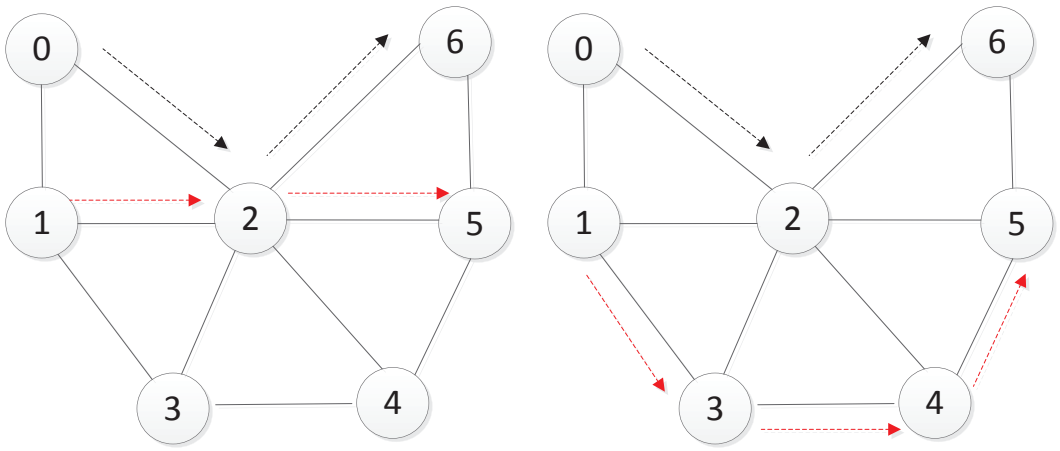
In this chapter, we study the performance of TCP traffics in multi-hop networks using EDMAC and DMAC with different queue model such as First-In-First-Out (FIFO), Round Robin (RR) and Deficit Round Robin (DRR). Using EDMAC and Round Robin based queue can improve the fairness between flows in the network and help to estimate the throughput of a flow given its route and network topology. These observations can help us to design a better routing algorithm for 60 GHz network as discussed in the next chapter.

Chapter 5

Single Path Routing in 60 GHz Networks

5.1 Introduction

In the previous chapter, we studied the performance of EDMAC and DMAC for a random generated topology using hop count as the routing metric. However, the shortest path routing protocol does not always fully utilize the high spatial reuse properties of directional antennas. In this chapter, we first study a simple example to illustrate the limitation of shortest path routing. We then study some atomic scenarios and find that flows passing the same node equally share the bandwidth of the node using EDMAC and Round Robin (RR) queue. However, this does not hold using DMAC. This also demonstrates that EDMAC improves the fairness between flows relative to DMAC. Based on these atomic scenarios, we can approximately predict the throughput of a flow by its path. Thus, we can choose the fattest path, i.e, the path for which the link on the path with minimum bandwidth is maximized, as a route from a source to a destination. We will also refer to this fattest path as the max-min bandwidth path. However, there may be more than one fattest path to a destination in the network. As discussed in previous chapter, the throughput decreases when hop count increases if the total hop count on the path is below 5. Therefore, we propose two heuristic routing algorithms that combine fattest path and minimum hop together. Simulation results show that the proposed algorithms increase the capacity of the network by 10% on average. However, the proposed algorithms are not guaranteed to perform better than shortest path routing for all topologies. This is because our prediction model is too simple that it assumes links do not interfere with each other if they are node-disjoint. We then add a simple interference model and demonstrate by simulations that combining fattest path and minimum hop together can improve the throughput of the network.



(a) Minimum hop count routing (b) Node disjoint path routing

Figure 5.1: Minimum hop count routing and node disjoint path routing

Table 5.1: Throughput (in Mbps) using minimum hop and node disjoint path routing

	directional antenna			omni-directional antenna		
Routing algorithm	Aggregated	Flow 1	Flow 2	Aggregated	Flow 1	Flow 2
Minimum hop count	226	113	113	202	101	101
Node disjoint path	402	218	184	164	82	82

5.2 Limitations of minimum hop count routing

A 60 GHz network can achieve high spatial reuse using directional antennas. However, minimum hop routing protocols may not always fully utilize the high spatial reuse property of directional antennas. For example, in the topology shown in Fig. 5.1, there are two flows in the network. Flow 1 is from node 0 to node 6 and flow 2 is from node 1 to node 5. Fig. 5.1(a) and Fig. 5.1(b) show the routes of the flows using minimum hop count routing and node-disjoint path routing respectively. Table 5.1 shows the throughput of the two routing protocols using directional antenna in 60 GHz using EDMAC and omni-directional antenna (we set the omni-directional antenna gain equal to the main-lobe gain of directional antennas). In the omni-directional antenna network, nodes $0-3$ cannot transmit simultaneously, and thus it is impossible to exploit spatial reuse as shown in Table 5.1. In this case, the minimum hop count routing protocol will outperform the node-disjoint routes. However, using directional antennas, node-disjoint links usually do not interfere with each other. Therefore, the node disjoint path routing performs better than minimum hop count routing since two flows do not interfere with each other.

5.3 The fattest path routing protocol

As described in previous section, the node-disjoint path routing generally outperforms shortest path routing. However, in many topologies, the node-disjoint paths between k source-destination pairs may not exist. Therefore, when a new flow joins the network, it would choose a path that can achieve the highest throughput. In Chapter 4, we found that we can approximately estimate the bandwidth of a link, $e(i,j)$, using number of active links crossing nodes i and j. If we assume n_i is the number of active links associated with node i, the maximum bandwidth of each link associated with node i is $1/n_i$. Thus, we have

$$e(i,j) = \min(\frac{1}{n_i}, \frac{1}{n_j}) \tag{5.1}$$

Algorithm: Find the maximum throughput path P for a flow from s to d
Input: graph $G = (V, E)$;$n_i, i \in V$
Output: path P
Method:
1. **FOR** each node $i \in V$
2. **IF** node i is the source or destination node
3. $n_i = n_i + 1$
4. **ELSE**
5. $n_i = n_i + 2$
6. **FOR** each edge $e(i,j)$ in E
7. $e(i,j) = \min(\frac{1}{n_i}, \frac{1}{n_j})$
8. $P = Dijkstra\text{-}F(V, E)$
9. **FOR** each node i in V
10. $n_i = n_i - 2$

Figure 5.2: Find the path with maximum throughput.

The throughput of the flow along path p, $T(p)$, is determined by the link on the path which has the minimum bandwidth. If we denote a h hops path $p = (v_1, v_2, ..., v_{h+1})$, where v_i is the ith node on the path, we have

$$T(p) = \min_{i=1,2,..,h} e(v_i, v_{i+1})$$

We develop the algorithm shown in Figure 5.2 to find the fattest path, i.e. the path has maximum path capacity. The key component of the algorithm is Dijkstra-F function shown in Figure 5.3. In Dijkstra-F function, we modified the Dijkstra algorithm to find the fattest path from the source to the destination.

Algorithm: $Dijkstra$-F
Input: graph $G = (V, E)$
Output: path P
Method:

1. **FOR** each $v \in V - s$, let $v.fat = 0$
2. $s.fat = \inf$
3. Insert all node in a priority queue Q keyed by the fat field
4. **WHILE** Q is not empty
5. find and remove node u in Q whose field $u.fat$ is largest among queue elements.
6. **FOR** all nodes v such that $(u, v) \in E$
7. **IF** $v.fat < \min(u.fat, e(u, v))$
8. $v.fat = \min(u.fat, e(u, v))$
9. update Q to reflect changed value of $v.fat$
10. $v.pred = u$

Figure 5.3: Find the fattest path based on Dijkstra algorithm.

Table 5.2: Throughput using different flow paths

	Throughput (Mbps) using EDMAC		
case	Aggregate	Flow 1	Flow 2
case 1	230	115	115
case 2	446	233	233
case 3	397	213	184
case 4	250	126	124

5.4 Performance evaluation of fattest path routing in a simple scenario

In this section, we study a simple scenario, as shown in Fig. 5.4, to evaluate the performance of min-hop routing and fattest path routing protocols. Flow 1, which is from node 0 to node 7, joins the network first. Flow 2, which is from node 1 to node 6, joints the network after Flow 1. Using the min-hop routing protocol, Flow 1 makes an equiprobable choice between paths $0-2-7$ and $0-3-7$. Flow 2 always choose the path $1-2-6$. Using the fattest path routing protocol, flows may choose routes as shown in cases $2-4$, but will never choose route as shown in case 1. The aggregated throughput of the network for these four cases is shown in Table 5.2. We can see that case 2 and case 3 have much higher aggregated throughput than case 1 and case 4. This is because these two flows choose node-disjointed paths. The aggregated throughput of case 2 is higher than case 3 since the path of flow 2 has fewer hops. The aggregated throughput

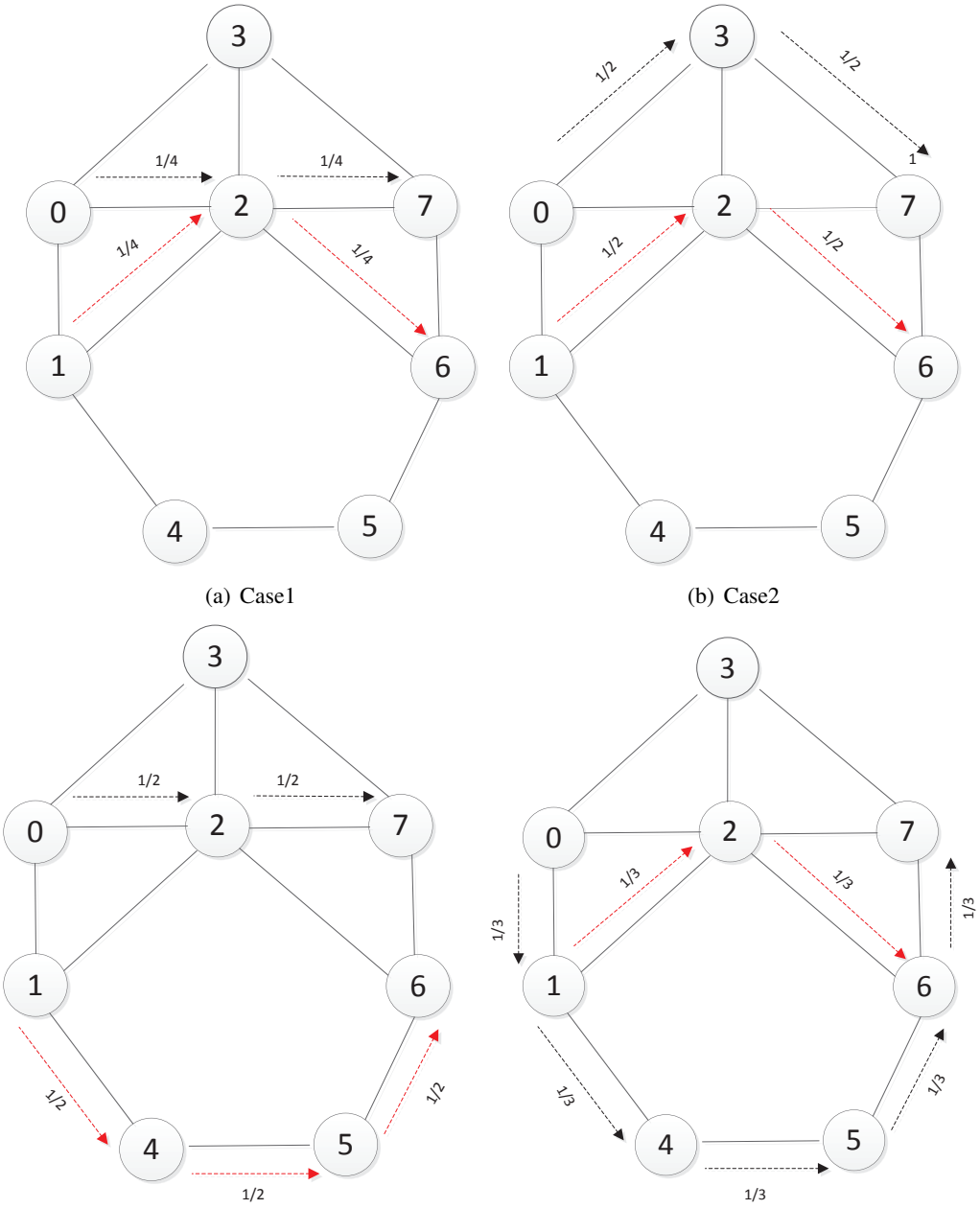


Figure 5.4: A simple scenario for min-hop and fattest path routing

of case 4 is a little bit higher than case 1, since the bottleneck nodes of case 4, nodes 1 and 6, have fewer active links than the bottleneck node 2 of case 1. From this simple scenario, we can know that both min-hop routing and fattest path routing can find the best routes, as in case 2. However, neither of these two routing algorithms are guaranteed to choose the best routes. In particular, when flow 1 joints the network, any path from node 0 to node 7 is the fattest path. Therefore, using fattest path alone, flow 1 may choose some bad paths such as the path shown in case 4. This motivates us to combine the min-hop and fattest path algorithms as discussed in next section.

5.5 Hybrid routing algorithms

In this section, we propose two heuristic routing algorithms that combine fattest path and min-hop routing together. The first algorithm is called fattest path first and hop count second (FP-HOP) algorithm. In the FP-HOP algorithm, when a flow joints the network, we run the fattest path algorithm shown in Fig. 5.2 for K times and then get K fattest paths. Note that some of these K fattest paths may be identical. Then we choose the path which has minimum hop count among these K paths. The second algorithm is called hop count first, fattest path second (HOP-FP) algorithm. In the HOP-FP algorithm, we first use K-shortest path algorithm in [35] to find the first K shortest paths, and then choose the fattest path among them. If there are multiple fattest paths, we choose the one with minimum hop count.

To evaluate the performance of the hybrid routing algorithms, we place 50 nodes (node 0 to node 49) in a 45m$\times 20$m grid, i.e. the coordinate of node i is $(5 \times (\lfloor i/10 \rfloor), 5 \times (i \mod 5))$ as shown in Fig. 5.5. We then displace each node by a distance randomly selected from the interval [0,5] meters as shown in Fig 5.5. We simulate 5 such topologies, each having 5 TCP flows, where the ith flow is from node $i-1$ to node $50-i$, that join the network simultaneously. In order to remove route discovery overhead, we run FP-HOP and HOP-FP offline, and generate routing tables for each node. We then import these routing tables into the ns-2 simulator, and do not change the routes during the simulation.

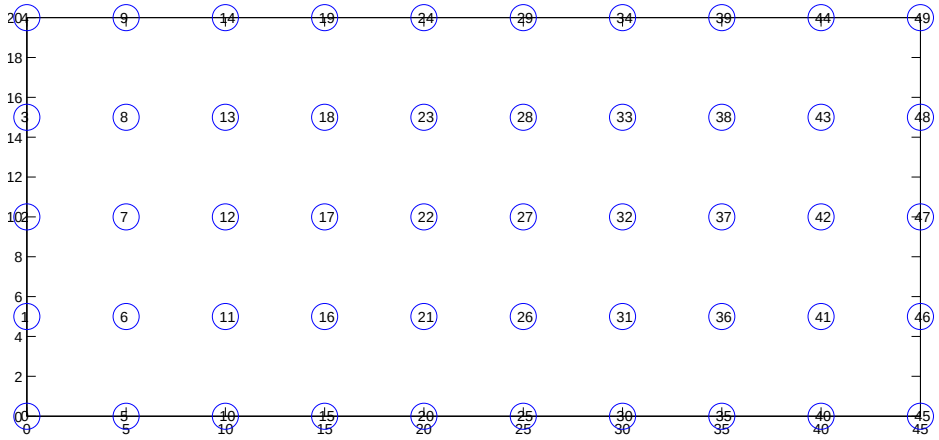


Figure 5.5: 50 nodes placed in a $45\text{m}\times20\text{m}$ grid

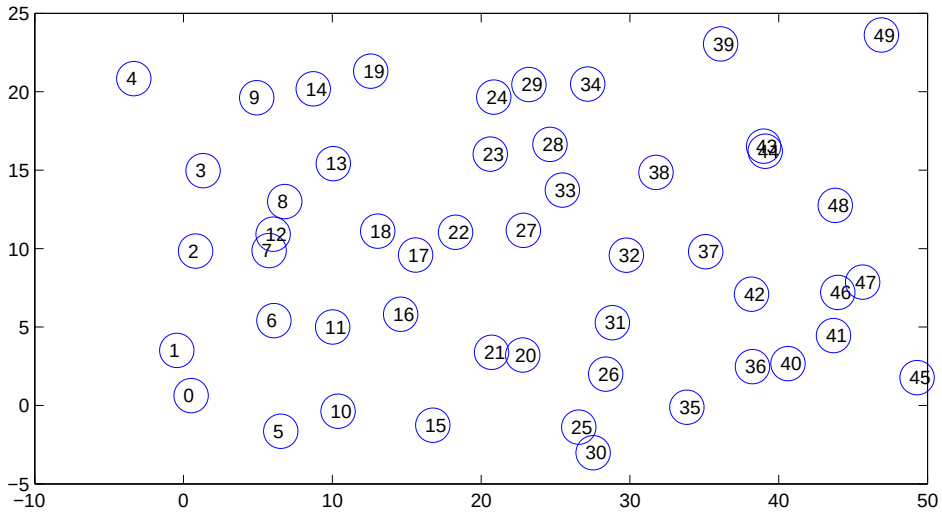


Figure 5.6: Randomly generated simulation topology

Table 5.3: Aggregated throughput using bandwidth estimation model without interference model(Mbps)

Topology	min-hop	HOP-FP	FP-HOP
1	308	397	347
2	278	335	317
3	285	398	310
4	259	416	202
5	308	293	299
Average	288	368	295

Table 5.4: Aggregated throughput using bandwidth estimation model with interference model(Mbps)

Topology	min-hop	HOP-FP	FP-HOP
1	308	353	438
2	278	377	392
3	285	417	406
4	259	384	387
5	308	349	409
Average	288	376	406

Table 5.3 shows the aggregated throughput of the network for 5 randomly generated topologies. We can see that the HOP-FP algorithm increases the capacity of the network by approximately 27% on average. However, the proposed FP-HOP algorithm has a similar performance compared to the shortest path algorithm. This is because our bandwidth prediction model is too simple in that we assume links do not interfere with each other if they are node disjoint. In reality, nodes will interfere with other nodes that are close to it. Therefore, a simple interference model is proposed to estimate the bandwidth of a link $e(i,j)$ between node i and j. If we assume nodes that within distance d will interfere with each other, then node i will have a neighbor node set A_i that contains all nodes that are within distance d. The new link estimation that considers interference from neighbors is

$$e(i,j) = \min\left(\frac{1}{n_i + \sum_{a \in A_i} n_a}, \frac{1}{n_j + \sum_{a \in A_j} n_a}\right) \tag{5.2}$$

Table 5.4 shows the aggregated throughput of the network using HOP-FP and FP-HOP algorithms with new bandwidth estimation model in Equation 5.2, where d is set to 5 meters. To compare with the result using bandwidth estimation model without the interference model, the same 5 topologies are used as as in Table 5.3. After considering interference into the bandwidth estimation, we can see that the FP-HOP algorithm increases the capacity of the

network by approximately 40% on average. By comparison, the HOP-FP algorithm does not improve as much relative to the no interference case. This is because in HOP-FP the algorithm, hop count is the dominant criteria in selecting paths.

5.6 Conclusion

In this chapter, we study the performance of single path routing for network using EDMAC. We find that the shortest path routing protocol does not always fully utilize the high spatial reuse property of directional antennas. Moreover, we find that flows that pass the same node equally share the bandwidth of the node using EDMAC and Round Robin (RR) queue. Therefore, we can approximately predict the throughput of a flow by knowing the path. This enables us to choose the fattest path, i.e, the path for which the link on the path with minimum bandwidth is maximized, as a route from a source to a destination. The fattest path implicitly utilized the high spatial reuse properties of directional antennas to increase the capacity of the network. However, analysis showing that using the fattest path algorithm alone may find a route with more hops which reduce the throughput of the network. We propose two heuristic routing algorithms named HOP-FP and FP-HOP that combine fattest path and minimum hop routing together with and without the consideration of interference and validate them using ns-2 simulator. Simulation results show that without the consideration of interference, both HOP-FP and FP-HOP the HOP-FP algorithms are not guaranteed to perform better than shortest path routing for all topologies. However, the HOP-FP algorithm increases the capacity of the network by 27% on average and the FP-HOP algorithm also outperforms the shortest path on average. With the consideration of interference, both algorithms outperforms the shortest path algorithm on all topologies. The HOP-FP and FP-HOP algorithms increase the capacity of the network by 30% and 40% respectfully. Both HOP-FP and FP-HOP without the consideration of interference can be implemented as online routing algorithms using EDMAC without extra overhead since the number of active links are monitored and used by EDMAC. However, HOP-FP and FP-HOP with the consideration of interference require the MAC layer to have extra function to monitor the interference around a node.

Chapter 6

Multipath Routing in 60 GHz Networks

6.1 Introduction

Recent technology advances are poised to enable low-cost, low-power communications in the 7 GHz of unlicensed spectrum at 60 GHz millimeter wave (mmW) frequencies. However, mmW systems that meet the Gb/s data rate demands of wireless multimedia applications must overcome severe propagation effects, including high path loss and high diffraction loss [4]. Consequently, nodes in the network have to use directional antennas to communicate at high data rates. The narrow main beam width of directional antennas introduces many design challenges for Medium Access Control (MAC) protocols, at the same time, it can provide several benefits such as high gain and high spatial reuse. Many directional MAC protocols, such as DMAC [24] and EDMAC [36], have been proposed to improve the performance of the network at the MAC layer. However, fewer directional routing protocols have been proposed to improve the capacity of the network by utilizing the benefit of high spatial reuse.

In this chapter, we employ node-disjoint multipath routing schemes to utilize the high spatial reuse property of directional antennas, and thus we improve the throughput of a TCP flow. In the node-disjoint multipath routing scheme, packets of a flow between the source and the destination may traverse through different paths that have no nodes in common. The node-disjoint multipath routing scheme has been proposed in Mobile Ad hoc Network (MANET) [37] in omni-directional antenna networks to improve the reliability of a flow. However, the node-disjoint multipath routing scheme cannot improve the throughput of a TCP flow in an omni-directional antenna network, as we demonstrate with a toy example in Section 6.2.

How to discover the node-disjoint paths is a challenging problem using directional antennas. In [38], Lee et. al extended Dynamic Source Routing (DSR) [12] to find two node-disjoint paths for omni-directional antenna networks. However, there are several key differences between

using directional antennas and an omni-directional antenna in a route discovery process. For example, a node that uses a directional antenna does not have the ability to send a message to all its neighbors at once using broadcast. In [39], Choudhury et. al studied DSR for single-path routing in directional antenna networks. However, this scheme cannot find multiple node-disjoint paths. In [40], Li et. al used multipath source routing to address the deafness problem caused by directional antennas. However, the authors did not give detailed descriptions about how to obtain multipath routes nor did they analyze the performance of the discovery process. In [41], Hu et. al considered hop counts and beam overlaps to select two routes from the source to the destination. However, the author assumed nodes are capable of generating two beams and can radiate to two different nodes at the same time.

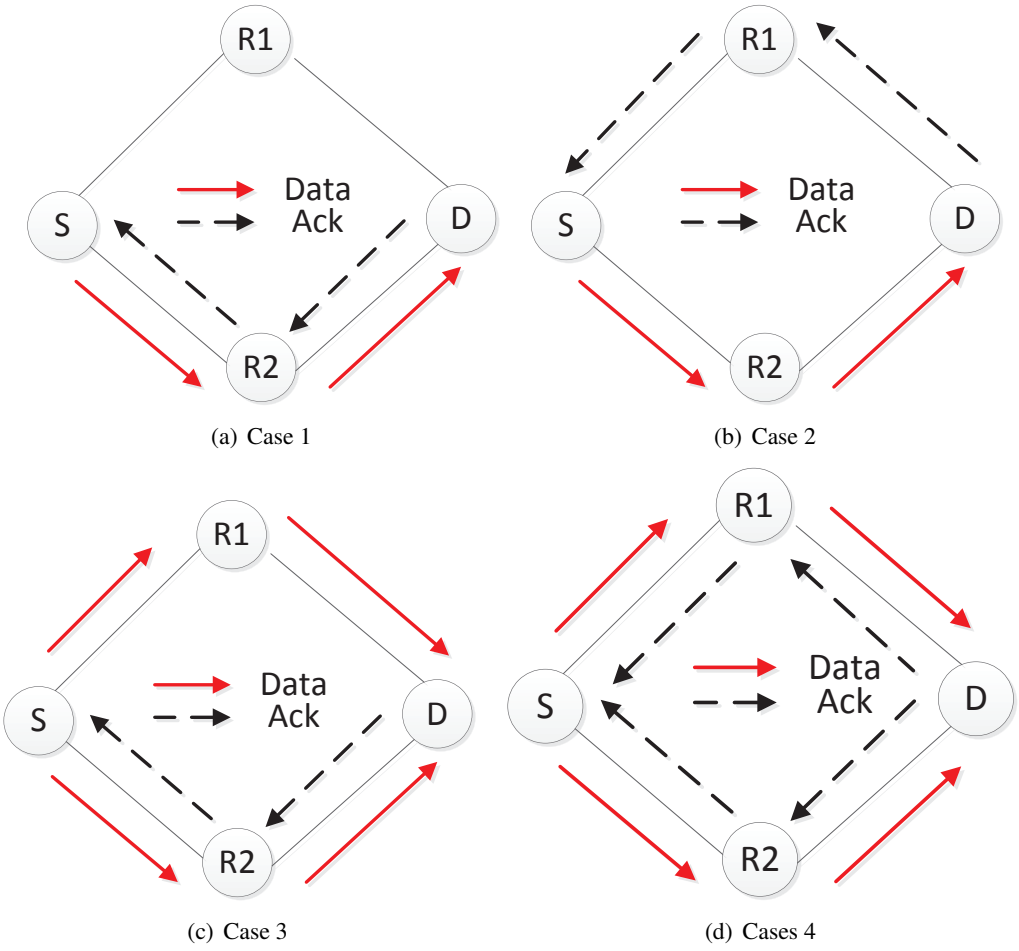
The chapter is organized as follows. In section 6.2, we first use a simple topology to show that node-disjoint multipath routing can improve the throughput of a directional antenna network. In Section 6.3, we study the performance of node-disjoint multipath routing in a larger randomly generated topology with more hops on the path. Based on the study, in Section 6.4, we propose a dynamic multipath routing algorithm that extends the existing DSR protocol. In Section 6.5, we evaluate the performance of different multipath routing discovery algorithms in the ns-2 simulator [42]. We also evaluate the performance of the node-disjoint multipath routing protocol for scenarios with one or multiple TCP flows in the network. In the last section, we conclude this chapter.

6.2 Node-disjoint routing with directional MAC protocols

In this section, we study the performance of a TCP flow that employs the node-disjoint multipath routing scheme and different directional MAC protocols in the example shown in Fig. 6.1. In the figure, the source node S initiates a TCP flow to destination node D. Since they are out of each other's transmission range, relay nodes R1 or R2 may be selected to forward packets.

We designed four test cases to evaluate the performance of different node-disjoint multipath routing algorithms. Case 1, as shown in Fig. 6.1(a), is the baseline scenario that both TCP data and ack choose the same path where there is no benefit from node-disjoint multipath routing. In case 2, node S sends TCP data packet via node R2 and node D sends TCP ack data via node

R1 as shown in Fig. 6.1(b). In case 3, node S tosses a coin to decide which relay node will be chosen to forward the current TCP data packet, while node D always chooses node R2 to forward TCP ack packets as shown in Fig. 6.1(c). In case 4, both node S and D toss a coin to decide which node will be chosen as the relay node as shown in Fig. 6.1(d).



(a) Case 1 (b) Case 2

(c) Case 3 (d) Cases 4

Figure 6.1: A simple scenario for node disjoint multipath routing

Table 6.1: Performance evaluation for node disjoint multipath routing

case	omni	EDMAC	DMAC
1	194	221	172
2	189	308	236
3	189	293	236
4	189	308	248

Table 6.1 shows the throughput of different cases using omni-directional and directional antennas. In the simulation, we assume the Equivalent Isotropically Radiated Power (EIRP) of the omni-directional antenna is equal to Effective Radiated Power (ERP) of the directional

antenna's main lobe direction. Thus, the data rates of links formed by omni-directional and directional antenna are the same. We choose DMAC and EDMAC as the MAC protocol for directional antenna.

In the omni-directional network, using the node-disjoint multipath routing does not improve the throughput of the TCP flow. This is because in an omni-directional antenna network, nodes S, R1 and R2 would not be able to transmit simultaneously since they are in the same carrier sense range. We even notice that the throughput using node-disjoint multipath route decreases a little bit. This is because in the single-path routing case (Case 1), there are only three nodes (Node S, R2 and D) contending for the channel, while in case 2, 3 and 4, there are four nodes (Node S, R1, R2 and D) contending for the channel in the node-disjoint multipath route cases. Collisions increase the Round Trip Time (RTT) of the TCP flow thus decrease the throughput of the TCP flow.

In a directional antenna network, however, we see in Table 6.1 that the node-disjoint multipath routing scheme increases the throughput of the TCP flow often by 30% or more over the single-path routing scheme. This is because the node-disjoint multipath routing scheme utilizes the spatial reuse enabled by directional antennas. This enables node-disjoint links to be active simultaneously. For example, node S can send the first TCP packet to R1, and then sends the second TCP packet to R2 while R1 is forwarding the first TCP packet to node D. Similarly, node D can send TCP ack packets to node S via nodes R1 and R2.

We also notice that EDMAC always performs better than the DMAC by choosing an optimized contention window that alleviates the deafness problem, as discussed in Chapter 4. In the single-path routing scheme, directional antenna networks using DMAC perform worse than the omni-directional antenna network because of the deafness problem. In contrast, the directional antenna network using EDMAC always performs better than the omni-directional antenna network. In particular, EDMAC using node-disjoint multipath routing increases the throughput of the TCP flow by more than 65%. Therefore, in this chapter, we use EDMAC as the MAC layer protocol.

6.3 Node-disjoint multipath routing in 60 GHz network

In this section, we study the performance of node-disjoint multipath routing in the same topology as in the previous chapter. In the simulation topology, 50 nodes (node 0 to node 49) are placed on a 45m$\times 20$m grid, i.e. the coordinates of node i is $(5 \times \lfloor i/10 \rfloor, 5 \times (i \mod 5))$. We then displace each node by a distance randomly selected from the interval [0,5] meters with a randomly selected direction. In this randomly generated topology, the source is typically about 6 hops away from the destination and there are more than two node-disjoint paths between them.

We take the single path routing as the baseline case, i.e case 1. In particular, as shown in Fig. 6.2, both TCP data and ack packets choose the same path from Node 0 to Node 49, i.e. the path along Node (0, 2, 13, 23, 34, 39, 49). In case 2, TCP data and ack packets use different node-disjoint paths. In particular, as shown in Fig. 6.3, TCP data packets from Node 0 to Node 49 choose the path along Node (0, 2, 13, 23, 34, 39, 49) while TCP ack packets from Node 49 to Node 0 choose the path along nodes (49, 48, 42, 31, 21, 10, 0). In case 3, TCP data and ack packets are split into two nodes-disjoint paths. In other words, when the source or the destination has a TCP packet to transmit, it will randomly choose a path from the two node-disjoint paths. In particular, as shown in Fig. 6.4, each TCP data packet from Node 0 to Node 49 randomly chooses a path from node-disjoint paths along nodes (0, 10, 21, 31, 42, 48, 49) and nodes (0, 2, 13, 23, 34, 39, 49). Similarly, each TCP ack packet from Node 49 to Node 0 randomly chooses a path from node-disjoint paths along nodes (49, 48, 42, 31, 21, 10, 0) and nodes (49, 39, 34, 23, 13, 2, 0). In case 4, TCP data and ack packets are split into three node-disjoint paths. In other words, when the source or the destination have a TCP packet to transmit, it will randomly choose a path from three node-disjoint paths. In particular, as shown in Fig. 6.5, each TCP data packet from Node 0 to Node 49 randomly chooses a path from node-disjoint paths (0, 10, 21, 31, 42, 48, 49), (0, 2, 13, 23, 34, 39, 49) and (0, 11, 22, 32, 43, 49). Similarly, each TCP ack packet from Node 49 to Node 0 randomly chooses a path from node-disjoint paths (49, 48, 42, 31, 21, 10, 0), (49, 39, 34, 23, 13, 2, 0) and (49, 43, 32, 22, 11, 0).

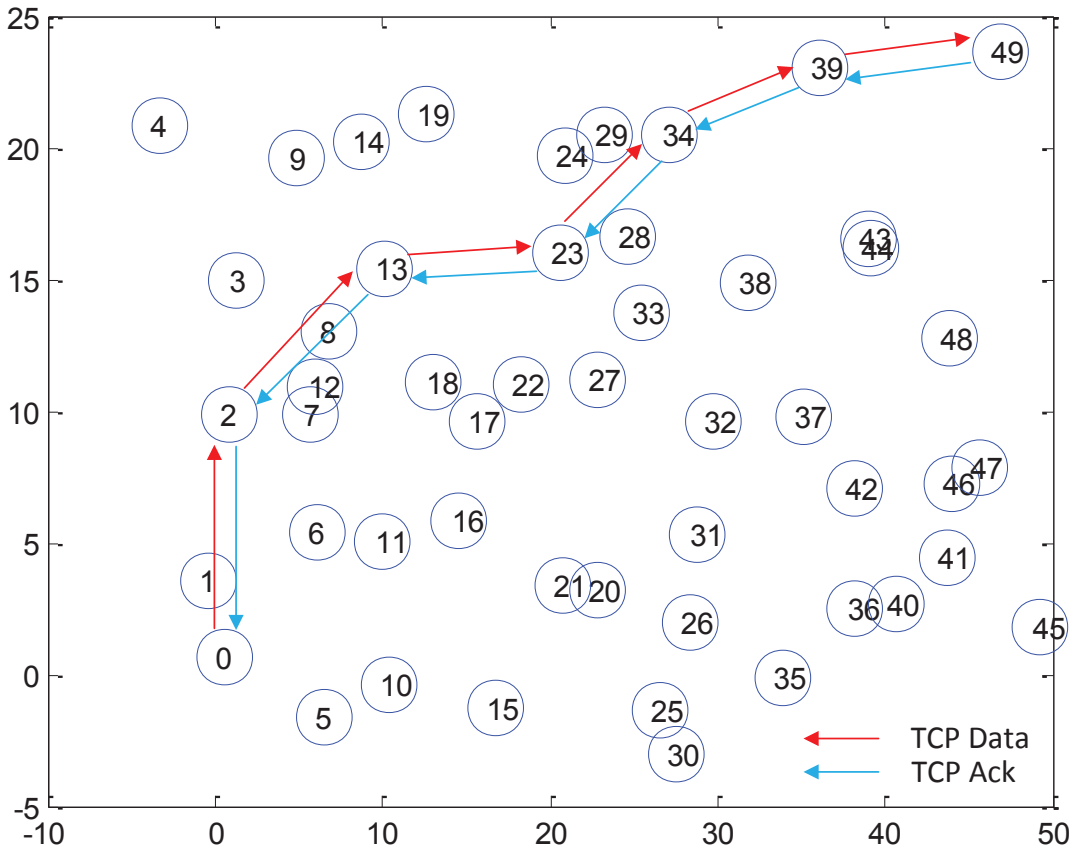


Figure 6.2: Case 1: single path routing between TCP source and destination

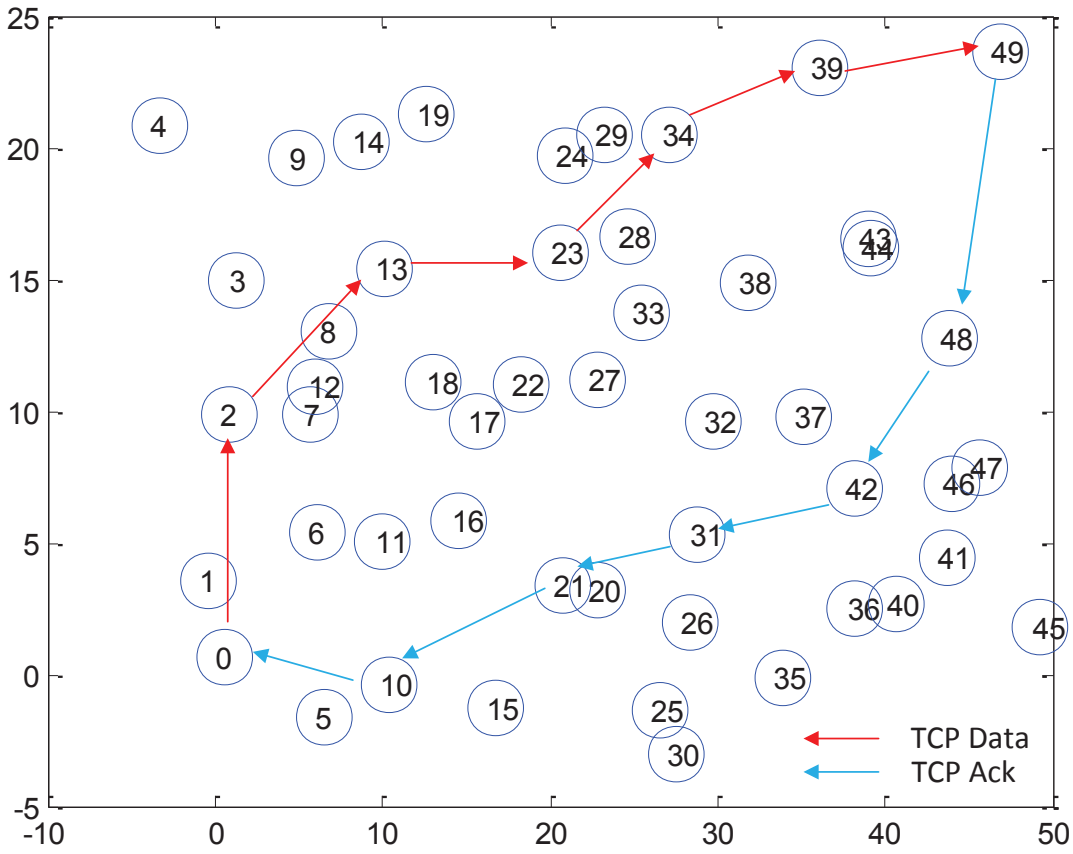


Figure 6.3: Case 2: TCP Data and Ack choose separate Paths

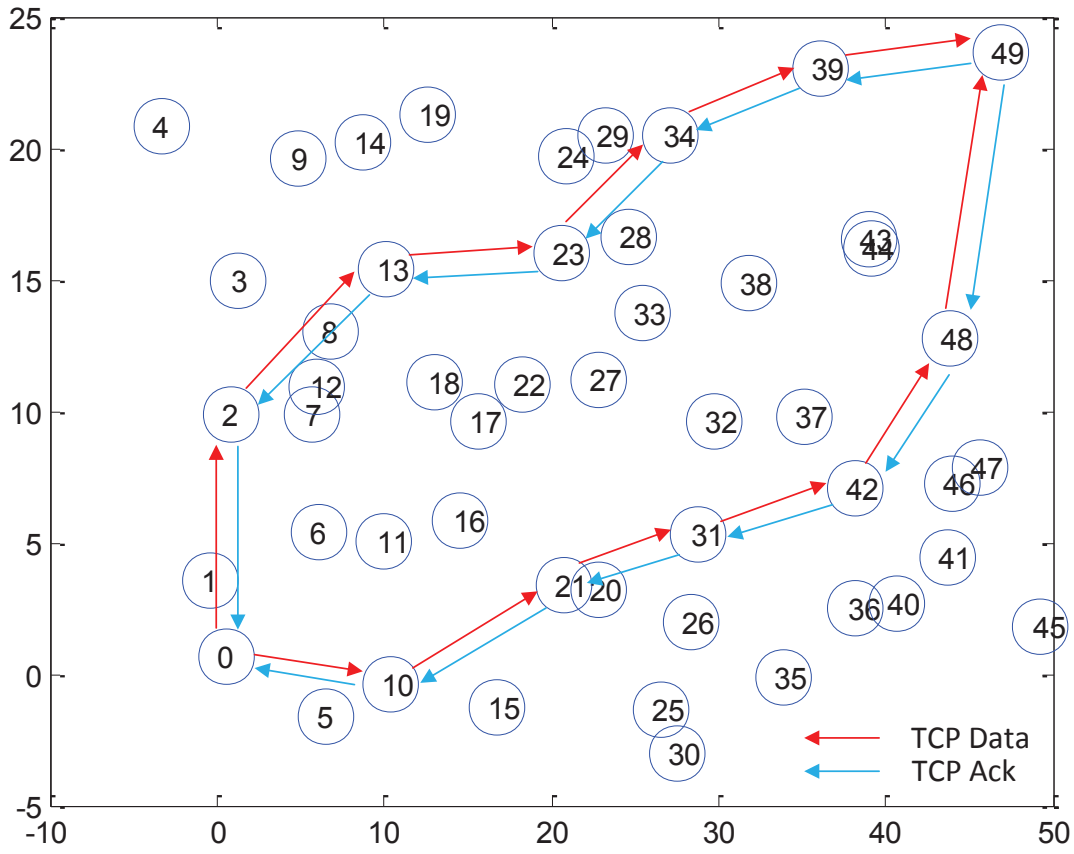


Figure 6.4: Case 3: TCP Data and Ack are split into two node-disjoint paths

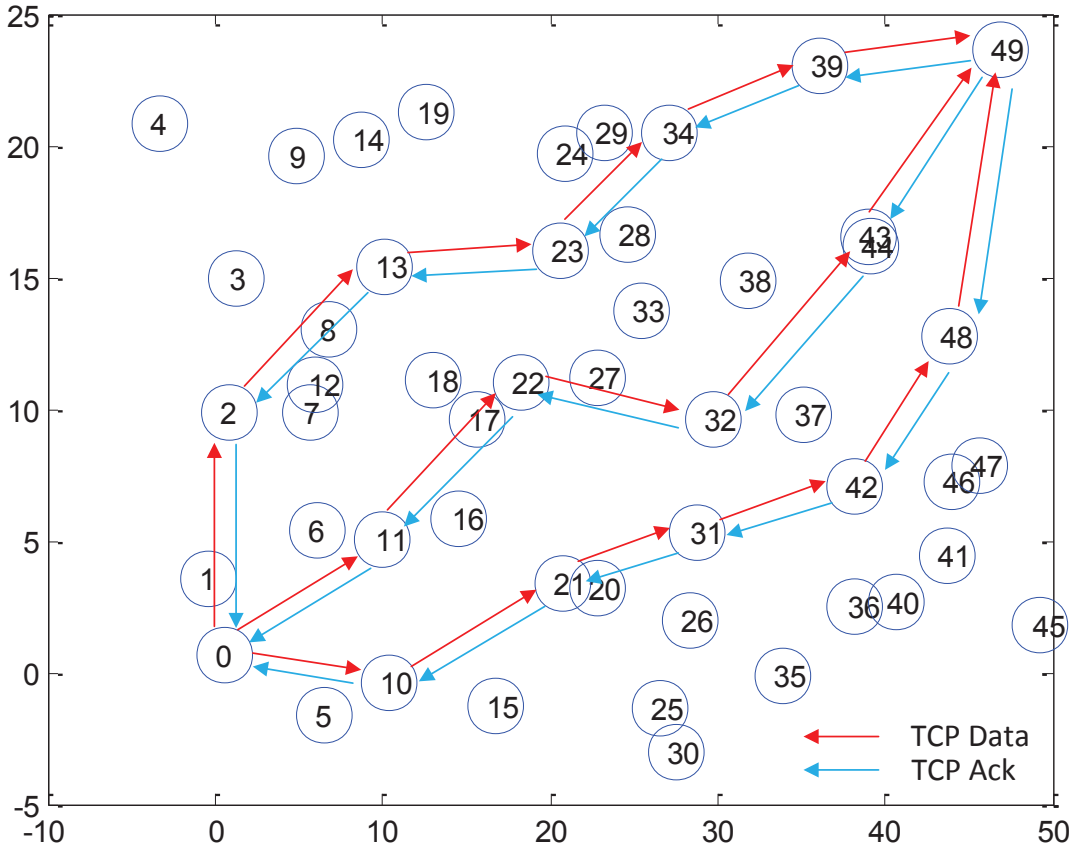


Figure 6.5: Case 4: TCP Data and Ack are split into three node-disjoint paths

Table 6.2: Performance evaluation for node disjoint multipath routing

case	Throughput (DupAck = 3)	Throughput (DupAck = 24)	Comment
1	156	156	Base line: Single Path
2	263	263	Multipath: Separate Data and ACK
3	234	258	Multipath: Split into 2 paths
4	265	311	Multipath: Split into 3 paths

The TCP throughput for the above four cases are shown in Table 6.2. Compared to single path routing case, using multiple node-disjoint routing increases the throughput by at least 50% due to spatial reuse enabled by directional antennas. Comparing with different multipath routing schemes, separating data and ack into different paths appears to have higher throughput than splitting TCP packets into two paths scheme when using TCP Tahoe with default parameters, i.e., the duplicated ACK threshold is set to 3 packets. This is because splitting TCP packets into different paths will introduce TCP packet reordering problem as studied in [16]. In particular, TCP data packets with higher sequence numbers may arrive earlier than the data packets with lower sequence numbers since they may take different paths. This may cause the source to receive multiple duplicated acks. In the default TCP Tahoe, three duplicated acks trigger the fast retransmission process, in which the source decreases its congestion window size. For example, the congestion window is reset to 1 in default TCP Tahoe. Figure 6.6 shows the congestion window size of the TCP flow for case 3 using default TCP Tahoe. We can see from Figure 6.6 that the congestion window oscillates frequently during the simulation time from 0 second to 2 second. These unnecessary fast retransmissions decrease the throughput of TCP flow.

There are many existing solutions to solve this problem as in [43]. In this chapter, we choose a simple solution among them by increasing the duplicated ack threshold from 3 to 24. By changing this parameter, the source only enter the fast retransmission stage when it receives 24 duplicated asks from the destination. Since the receive window of the TCP flow is set to 20 packets in our simulation experiments, no fast retransmission is triggered by setting the duplicated ack threshold to 24. After solving the packet reordering problem, the throughput of case 3 and case 4 increase by 10% and 27% respectively. Thus, the throughput of case 3 is close to the throughput of case 2.

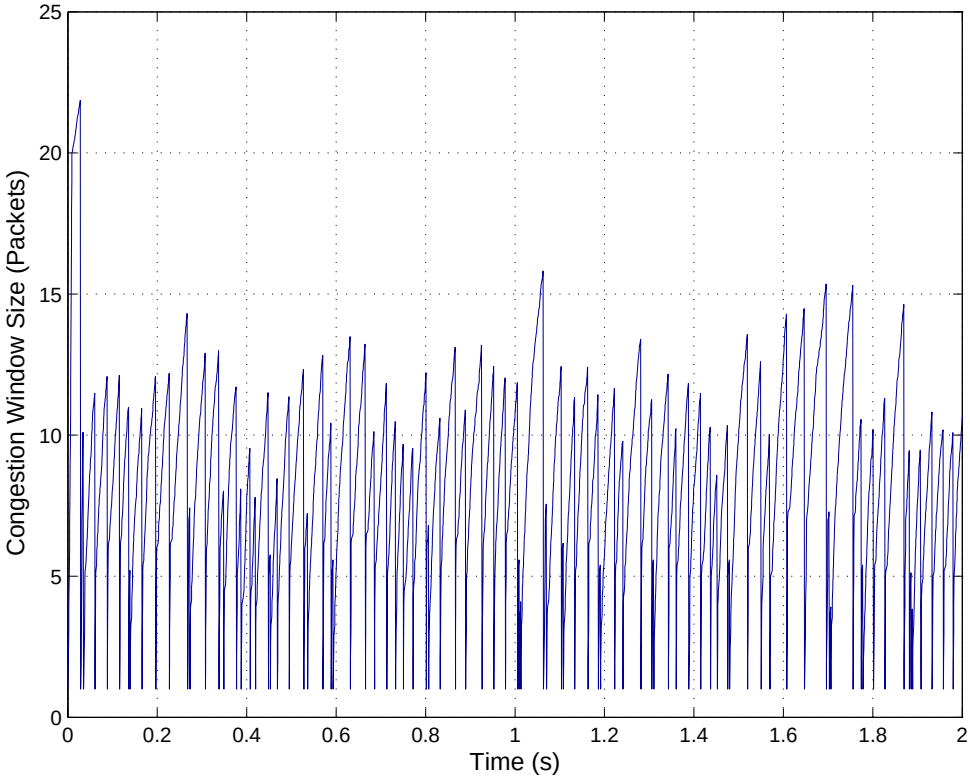


Figure 6.6: Congestion window size of the TCP flow in Case 4.

On the other hand, separating data and ack into different paths may introduce extra negotiation overhead between the source and the destination. In particular, the source and the destination are required to exchange information about the path selected in order to make sure they use different paths. More overhead will be introduced when links of a network change dynamically, e.g. due to the pedestrian blockage described in next chapter. Therefore, we choose the multipath routing scheme by simply splitting TCP packets into multiple node-disjoint paths. For example, as in case 4, by using three node-disjoint paths, the throughput of the TCP flow increases to 311 Mbps after solving the packet reordering problem. In the next section, we introduce a dynamic node-disjoint multipath routing protocol for directional antenna networks.

6.4 Dynamic node-disjoint multipath routing for directional antennas

In this section, we propose a dynamic node-disjoint multipath routing protocol that enables the source to transmit packets via multiple node-disjoint paths to the destination in a directional antenna network. Similar to DSR, the proposed protocol is a source-initiated on-demand routing protocol. When the source node attempts to send a packet to the destination node, it first checks its route cache. If there is no route to the destination, the source node initiates a route discovery process as described in Section 6.4.1 to find multiple routes to the destination. Among the discovered paths, a route selection process described in Section 6.4.2 will determine the node-disjoint paths that will be used.

6.4.1 Route Discovery

In the route discovery process in an omni-directional antenna network, the source node broadcasts a Route Request (RREQ) packet, i.e. the source node sends the RREQ without receiving acknowledgement from its neighbors. Nodes that receive an RREQ selectively rebroadcast the RREQ until the RREQ reaches the destination. However, nodes using directional antennas are not able to broadcast an RREQ to all neighbors at once. Therefore, in the directional DSR protocols [39] [40] [41], nodes use a "Beam Sweep" method that sends out RREQ packets to each beam direction one by one without expecting any acknowledgement. In this chapter, we assume instead that nodes employ a neighbor discovery protocol as in [44] and have learned

the directions and MAC addresses of all its neighbors. There are two advantages for a node to know the directions of all its neighbors.

First, the node can transmit an RREQ in each neighbor's direction instead of each beam direction. This "Neighbor Sweep" scheme may reduce the number of RREQs transmitted if the number of neighbors is smaller than the number of beam directions a node has to sweep. For example, in the simulation topology studied in this chapter, each node has about 12 neighbors on average. In contrast, each node has to sweep 36 beam directions in order to cover all directions using directional antennas with a 10 degree main beam width. Thus, the number of RREQs transmitted during a route discovery process can be reduced by an order of magnitude as shown in the simulation results later.

Second, a node can transmit an RREQ to each neighbor using unicast by knowing the neighbors' MAC addresses. In other words, a node can retransmit an RREQ to a neighbor several times before it receives an acknowledgement from the neighbor. This is different from the "Neighbor Sweep" scheme in which a node transmits an RREQ to each of its neighbors' directions only once. We name this scheme "Neighbor Unicast." In directional antenna networks, the deafness problem mentioned in [36] may have severe negative effects on the route discovery process if an RREQ is transmitted without acknowledgement. For example, in the simplified multipath scenario shown in Fig. 6.1(d), the source node first transmits an RREQ to node R1 and then transmits an RREQ to node R2. However, since R1 and S cannot sense each other, a collision may happen when R1 and S transmit an RREQ to R2 at the same time. When this happens, the source node fails to discover the two node disjoint paths to the destination. Moreover, even if both R1 and R2 receive the RREQ from the source, the RREQs they rebroadcast are likely to collide at the destination. Thus, no route may be found until the source initiates another RREQ after the route discovery request timer expires. Compared to the broadcast sweep scheme, the unicast scheme with packet retransmissions at the MAC layer proposed here may find more paths and node-disjoint paths from the source to the destination.

RREQs will be forwarded by nodes in the network until they reach the destination or their Time to Live (TTL) timer expires. In the original single-path DSR, a node only forwards the first RREQ it received from the same source with the same sequence number, and piggybacks its address information into the RREQ. We name this forwarding scheme "First Only." The

First Only forwarding scheme may introduce two problems in a dynamic node-disjoint multi-path routing protocol in a directional antenna network. First, due to the antenna sweeping, the first RREQ a node receives may not come from the shortest path, as mentioned in [39]. Therefore, the route discovery process is more likely to find longer routes from the source to the destination. Second, the First Only forwarding scheme may fail to find multiple node-disjoint paths as mentioned in [38]. Therefore, in this chapter, we compare two modifications of the RREQ forwarding scheme as follows.

The first scheme named "Shorter" enables a node to forward the RREQ received from a path that is shorter (strictly less) than all RREQs it has received from the same source with the same sequence number. The second scheme, named "Not Longer", allows a node to forward the RREQ received from a path that is not longer than all RREQs it has received from the same source with the same sequence number. In this chapter, we use hop counts to measure the length of a path, since an RREQ contains all nodes it has traversed and the hop count information can be easily obtained.

The destination will send a Route Reply (RREP) back to the source after it receives an RREQ message. An RREP message contains a route from the source to the destination learned from the RREQ message received. An RREP message traverses to the source along the same route it contains and the source stores the route into its route cache. In the original single-path DSR, the destination only replies to the first RREQ it receives. We name this reply scheme "First Only." The "First Only" reply scheme sends an RREP message to the source containing only one route. Thus, the source can only discover at most one route at the end of a route discovery procedure. In other words, the "First Only" reply scheme prevents the source from finding multiple paths in one discovery process.

Many RREQ reply schemes can be used to find multiple paths in one discovery process. In one scheme, the destination can reply the first $k > 1$ RREQs it receives. In another scheme [39], the destination can reply to RREQs it receives within a certain time period after it receives the first RREQ. In yet another approach [38], the destination can reply to the first RREQ and to RREQs that come from paths that are node-disjoint with the path found by the first RREQ. In this chapter, we employ the "Reply All" scheme that allows the destination to reply all RREQ it receives. Although the "Reply All" scheme creates extra overhead for transmitting more RREP

messages, it may have the following benefits.

First, comparing with the scheme that the destination only replies to RREQs from node-disjoint paths, the "Reply All" scheme enables the source and the destination to establish a cache of alternate paths. While these routes are not all node-disjoint, the "Reply All" scheme may be more robust in maximizing the number of node-disjoint paths when link failures occur after route discovery. For example, if the source only finds three node disjoint paths and one of those three paths goes down due to a broken link, the source will have only two node-disjoint paths left. In contrast, with the "Reply All" reply scheme, the source may be able to activate an alternate node-disjoint path from its cache that bypasses the broken link.

Second, using the "Reply All" scheme, the source and destination can select node-disjoint paths independently. This allows the source and the destination to use different paths to transmit data and ack, which may increase the throughput due to less intra-flow interference. The next subsection explains how the source and the destination select node-disjoint paths.

6.4.2 Node-disjoint paths selection

The maximum number of node-disjoint paths can be found by using max flow algorithms, such as the Ford-Fulkerson algorithm in [45]. First, the network topology is mapped to an undirected graph $G = (V, E)$, then each undirected edge is replaced by two directed edges (u, v) and (v, u) with an unit capacity. Then each vertex $v \in V$ is split into 2 vertices v_{in} and v_{out} and a unit capacity edge is added to connect v_{in} and v_{out}. All edges that are destined to node v are connected to v_{in} and all edges that come from node v are connected to v_{out}. Then the maximum flow paths found by the max flow algorithm will be the maximum node-disjoint paths found in the network [45].

However, there are some constraints on using the maximum flow algorithm. First, the topology of the network is required to be known and stored. For example, the source uses a link cache to store all available links in the network. However, the default DSR uses a path cache that stores all paths from the source to the destination. Therefore, extra effort is required to transfer the path cache to a link cache. Second, the complexity of the max flow algorithm is $O(V^2E)$ without using a complicated data structure [45]. Therefore, the computational overhead may be high if the topology of the network changes frequently. Third, the max flow

Algorithm: Find the node-disjoint paths set to the destination
Input: Candidate Set S_C contains all paths in the route Cache
Output: Node-disjoint multipath set S_{NDMP}
Method:
1. **WHILE** $S_C \neq \emptyset$
2. Find the shortest path p_s in S_C
3. Add path p_s into S_{NDMP}
4. Remove all paths in S_C that are not node-disjoint with p_s

Figure 6.7: The greedy algorithm to find all node-disjoint paths to the destination.

algorithm does not constrain the length of node-disjoint paths found and these longer paths may result in lower throughput. Therefore, in this chapter, the greedy algorithm shown in Fig. 6.7 is proposed to obtain the shortest node-disjoint paths set S_{NDMP} among the paths cached in the route discovery process.

Whenever a source sends a packet to the destination, it will randomly choose a path from S_{NDMP} and put the path information into the packet source header. The packet will be forwarded to the destination along the route. In this chapter, we do not consider any link broken problem. In other words, the S_{NDMP} does not change after the route discovery process. We will consider the link broken problem due to human blockage in the next chapter.

6.5 Performance evaluation

In this section, we evaluate the proposed dynamic multipath routing scheme in a static network, i.e. there is no link broken due to mobility or blockage in the network. We first evaluate the performance of the maximum flow algorithm and the proposed greedy algorithm that can find node-disjoint paths from the source to the destination. In particular, we evaluate these two algorithms offline, i.e. we assume the source knows the topology of the whole network, and employ these two algorithms to obtain node-disjoint paths. Second, we evaluate the performance of the proposed dynamic node-disjoint multipath routing schemes in various scenarios, i.e. we assume the source has no knowledge about the network and the source starts a route discovery process to find routes to the destination.

6.5.1 Performance of offline node-disjoint path algorithms

In this subsection, we assume the source node knows the topology of the network, i.e. the graph $G = (V, E)$ of the network, where V is the set of nodes and E is the set of links. First, the source node uses "graphmaxflow" function in MATLAB to obtain a set of node-disjoint paths that contains the maximum number of node-disjoint path in the network. Then, the source node obtain another set of node-disjoint path using the proposed greedy algorithm in Fig. 6.7. We randomly generate 80 simulation topologies using the same rule described in Section 6.2. Figure 6.8 and Figure 6.9 show the number of node-disjoint paths and the average length of the paths found by both algorithms for a flow from node 0 to node 49. As seen in Figure 6.8, the maximum flow algorithm can always find the maximum number of node-disjoint paths and the greedy algorithm can find the maximum number of node-disjoint paths in most of times. The average ratio of number of paths found by the greedy algorithm over the maximum flow algorithm for these 80 topologies is 0.95. Moreover, as seen in Figure 6.8, the average length of node-disjoint paths found by the maximum flow algorithm is always longer than the proposed greedy algorithm. The average ratio of the average length of node-disjoint paths found by the proposed greedy algorithm over the maximum flow algorithm for these 80 topologies is 0.67.

Figure 6.10 and Figure 6.11 show the average number of node-disjoint paths and the average length of paths found by both algorithms for 5 flows, where the ith flow is from node $i - 1$ to node $50 - i$. Similar with single flow scenario, the maximum flow algorithm can always find the maximum number of node-disjoint paths and the proposed greedy algorithm can find the maximum number of node-disjoint paths for most of times as seen in Figure 6.10. The average ratio of the number of paths found by the greedy algorithm over that found by the maximum flow algorithm for these 80 topologies is 0.95, which is the same as the single flow scenario. The average length of node-disjoint paths found by the maximum flow algorithm is also always longer than the proposed greedy algorithm as seen in Figure 6.10. The average ratio of the average length of node-disjoint paths found by the greedy algorithm over the found by the maximum flow algorithm for these 80 topologies is 0.77, which is a little bit larger than the single flow scenario.

To study the performance of TCP flows using paths found by the maximum flow algorithm

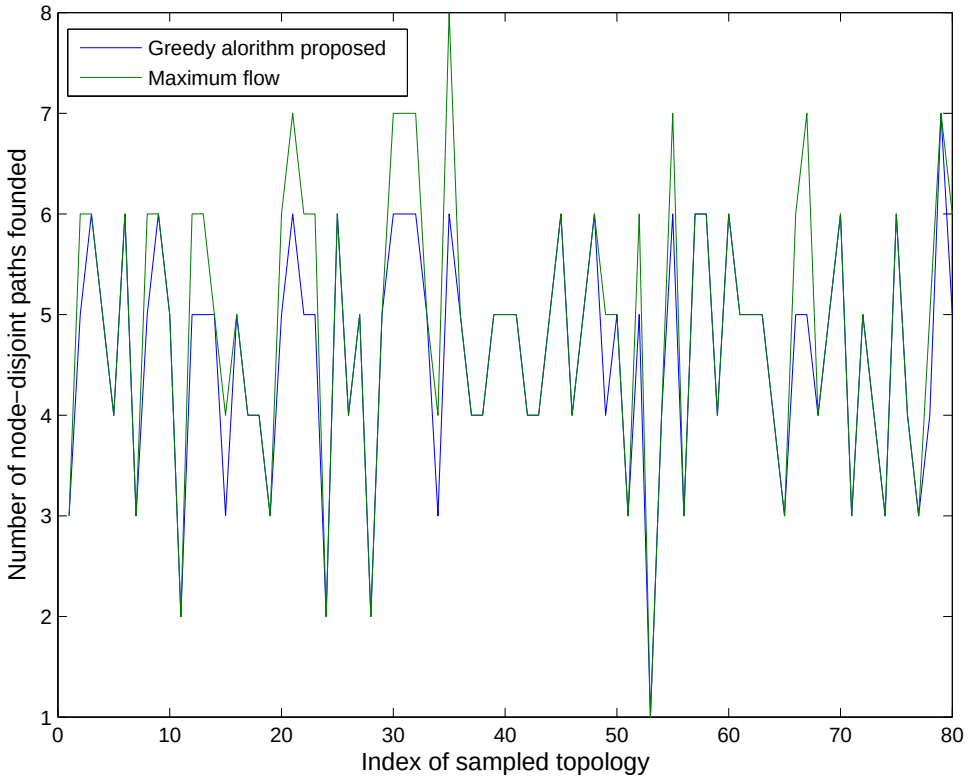


Figure 6.8: Number of node-disjoint paths from node 0 to node 49 found by the maximum flow algorithm and the proposed greedy algorithm.

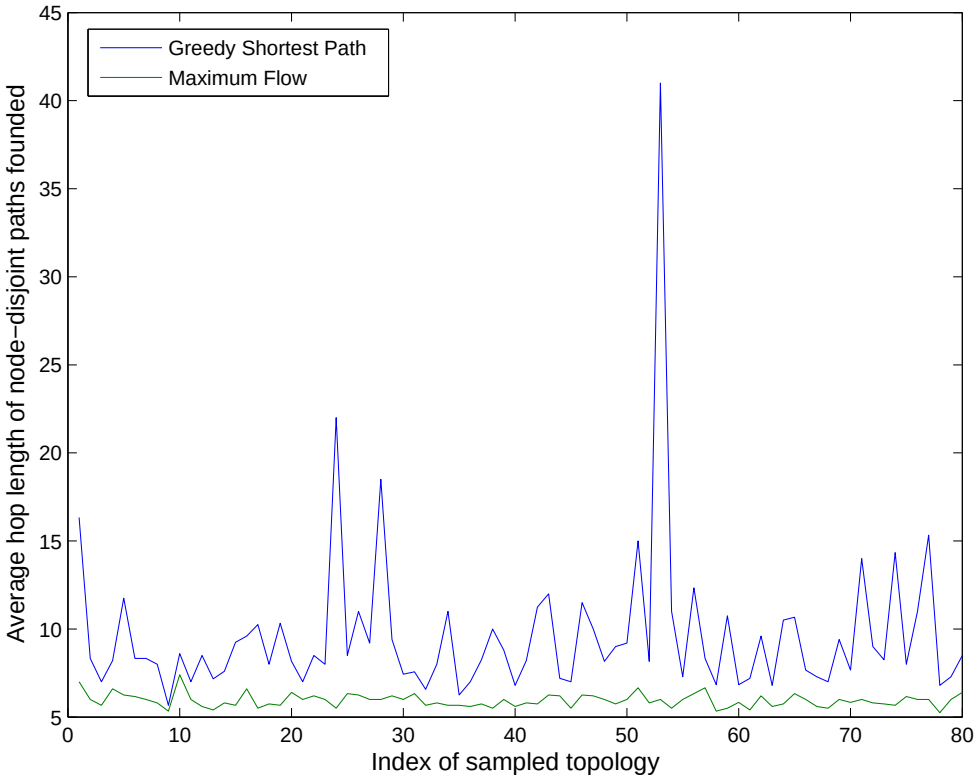


Figure 6.9: Average length of node-disjoint paths from node 0 to node 49 found by the maximum flow algorithm and the proposed greedy algorithm.

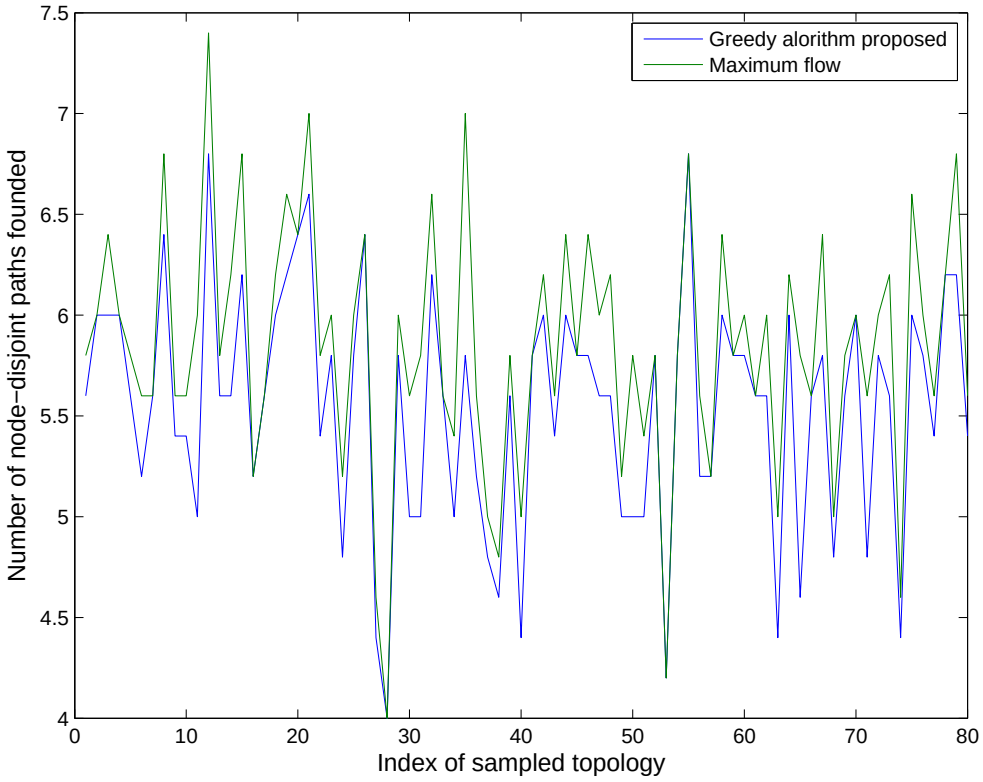


Figure 6.10: Number of node-disjoint paths from for 5 flows found by the maximum flow algorithm and the proposed greedy algorithm.

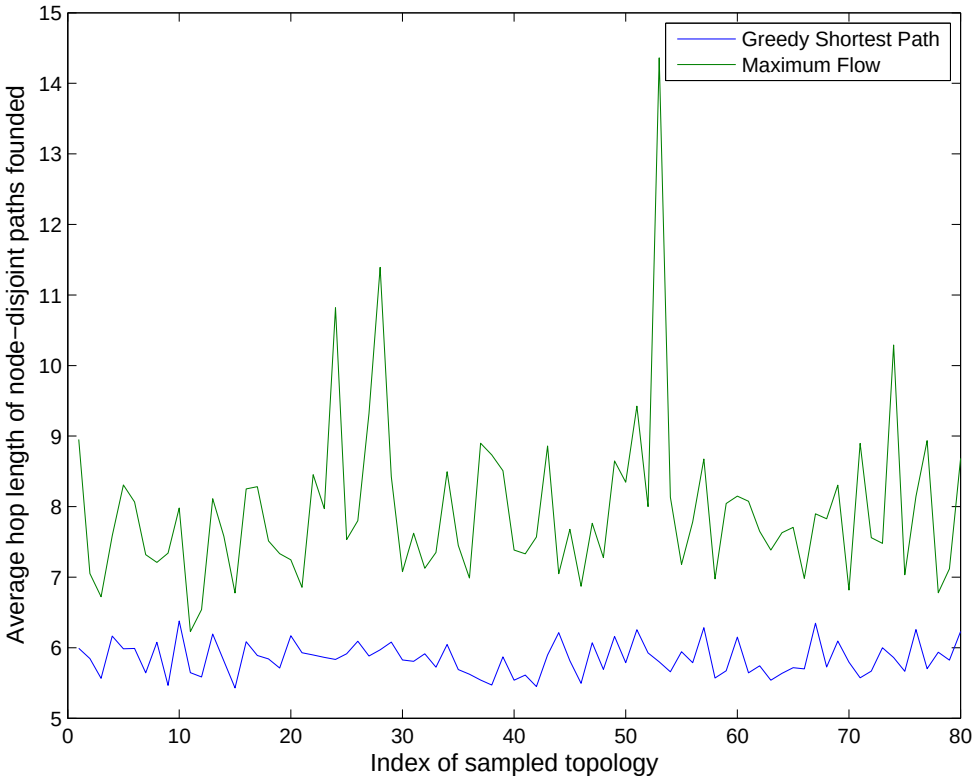


Figure 6.11: Average length of node-disjoint paths for 5 flows found by the maximum flow algorithm and the proposed greedy algorithm.

and the proposed greedy algorithm, Figure 6.12 shows the throughput of the flow from node 0 to node 49 using node-disjoint paths found by each algorithm for the same 80 topologies. The average throughput using the proposed greedy algorithm is 232 Mbps, which is higher than the value using the maximum flow algorithm 192 Mbps. This is because the average length of node-disjoint paths found by the proposed greedy algorithm is shorter, and the shorter paths provide higher throughput due to smaller Round Trip Time for the TCP flow as studied in Chapter 4.

Another interesting question is whether the throughput of the TCP flow increases when using more node-disjoint paths. Figure 6.13 shows the throughput ratio of the TCP flow from node 0 to node 49 using different number of node-disjoint paths over using single path.The throughput stops increasing when the flow uses more than four node-disjoint paths for both algorithms. Therefore, the extra node-disjoint paths found by the maximum flow algorithm cannot increase the TCP throughput if the proposed greedy algorithm could also find four or more node-disjoint paths. This also explains why the average hop length of the node-disjoint paths dominates the throughput of the flow, and thus the proposed greedy algorithm performs better than the maximum flow algorithm.

6.5.2 Performance of route discovery schemes

In previous subsection, we study the performance of two algorithms based on an assumption that the source node in the network has learned the topology of the network. However, this assumption is difficult to meet in real scenarios. First, gathering the topology information of the whole network introduces large overhead, such as flooding topology control messages in OLSR [14]. Moreover, when the topology of the network change frequently due to broken links, it is impossible for nodes in a large network to obtains the current network topology. In this subsection, we evaluate different route discovery schemes proposed in Section 6.4 that allows the source to find multiple node-disjoint paths to the destination. The proposed route discovery schemes are measured by the following metrics.

- N^{RREQ}, the total number of RREQ messages transmitted by nodes in the network, which measures the RREQ broadcast overhead.

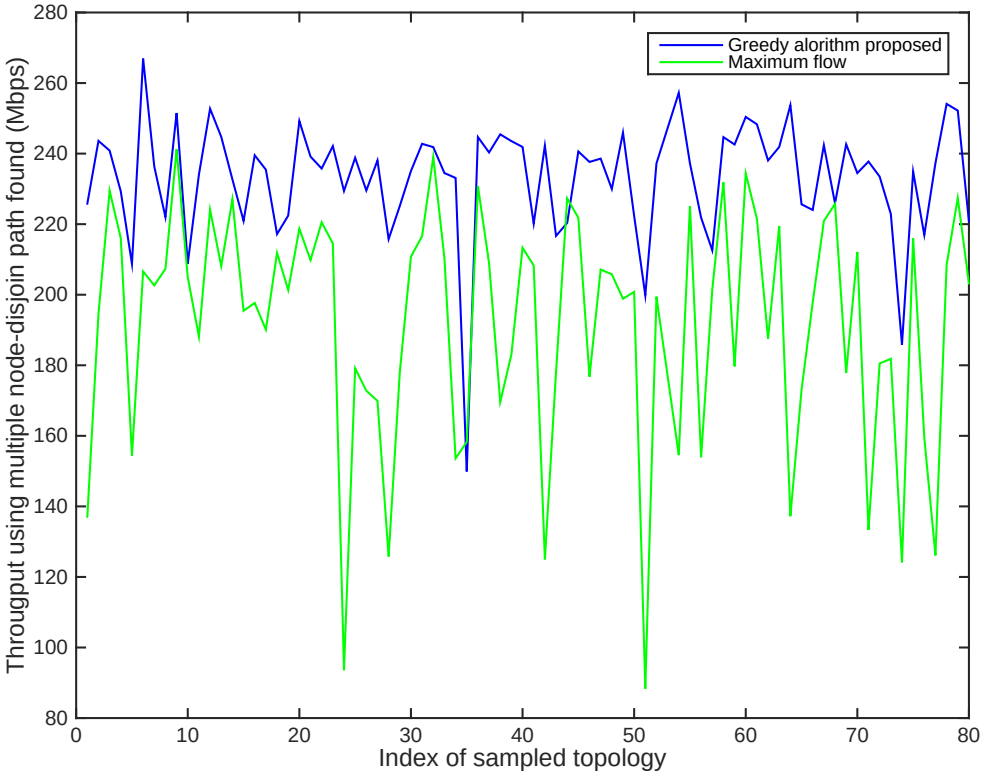


Figure 6.12: Throughput of the flow from node 0 to node 49 using node-disjoint paths found by maximum flow algorithm and the proposed greedy algorithm.

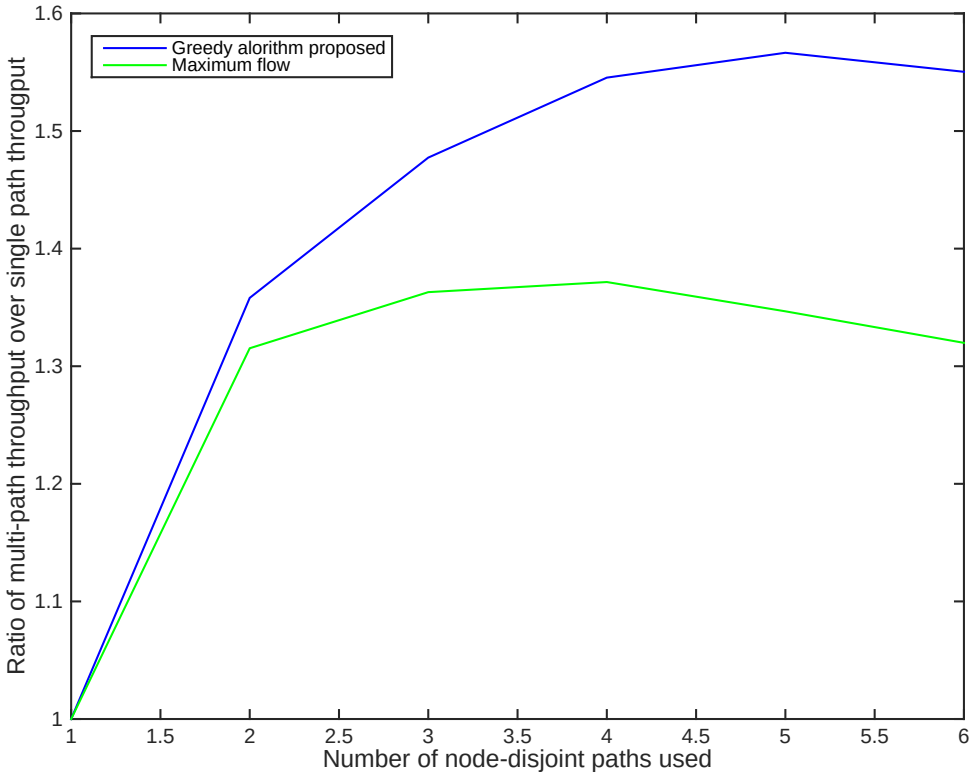


Figure 6.13: Throughput ratio of the flow from node 0 to node 49 using different number of node-disjoint paths over using single path found by maximum flow algorithm and the proposed greedy algorithm.

Table 6.3: Performance evaluation for difference route discovery schemes

RREQ Tx	RREQ FW	N^{RREQ}	$N_{\text{Send}}^{\text{RREP}}$	$N_{\text{Src}}^{\text{Path}}$	$N_{\text{Src}}^{\text{NDPath}}$	$N_{\text{Dst}}^{\text{Path}}$	$N_{\text{Dst}}^{\text{NDPath}}$	Throughput (Mbps)
Beam Sweep	First Only	1764	3	3	1	3	1	148
Beam Sweep	Shorter	2196	4	4	1	4	1	150
Beam Sweep	Not Longer	38589	289	80	3	85	3	238
Neighbor Sweep	First Only	568	4	4	1	4	1	134
Neighbor Sweep	Shorter	603	4	4	1	4	1	141
Neighbor Sweep	Not Longer	11854	185	121	4	121	4	268
Neighbor Unicast	First Only	568	4	4	1	4	1	167
Neighbor Unicast	Shorter	861	9	9	2	9	2	227
Neighbor Unicast	Not Longer	14181	305	288	3	300	4	297

- $N_{\text{Send}}^{\text{RREP}}$, the total number of RREP messages sent by the destination. This is also the number of RREQs that the destination receives since the destination will reply to all RREQs it receives.
- $N_{\text{Src}}^{\text{Path}}$, the number of paths discovered from the source to the destination.
- $N_{\text{Src}}^{\text{NDPath}}$, the number of node-disjoint paths discovered from the source to the destination.
- $N_{\text{Dst}}^{\text{Path}}$, the number of paths discovered from the destination to the source.
- $N_{\text{Dst}}^{\text{NDPath}}$, the number of node-disjoint paths discovered from the destination to the source.

To summarize, we compare

- Three RREQ transmitting schemes, i.e. “Beam Sweep”, “Neighbor Sweep” and “Neighbor Unicast”;
- Three RREQ forwarding schemes, i.e. “First Only”, “Shorter” and “Not Longer”

The performance of 9 combinations of different RREQ transmitting and forwarding schemes are summarized in Table 6.3. By utilizing the neighbor information, the “Neighbor Sweep” and “Neighbor Unicast” RREQ transmitting schemes reduce the number of RREQ nodes transmitted to approximate $1/3$ of the “Beam Sweep” scheme for all forwarding schemes. This is because using the “Beam Sweep” scheme, a node transmits the RREQ to all its beam directions, i.e. 36 beam directions in the simulation. In contrast, in the “Neighbor Sweep” and “Neighbor

Unicast" schemes, a node only transmits the RREQ to all its neighbors, i.e. 12 neighbors on average in the simulation topology.

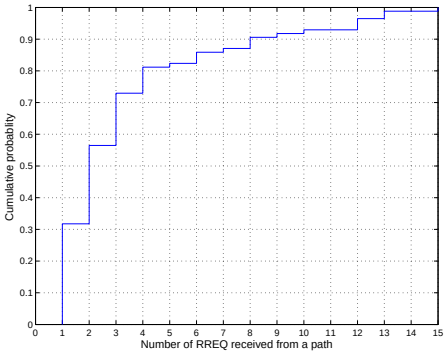
The "Neighbor Unicast" transmitting scheme has a higher number of RREQs transmitted than the "Neighbor Sweep" transmitting scheme especially for the "Shorter" and "Not Longer" forwarding schemes. This is because in the "Neighbor Unicast" transmitting scheme, when a node forwards an RREQ to a neighbor, it will retransmit the RREQ to the neighbor until it receives an ack from the neighbor or it reaches maximum retransmission times. In contrast, in the "Neighbor sweep" transmitting scheme, a node only sends RREQ to each neighbor once. Due to the deafness problem as mentioned in [36], some neighbors may not receive the RREQ successfully. Therefore, there will be fewer nodes in the network that receive the RREQ message and thus fewer RREQs are forwarded using the "Neighbor Sweep" scheme. This is also the reason that the destination sends fewer RREPs using "Neighbor Sweep" than using "Neighbor Unicast."

Using "Sweep" transmitting schemes, i.e "Beam Sweep" and "Neighbor Sweep" transmitting schemes, some RREQs arrive at the destination come from the same route when using "Not Longer" forwarding scheme. For example, using the "Beam Sweep" transmitting scheme and "Not Longer" forwarding scheme, among 255 RREQs received by the destination, only 85 of them come from distinct paths. This is because when a node transmits RREQs to each beam direction,

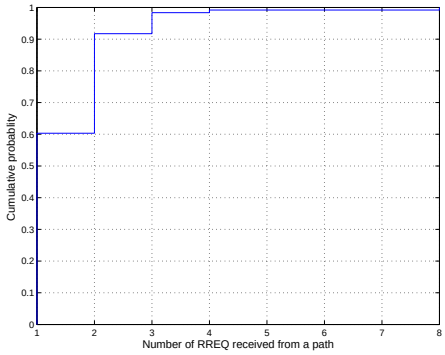
- some of its neighbors that sit at the boarder of two beams may receive and then rebroadcast two RREQs that come from two different beams;
- some of its neighbors that are close may receive multiple RREQs from side lobe beams.

Fig. 6.14(a) shows the cumulative distribution function for the number of RREQs that come from each of these 85 distinct paths. Among the 85 distinct paths, 31% of them, i.e. 27 paths are reported by an unique RREQ. However, most of paths are reported by more than one RREQs. In the extreme case, 15 duplicated RREQs come from the same path $(0, 11, 22, 28, 34, 39, 49)$. In this case, Node 11 receives two RREQs from Node 0 because node 11 sits at the boarder of two beams of Node 0 and Node 34 receives 15 RREQs from the main lobe beam and side lobe beams of Node 28 because these two nodes are close to each other.

Similarly, Fig. 6.14(b) shows the cumulative distribution function for number of RREQs that come from each of 121 paths found using "Neighbor Sweep" transmitting scheme and "Not Longer" forwarding scheme. Among these 121 distinct paths, 60% of them are reported by a single RREQ and fewer paths are reported by more than one RREQs comparing with using the "Beam Sweep" transmitting scheme. Based on the above analysis, to reduce the number of duplicated RREQs that come from the same path in the "Beam Sweep" and "Neighbor Sweep" transmitting schemes, nodes in the network are not only required to memorize the path of each RREQ they received but also drop the RREQ that comes from the same path with the same sequence number. However, this may consume extra resource, e.g CPU and memory, and reduces the number of paths found in a route discover process. In comparison, the "Neighbor Unicast" transmitting scheme does not suffer with this problem, since a node only receives a RREQ designated to it. Therefore, the "Neighbor Unicast" scheme may consume fewer resource than other two schemes and also finds more paths in a route discovery process.



(a) Cumulative distribution function for the number of RREQs that comes from each of 85 distinct paths using the "Beam Sweep" transmitting scheme and the "Not Longer" forwarding scheme.



(b) Cumulative distribution function for the number of RREQs that comes from each of 121 distinct paths using the "Neighbor Sweep" transmitting scheme and the "Not Longer" forwarding scheme.

Figure 6.14: Cumulative distribution function for the number of RREQs that comes from each path for different RREQ transmitting and forwarding schemes.

Among the forwarding schemes, "Not Longer" finds an order of magnitude more paths than the "First Only" and "Shorter" forwarding schemes. At the same time, the number of RREQs transmitted using the "Not Longer" scheme during a route discovery process is much larger as the other two forwarding schemes. However, the "First Only" and "Shorter" schemes cannot find multiple node-disjoint paths in most scenarios. In comparison, the "Not Longer" scheme

may find almost all node-disjoint paths.

Since multipath routing can achieve higher throughput than the single-path routing, the "Not Longer" forwarding scheme achieves higher throughput than the other two forwarding schemes. For the "First Only" forwarding scheme, all three RREQ transmitting schemes have similar throughput since none of them can find multiple node-disjoint paths. For the "Shorter" forwarding scheme, the "Neighbor Unicast" RREQ transmitting scheme achieves higher throughput than the other two schemes since it can find two node-disjoint paths. This is because more nodes in the network can receive RREQs using the "Neighbor Unicast" transmitting scheme that employs RREQ retransmissions. In contrast, in the "Sweep" transmitting schemes, nodes send to each neighbor or beam direction only once. While using the "Not Longer" forwarding scheme, the "Neighbor Unicast" and "Neighbor Sweep" transmitting schemes that transmit RREQ to all neighbors can find more node-disjoint paths than the "Beam Sweep" transmitting scheme that transmits RREQs to all beam directions. This is because in the simulation topology, the number of neighbors of a node is smaller than the number of beam directions, thus there are fewer RREQs transmitted in the route discovery process, which reduces the number of RREQ collisions. Thus, more nodes in the network can receive and forward the RREQ and more paths and node-disjoint paths can be found.

To summarize, the "First Only" RREQ forwarding scheme should not be selected since it cannot find multiple node-disjoint paths. The "Beam Sweep" transmitting scheme also should not be used since it introduces more RREQs transmission overhead and finds fewer node-disjoint paths. The "Not Longer" forwarding schemes can be used in order to find the maximum number of node-disjoint paths. Along with using the "Not Longer" forwarding scheme, the "Neighbor Unicast" transmitting scheme can find more than twice as many paths than the "Neighbor Sweep" transmitting scheme while introducing only 30% more RREQ transmissions. The "Shorter" forwarding scheme can reduce the RREQs transmission by the order of magnitude comparing with "Not Longer," however, it finds far fewer node-disjoint paths. For example, as shown in Table 6.3, two node-disjoint paths are found using the "Neighbor Unicast" transmitting scheme and no node-disjoint path is found using the "Neighbor Sweep" transmitting scheme.

Furthermore, different combinations of RREQ transmitting schemes and RREQ forwarding schemes have their advantages and disadvantages. For example, using the "Neighbor Unicast" transmitting scheme along with the "Not Longer" forwarding scheme finds the maximum number of paths and node-disjoint paths but it has highest RREQ transmission overhead. Using the "Neighbor Unicast" transmitting scheme along with "Shorter" forwarding scheme has the least RREQ transmission overhead and but finds a limited number of node-disjoint paths, e.g. only two node-disjoint paths in the simulated topology. Thus, what scheme should be chosen depends on the scenario. For example, the "Neighbor Sweep" or the "Neighbor Unicast" RREQ transmitting scheme with the "Not Longer" RREQ forwarding scheme may perform better for the static network evaluated in this chapter.

6.5.3 Performance evaluation for TCP flows

To further investigate the directional node-disjoint multipath routing schemes proposed in this chapter, we evaluate scenarios with one or multiple TCP flows in the network. To compare with the single path routing studied in Chapter 5, we use the same topologies as in Chapter 5. In each topology, the TCP flows, where the ith flow is from node $i-1$ to node $50-i$, join the network simultaneously. Different from the single path routing studied in Chapter 5 that the path from the source and the destination is calculated offline and is imported into ns-2 simulator, the directional node-disjoint multipath routing protocols proposed in this chapter discover and select routes dynamically.

Table 6.4 and 6.5 summarize the aggregate throughput for the scenarios with only one TCP flow and 5 TCP flows in the network respectively. The throughput of each flow is measured when it becomes steady after the route discovery stage.

On average, the aggregate throughput of the network using node-disjoint multipath routing protocol outperforms the single-path routing by more than 77% for the single TCP flow scenario and more than 14% for the multiple TCP flows scenario. However, the performance improvement is not guaranteed for all topologies when there are multiple TCP flows in the network. This is because although the intra-flow interference, i.e. the interference generated by traffic of the same flow, is reduced using multiple node-disjoint paths routing; the inter-flow interference, i.e. the interference generated by traffic of other flows. Therefore, the throughput

Table 6.4: Throughput of the TCP flow from node 0 to node 49(Mbps)

Topology	Single Path	Multipath using	
		"Neighbor Sweep"	"Neighbor Unicast"
1	150	268	297
2	153	242	263
3	157	259	251
4	154	281	220
5	142	293	310
Average	151	269	268

Table 6.5: Aggregated throughput (Mbps)

Topology	Single Path	Multipath using	
		"Neighbor Sweep"	"Neighbor Unicast"
1	308	328	395
2	278	326	319
3	285	326	384
4	259	384	336
5	308	293	299
Average	288	367	327

of the single path routing that considers inter-flow interference, e.g. using FP-HOP or HOP-FP algorithm in Chapter 5, may perform better than node-disjointed multipath routing without considering inter-flow interference.

6.6 Conclusion

In this chapter, a dynamic node-disjoint paths routing protocol is proposed to improve the performance of TCP flows. In particular, we develop an online node-disjoint path discovery process to find multiple node-disjoint paths between the source and the destination without the knowledge of the global topology. From the ns-2 evaluation of different RREQ transmitting and forwarding schemes, we find that using the "Neighbor Sweep" or the "Neighbor Unicast" transmitting scheme along with the "Not Longer" forwarding scheme can discover the maximum number of node-disjoint paths in the network. However, new forwarding schemes can be proposed in future to further improve the performance of the discovery process, for example, by reducing the number of RREQ transmitted in the network. We also show that the node-disjoint multipath routing algorithm can increase the aggregate throughput of the network compared to

single-path routing schemes when there are one or multiple flows in the network. However, the performance improvement is not guaranteed because the node-disjoint multipath routing algorithm does not consider inter-flow interference. An enhanced node-disjoint multipath routing algorithm that does consider inter-flow interference is future work.

Chapter 7

Link outages by pedestrian blockage in 60 GHz Networks

7.1 Introduction

In Chapters 5 and 6, we study the throughput of one or more TCP flows in a static network. In particular, nodes in the network are not moving and links between nodes are not broken. In this chapter, we study a more dynamic network in which links between nodes are on and off frequently by introducing pedestrian blockage. As mentioned in Chapter 2, a human that stands in the middle between a transmitter and receiver pair can reduce the received power of a Line-of-Sight (LoS) link by 30dB. Therefore, in most cases, the link between a transmitter and receiver will be broken if one or more people block the signal. Therefore, by introducing pedestrian in the network, links in the network are on and off frequently, which makes the topology of the network change dynamically.

Many papers in the literature have investigated the broken link problem caused by node mobility in Mobile Ad Hoc Networks (MANETS) with omni-directional antennas [12–15]. In particular, many on-demand routing protocols, such as DSR [12] and AODV [13], were proposed to quickly find an alternate path when the current path is broken. Moreover, there were many papers focused on improving the performance of TCP over MANETS [46–48]. The original TCP performs badly when links are broken frequently due to node mobility because TCP is unable to distinguish whether a loss is due to a route failure or to a network congestion. To solve these problems, cross layer messages, e.g Explicit Link Failure Notification, are used to differentiate the reason for a packet loss [46]. Fixed Retransmission Time Out (RTO) is also used to handle frequent route failures [48].

Although outages caused by pedestrian blockage in 60 GHz directional antenna networks has some similarities with outages caused by node mobility in omni-directional antenna networks, there are also some differences. First, in a MANET, when a link breaks, it may never

come back or it may come back after a long period. This outage period is difficult to estimate when nodes in the network move randomly using the Random Waypoint model [17]. In comparison, a link that is broken by the blockage of a pedestrian will recover after a short period, since the pedestrian will move out of the LOS path a few seconds after blocking it. Second, in an omni-directional antenna network, a TCP flow does not typically use a multipath routing scheme since TCP with multpath routing may perform worse than with the single path routing [16]. Instead, MANET uses single path routing with multiple backup paths. In other words, the source and the destination find multiple paths between them but only use one path to transmit packets. An alternate path is used only if they detect the current path is broken. In comparison, a TCP flow in a directional antenna network can split packets into multiple routes to improve the performance in a static network as shown in Chapter 6. Therefore, when one of multiple active routes breaks, a directional antenna network is likely to have different solutions from the MANET.

In this chapter, we first study the characteristics of pedestrian blockage in Section 7.2. We then study the characteristics of route blockages in Section 7.3. Based on these observations, Section 7.4 proposes two solutions for the link blockage problem by introducing a blockage timer for the broken path in Section 7.4.1 and employing the multipath routing scheme in Section 7.4.2. The performance of proposed solutions are evaluated by ns-2 simulations in Section 7.5. In the last section, we conclude this chapter.

7.2 Characteristics of link outages by pedestrian blockage

In this section, we characterize link failures caused by pedestrians moving around the network. In particular, we focus on two properties of a link blockage. First is the probability P_b that a link is blocked, which is the fraction of time that the LoS path is blocked by pedestrians. The second property is the average duration T_b of a link blockage event, which is the time interval that starts when the link is blocked by one or more pedestrians and ends when all pedestrians move away from the LoS path. This parameter is important since it gives an estimate of how long this link should not be used after it is blocked. We first build two analytic models for P_b and T_b, and then validate these models via simulations.

There are many random mobility models [17]. In this chapter, we use the "random direction model with bounce" model since it generates a uniform spatial distribution of pedestrians [17]. In particular, when a pedestrian starts a walk, he first selects the moving direction uniformly from $[0, 2\pi)$ and the walking speed uniformly from $[v_{\min}, v_{\max}]$. He then chooses the duration of the walk uniformly from $[w_{\min}, w_{\max}]$. When the pedestrian finishes the walk, he will pause for a duration uniformly chosen from $[p_{\min}, p_{\max}]$ before starting the next walk. When the pedestrian hits a boundary of the area, he only changes the direction of walk following the reflection of light.

7.2.1 Analytic models for link outages by pedestrian blockage

In Chapter 2, we define the pedestrian blockage model as follows. We assume a pedestrian is represented by a circle with radius r, which represents the projection of a human body onto the xy-plane. When a pedestrian crosses the LoS of a link, the human body attenuates the received signal strength. If a pedestrian stays between the transmitter and receiver in the xy-plane, a blockage occurs. Note that we also assume that when a pedestrian's cross-section covers a node, he will walk under the node and thus will not block any link associated with the node.

Based on the above model, a link between nodes A and B with length d will be blocked if the center of a person's body is in the blockage region as shown in Figure 7.1. The perimeter L and the size S of the blockage region are $2d + 2\pi r$ and $2dr - \pi r^2$ respectfully. Since the probability distribution of a pedestrian's location is uniform, the probability that a pedestrian is in the blockage region and thus blocks the link is S/A, where A is the size of area in which pedestrians are walking. Since the mobility model of each pedestrian is independent and identically distributed, when there are N pedestrians in the area, we can obtain the first model of the probability that the link is blocked

$$P_b = 1 - \left(1 - \frac{S}{A}\right)^N$$

From another perspective, the blockage region of a link acts as a server with infinity capacity, pedestrians arrive in the blockage region, stay for a period and then leave. The server is busy if there is one or more pedestrians are in the blockage region. Therefore, the probability that the server is busy is the same as the probability that the link is blocked. Similarly, the

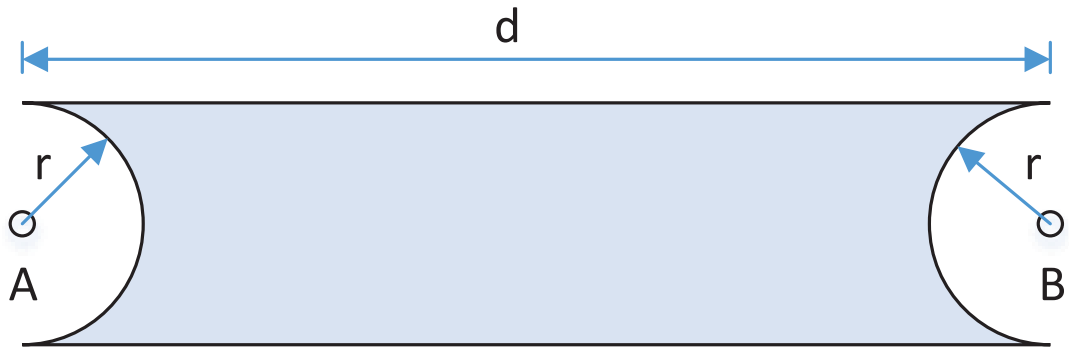


Figure 7.1: The blockage region of a link

average length of a busy interval is the same is the average duration of a link blockage event T_b. The authors in [49] have modeled the average arrival rate $E[M]$ that a pedestrian moves into a region in Equation (7.1) and the average length $E[T]$ that a pedestrian stays in a region in Equation (7.2) given the perimeter length of the region L, the size of a region S, the average moving speed of a pedestrian $E[v]$ and the density of pedestrians ρ that are in motion in the area as follows

$$E[M] = \frac{\rho E[v] L}{\pi} \tag{7.1}$$

$$E[T] = \frac{\pi S}{E[v] L} \tag{7.2}$$

If we assume pedestrians arrive in the blockage region according to a Poisson process with rate λ and stay in the blockage region for a period following an exponential distribution with parameter μ, we have

$$\lambda = \frac{\rho E[v] L}{\pi} = \frac{N E[v](2d + 2\pi r)}{A\pi} \tag{7.3}$$

$$1/\mu = \frac{\pi S}{E[v] L} = \frac{\pi(2dr - \pi r^2)}{E[v](2d + 2\pi r)}. \tag{7.4}$$

From M/M/∞ queueing theory, the probability that a server is busy, which equals the probability that a link is blocked, is

$$P_b = 1 - e^{-\lambda/\mu} = 1 - e^{\frac{-NS}{A}}. \tag{7.5}$$

Comparing Equation (7.2.1) and Equation (7.5), the two proposed link blocking probability models are approximately the same if S is much small than A. In this scenario, both equations could be approximately simplified as

$$P_b \doteq \frac{NS}{A} = \frac{N(2dr - \pi r^2)}{A} \tag{7.6}$$

From Equation (7.6), we can see that the link probability increases linearly as the length of the link and the number of pedestrians in the area increase when the length of the link is short.

From M/M/∞ queueing theory, we can also obtain the average length of a busy interval, which is also the average duration of a link blockage event T_b as

$$T_b = \frac{1}{\lambda}(e^{\lambda/\mu} - 1) = \frac{1}{\lambda}\frac{P_b}{1 - P_b} \tag{7.7}$$

Substituting Equation (7.2.1) and Equation (7.5) into Equation (7.7), we can obtain two models for the average length of a link blockage event

$$T_b = \frac{1 - (1 - \frac{S}{A})^N}{\lambda(1 - \frac{S}{A})^N} \tag{7.8}$$

$$T_b = \frac{1}{\lambda}(e^{\frac{NS}{A}} - 1) = \frac{A\pi(e^{\frac{NS}{A}} - 1)}{NE[v]L} \tag{7.9}$$

When NS is much smaller than A, which indicates λ/μ is close to 0, Equation (7.9) can be approximately simplified as

$$T_b = \frac{\pi S}{E[v]L} = \frac{\pi(2dr - \pi r^2)}{E[v](2d + 2\pi r)}, \tag{7.10}$$

which is same as Equation (7.4). From Equation (7.10), we can see T_b increases inversely when the average moving speed of pedestrians in the area increases. Moveover, T_b is less related with the length of the link when the length of the link increases.

7.2.2 Models Validation

In this subsection, we validate the models proposed in the previous section with the same topology as in the previous chapter. In the simulation topology, 50 nodes (node 0 to node 49) are placed on a 45m$\times$20m grid. The coordinates of node i are $(5 \times \lfloor i/10 \rfloor, 5 \times (i \mod 5))$.

Table 7.1: Three configurations of random direction movement patterns

Config	Short Name	$[v_{min}, v_{max}]$ (m/s)	$[w_{min}, w_{max}]$ (s)	$[p_{min}, p_{max}]$ (s)
1	Higher Speed w/ Pause	[0.1, 1.5]	[2, 6]	[0, 1]
2	Higher Speed w/o Pause	[0.1, 1.5]	[2, 6]	[0, 0]
3	Lower Speed w/o Pause	[0.1, 0.9]	[2, 6]	[0, 0]

We then displace each node by a distance randomly selected from the interval [0,5] meters in a randomly selected direction. We study scenarios in which we vary the moving speed and the number of pedestrians that are moving in a $55\text{m}\times30\text{m}$ area. For each scenario, we randomly generate 10 topologies. For each topology, we run the simulation for 10 rounds. In each round of the simulation, we randomly place all pedestrians in the area at time 0 and they start moving for 100 seconds following the random directional model with bounce strategy.

Three random direction movement configurations with different parameters are listed in Table 7.1. All three configurations have the same distribution for the duration of a walk. However, pedestrians do not pause in configurations 2 and 3. Moreover, pedestrians in Configuration 1 and 2 move at same speed on average, which is faster than pedestrians in configuration 3. Therefore, we name configuration 1 as "Higher speed with pause", configuration 2 as "Higher speed without pause" and configuration 3 as "Lower speed without pause".

We first study a simulation topology in which only one pedestrian moves in the area using movement configuration 3. Figure 7.2 shows the scatter plot for the length of all links in the topology and the link blocking probability of these links. As shown in the figure, the link blocking probability obtained by the simulation fits well with the two analytic models in Equation (7.2.1) and Equation (7.5). Since the length of links in this topology is shorter than 13 meters, the size of a link blockage region is much smaller than the size of the area. Therefore, the two analytic models give almost the same estimate.

Since there is only one pedestrian in this scenario, the blockage frequency of a link is the same as the arrival rate that a pedestrian moves into the blockage region of a link. Figure 7.3 shows the scatter plot for the length of all links in one simulation topology and the arrival rate of pedestrians into the blockage region of a link. Similar to the plot of the analytic model in Equation (7.3), the arrival rate obtained by the simulation increases linearly with the length of the link. However, the arrival rate obtained by the analytic model is larger than the simulation results especially when the link lengths are short. Similarly, in this scenario, the average length

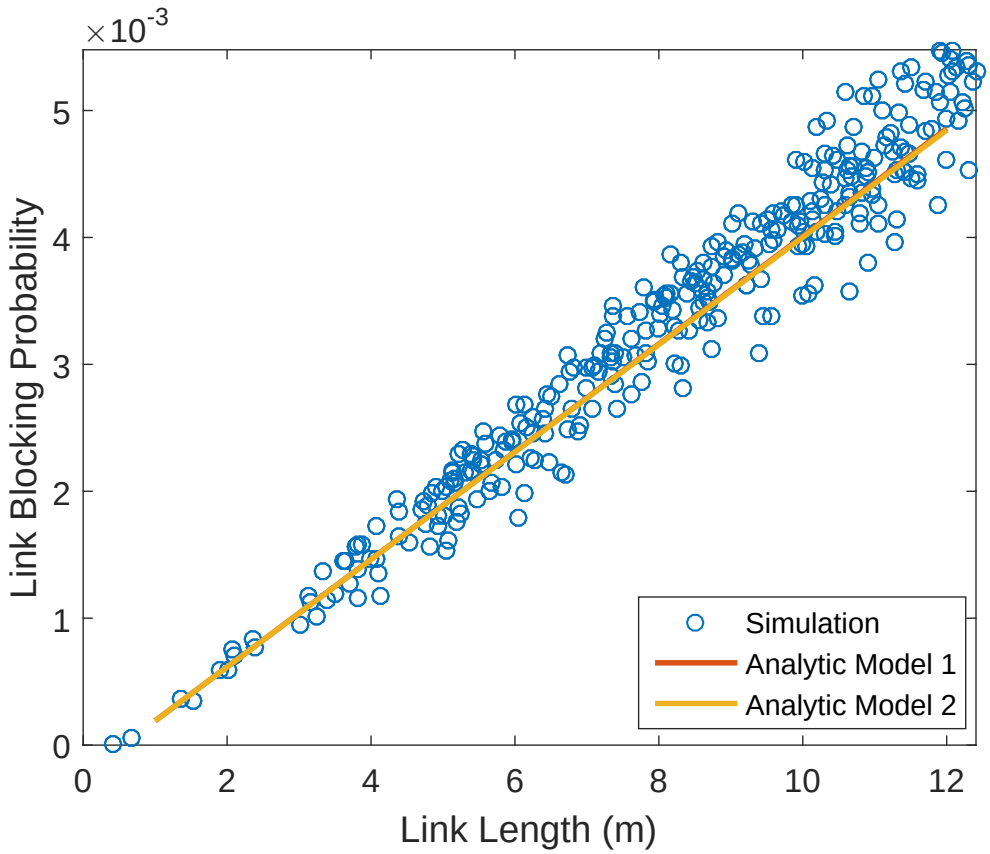


Figure 7.2: Blocking probability that a link is blocked versus the length of the link

of a blockage is the same as the average length that a pedestrian stays in a link blockage region. Figure 7.4 shows the scatter plot for the length of all links in one simulation topology and the average length that a pedestrians stays in the blockage region of a link. T_b obtained by the simulation follows the same trend with the analytic models in Equations (7.8) and (7.9). However, the T_b obtained by the analytic models is smaller than the simulation results especially as the link length becomes shorter. From Equation (7.7), we knows that T_b has an inverse relationship with λ when the link blocking probability is fixed. Therefore, the overestimation of the λ by the analytic model leads to the underestimation of the T_b.

Figure 7.5 and 7.6 show the link blocking probability and average link blockage time versus the length of links when there are 5 and 10 pedestrians. Similar with the scenario in which there is only one pedestrian in motion, the analytic models fit well for the link blocking probability and underestimate the average link blockage time especially when the link length is short.

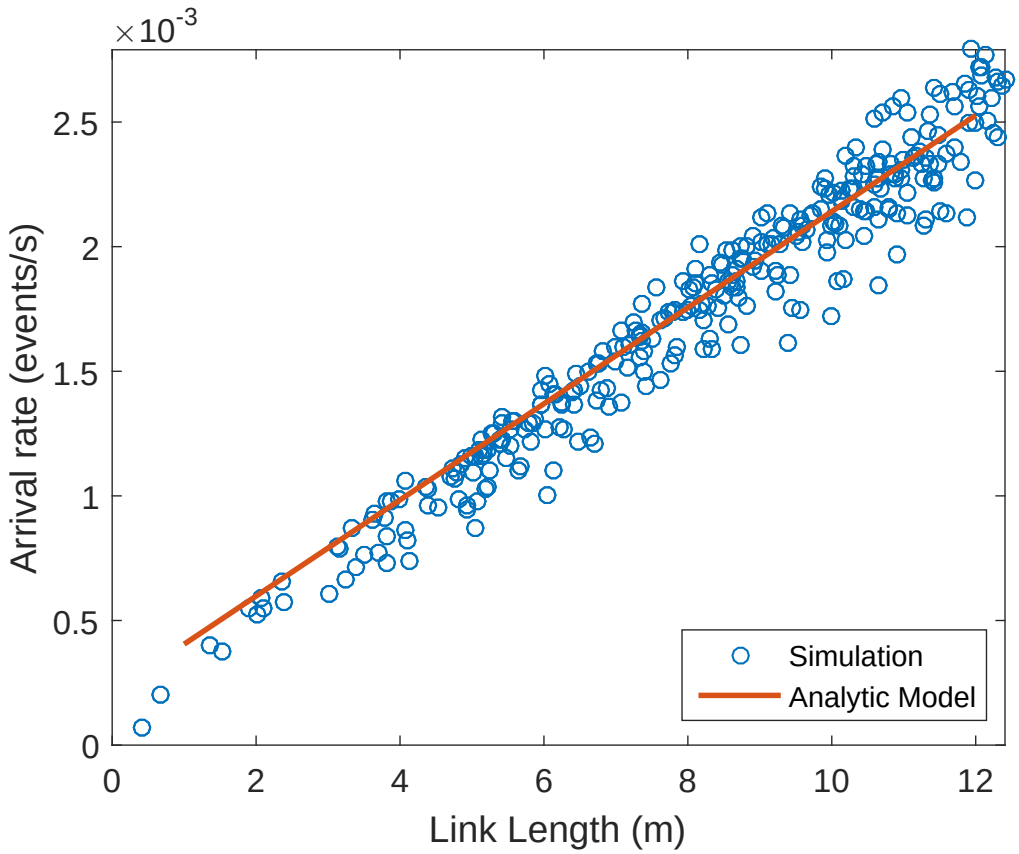


Figure 7.3: Arrival rate that a pedestrian moves into the blockage region of a link versus the length of the link

Figure 7.7 shows the average link blocking probability for all links in the network with a varying number of pedestrians and different moving configurations. Same as the analytic model in Equations (7.2.1) and Equations (7.5), the average link blocking probability increases as the number of pedestrians increases in the network. However, for the same number of pedestrians, the average link blocking probability for all links in the network is not related to the average speed that pedestrians are moving. Therefore, the average link blocking probability for all links in the network is the same for all pedestrian movement configurations.

Figure 7.8 shows the average link blockage time for all links in the network with a varying number of pedestrians and different moving configurations. Similar with the analytic model in Equations (7.8) and Equations (7.9), the average blockage time increases inversely when the average moving speed increases. Moveover, average blockage time increase slightly as the number of pedestrians increases. As expected, the value obtained by the analytic model is

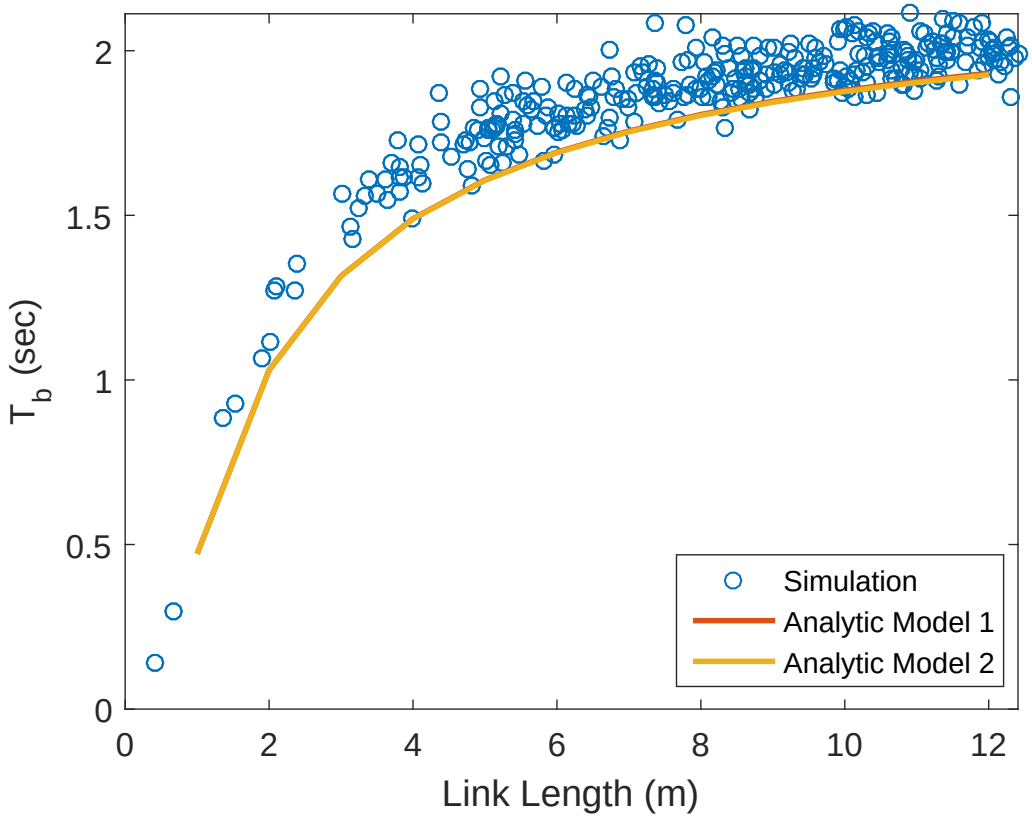


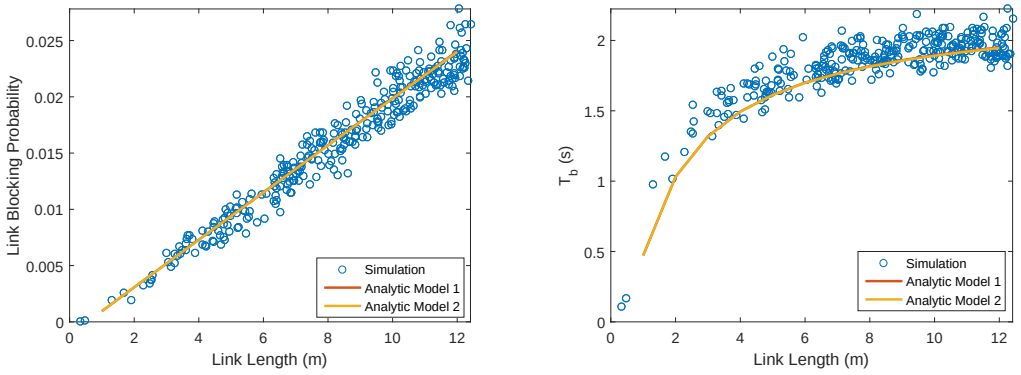
Figure 7.4: Average length that a pedestrian stays in the blockage region of a link versus the length of the link

smaller than the simulation results due to the underestimation as mentioned before.

7.3 Characteristics of route outages by pedestrian blockage

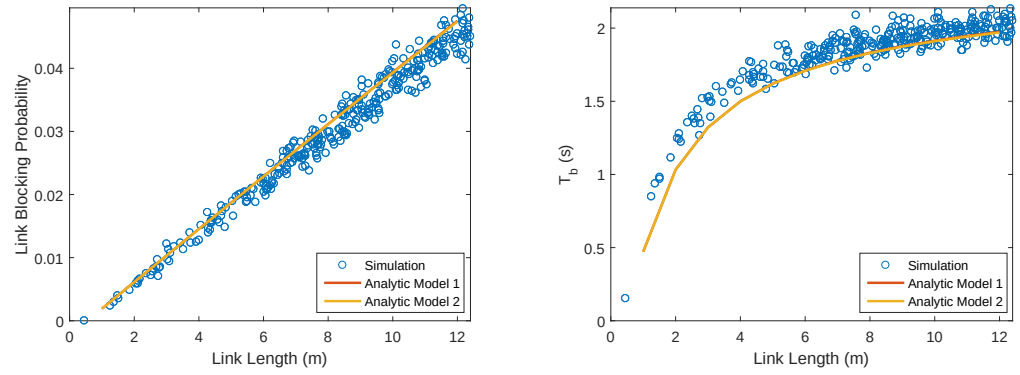
In the previous section, we observed that a longer link is blocked more frequently and for a longer period. In this section, we study the blockage characteristics of a route that consists of several links. A route is considered to be blocked if any of its links is blocked. For a k-hop route $p_k = (v_1, v_2, ..., v_{k+1})$, where v_i is the ith node on the route, we define the length of the route $l(p_k) = \sum_{i=1}^{k} d_i$, where d_i is the Euclidean distance between node v_i and v_{i+1}. In this chapter, for a route with multiple links, we refer to the fraction of time that one or more links are blocked as the route blocking probability of the route.

Figure 7.9 shows the route blocking probability versus the length of the route when 10 and



(a) Blocking probability (b) Average length of a blockage event

Figure 7.5: Link blockage statistics when 5 pedestrians in the area



(a) Blocking probability (b) Average length of a blockage event

Figure 7.6: Link blockage statistics when 10 pedestrians in the area

30 pedestrians with moving configuration 3. Each group of scatter plots contains the results of 10 simulation topologies and for each topology, the first 300 routes that have the minimum hop counts between node 0 and 49 are studied. Same as the analytic model in Equation (7.2.1), the route blocking probability increases when the length of the route becomes longer. However, the route blocking probability increases faster when more pedestrians are in the area. Moreover, the average route blocking probability of all routes in the network is 56.89% when 30 pedestrians are in the area, which is much higher than 24.07% that is the value when 10 pedestrians are in the area.

On average, routes that have fewer hops have shorter length. For example, the average length of routes that have 6 hops is 50.49 meters, which is shorter than 56.50 meters, which is the average length of routes that have 7 hops. Therefore, routes with fewer hops have lower

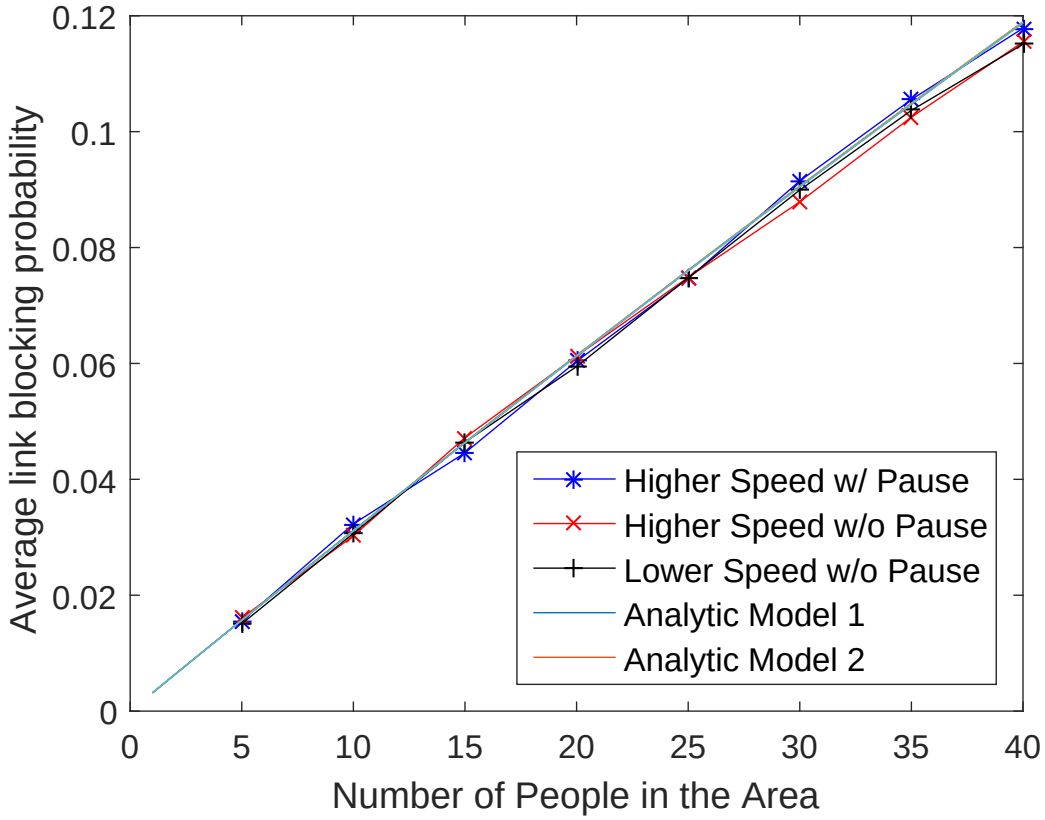


Figure 7.7: Average link blocking probability versus different number of pedestrians

route blocking probability on average. Therefore, the minimum hop algorithm may still be a good candidate to select routes in networks with pedestrian blockage.

7.4 Proposed solutions to mitigate pedestrian blockage problems

Based on the unique properties of link blockage using directional antenna in 60 GHz networks, we now propose two schemes to mitigate the link outages problem introduced by pedestrian blockage. Each scheme can work independently and these two schemes can also be employed together to achieve better performance.

7.4.1 A new route blockage timer

In a traditional MANET, a link is broken usually because the transmitter and the receiver move out of the communication range. Once two nodes move away from each other, the link between

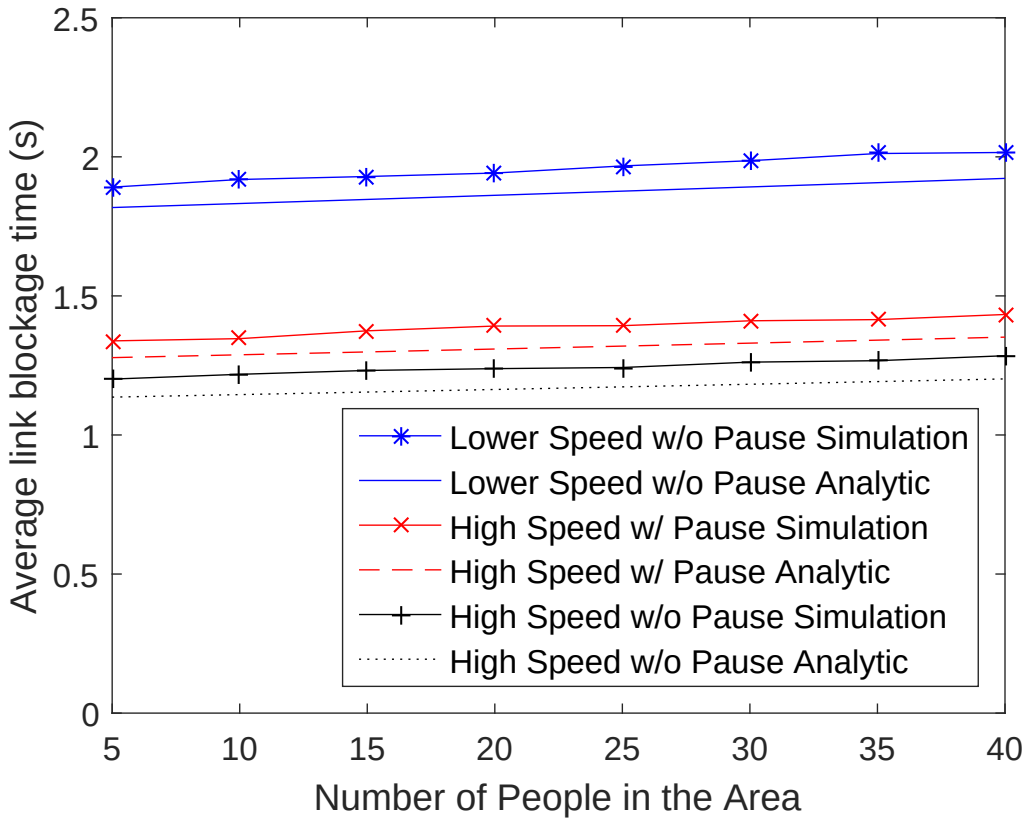


Figure 7.8: Average length of a link blockage T_{Blocked} versus different number of pedestrians

them may not recover for a long period. Therefore, in MANET source routing protocols, after a node detects a broken link, all routes that contain the link will be removed from the route cache [12]. However, in the 60 GHz network as mentioned in the previous section, a link gets broken when a pedestrian moves across the link and recovers after a short period when the pedestrian moves away. Therefore, after a node detects a broken link, all routes that contain the link should not necessarily be deleted as in MANET routing protocol, since the link may recover after a short period. Instead, these routes should be marked as blocked and should not be used temporarily.

To manage the blockage status of a route, in this section, a new route blockage timer is proposed to add to routes that contain a temporarily blocked link. In particular, as in the source routing protocol proposed in Chapter 6, when a node, say node A, notices the link between the next hop node on the route, say node B, is broken after detecting a MAC layer transmission

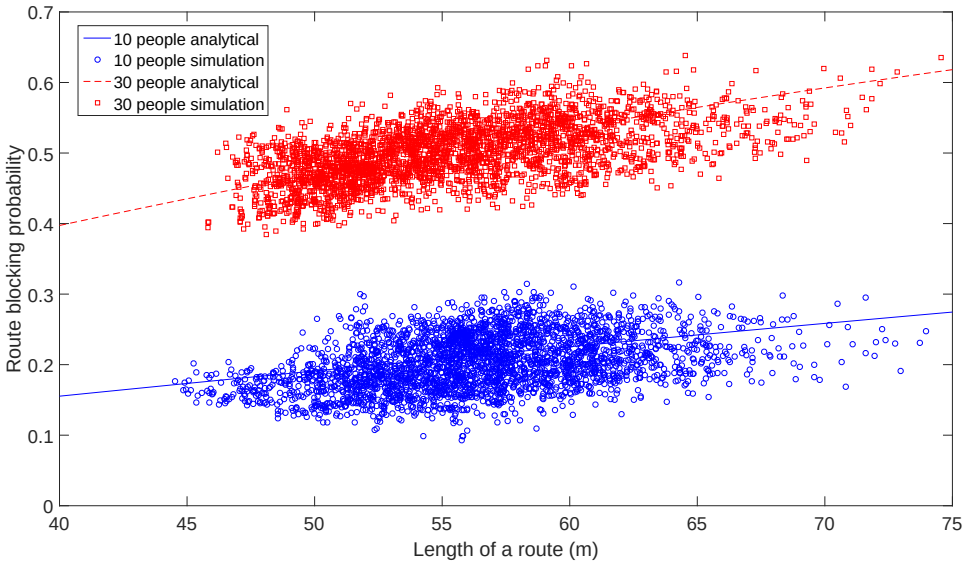


Figure 7.9: Route blocking probability versus the length of the route

failure, node A will send towards the source a unicast dead link notification message, which contains information that link (A, B) is broken. When the source receives the dead link notification message, it will mark all routes in its route cache that contain the link (A, B) as blocked. The source will also start a new blockage timer for each route that contains the broken link. The timer counts for a time interval T_{bt} and the route will be marked as unblocked again after its blockage timer expires. Note that when a route that is already marked as blocked and has a blockage timer counting down when it receives a new dead link notification message telling that one of its links is broken, its blockage timer will be reset to the initial value T_{bt}. The source will only choose a route that is not marked as blocked to send packets to the destination.

By introducing the blockage timer scheme, the number of route discovery processes triggered may be reduced. In the directional source routing schemes proposed in Chapter 6, the source and destination can find multiple paths to each other after a route discovery process. Without using the blockage timer scheme, the source will remove one or more routes in the route cache after receiving a dead link notification message. Thus, after the source receives several dead link notification messages, there will be no route in the route cache. Therefore, the source has to initiate a new route discovery process to find new routes. By introducing the blockage timer scheme, the source does not delete a route that contains a broken link in the

cache and the route can be used again after its blockage timer expires. If there are multiple paths in the cache to the destination, the probability that all routes in the caches are marked as blocked is small. Therefore, the source does not need to initiate a new route discovery process that introduces extra overhead.

However, using the blockage timer scheme may introduce extra dead link notification messages and cause additional delay in the network. For example, for a path (S, A, B, D) from the source S to the destination D, if the source S receives a dead link notification from node A saying that link (A, B) is blocked, the source will mark the route as blocked and will not use it to transmit a packet for a time period T_{bt}. During this time period, there is some possibility that link (B, D) is broken and is still broken when the route is marked as unblocked after its blockage timer expires. In this case, the source is not able to know the route is broken. If the source selects the route to transmit a packet, a new dead link notification will be generated by node B telling the source that link (B, D) is broken. Furthermore, packets sent on this path will have been delayed.

To summarize, the route blockage timer scheme introduces a tradeoff between the number of route discovery processes trigged by removing all blocked routes in the cache and the number of dead notifications trigged by choosing a blocked route to transmit a packet. As studied in the previous chapter, a route discovery process is an RREQ flooding process that introduces more overhead than a unicast dead link notification message. However, if too many dead link notification messages are trigged using the route blockage timer scheme, the benefits of the scheme will be reduced. The performance of the route blockage timer scheme will be studied in Section 7.5.

The performance of the route blockage timer scheme may be affected by two parameters. The first parameter is the length of the blockage timer T_{bt}. If T_{bt} is too small, the link may still be blocked by the pedestrians. In this case, an extra dead link notification message will be generated in the network. On the other hand, if T_{bt} is too big, the difference between using the blockage timer and removing the route from the cache is small. Thus, there may be little or no performance improvement. In this chapter, we propose a heuristic solution to choose a blockage timer that is around the average duration of a link blockage obtained from section 7.2.

The second parameter is the size of the route cache that a node uses. For MANETs, Hu

and Johnson [50] concluded that a small route cache that contains 2 to 3 routes performs better than a bigger cache since routes stored in the cache become obsolete as time goes on. However, this conclusion may not apply to the scenario studied here, because links are broken due to pedestrian blockage not node mobility. In particular, a route found in the route discovery process does not become obsolete as time goes on; it is broken or not broken depending on whether pedestrians occupy links on the route. On the one hand, a bigger cache that contains more routes may perform better since it reduces the probability that all routes in a cache are marked as blocked. On the other hand, if a majority of routes in the route cache are blocked but marked as unblocked, more routes in the cache can trigger more dead link notification messages. Therefore, the size of the route cache need to be tuned in different scenarios in order to improve the performance of the proposed route blockage timer scheme.

7.4.2 Multipath Routing

As we studied in the previous chapter, multi-path routing can improve the throughput of one or multiple TCP flows in a 60 GHz network. However, Chen et. al in [16] showed that TCP performs worse over multi-path routing than over single path routing in a MANET when links in the network are broken frequently due to node mobility. Instead, they proposed to use a backup path when the current path is broken. Therefore, it is not clear whether the node-disjoint multipath routing protocol proposed in the previous chapter can improve the throughput when the network is subject to pedestrian blockage.

Similarly, it has been shown [46] that TCP performs worse when links in the network are broken frequently due to node mobility. For example, if the current path in use is broken, the source may stop sending until a TCP retransmission timeout occurs. However, A TCP retransmission timeout may occur after a long time period since the length of a retransmission timeout (RTO) will be doubled for every retransmission until it reaches the maximum value. Thus, several methods are proposed to improve the TCP performance over MANET. For example, explicit link failure notification (ELFN) was proposed in [47] to inform the sender to freeze its TCP timers since the link is broken. In the chapter, we use the Fixed RTO scheme proposed in [48] to improve the performance of the TCP when a route failure happens. In the RTO scheme, the RTO is not doubled when the packet is retransmitted due to a link failure. In other

words, the RTO remains the same until the source receives the acknowledgment of the packet.

7.5 Performance evaluation

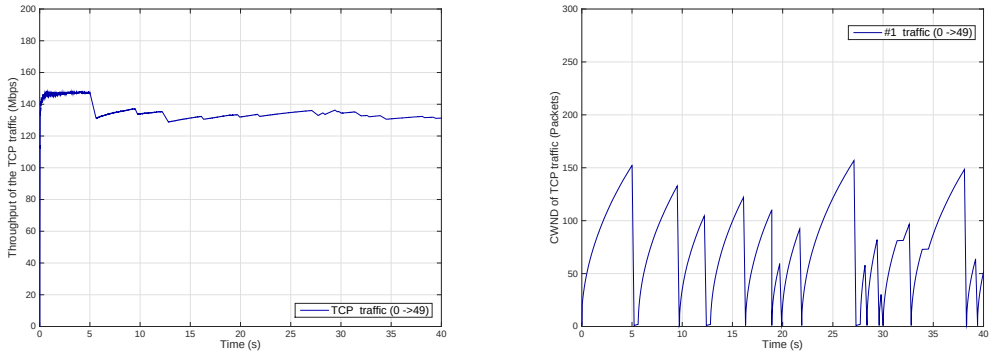
In the previous section we proposed to improve TCP throughput using (1) a route blockage timer and (2) multi-path routing. In this section, we evaluate the performance of these proposed solutions for the pedestrian blockage problem using ns-2 simulator. The same simulation topology is used as in the previous chapter. In the simulation topology, 50 nodes (node 0 to node 49) are placed on a 45m$\times 20$m grid, i.e. the coordinates of node i is $(5 \times \lfloor i/10 \rfloor, 5 \times (i \mod 5))$. We then displace each node by a distance randomly selected from the interval [0,5] meters with a randomly selected direction. A TCP Tahoe flow is generated from node 0 to node 49 at time t=0. In this randomly generated topology, the source is typically about 6 hops away from the destination and there are more than two node-disjoint paths between them. Pedestrians are dropped in the simulation area at time t=5.0 seconds and start moving following the random direction model with the same parameters as Configuration 1 in Table 7.1.

We first evaluate the performance of the proposed solutions when there are fewer pedestrians, e.g. 10 pedestrians, moving in the area in Section 7.5.1. In particular, we first evaluate the performance of the two proposed schemes separately and evaluate the combined solution later. We then extend the study in the scenario that there are more pedestrians, e.g. 30 pedestrians, moving in the area in Section 7.5.2.

7.5.1 Performance evaluation when 10 pedestrians are in the area

In this subsection, we first evaluate the performance of the proposed blockage timer scheme with single path routing and then evaluate the performance of the proposed multipath routing scheme without using the blockage timer scheme. In the end, we study the performance of the solution that combines the two proposed solutions together.

The baseline scheme for all evaluations is the case that neither of the proposed solutions is employed. In particular, in each route discovery process initiated by the source, nodes in the network will use "Neighbor Sweep" RREQ transmission scheme and "First Only" RREQ forwarding scheme to deliver the RREQs generated by the source to the destination as discussed



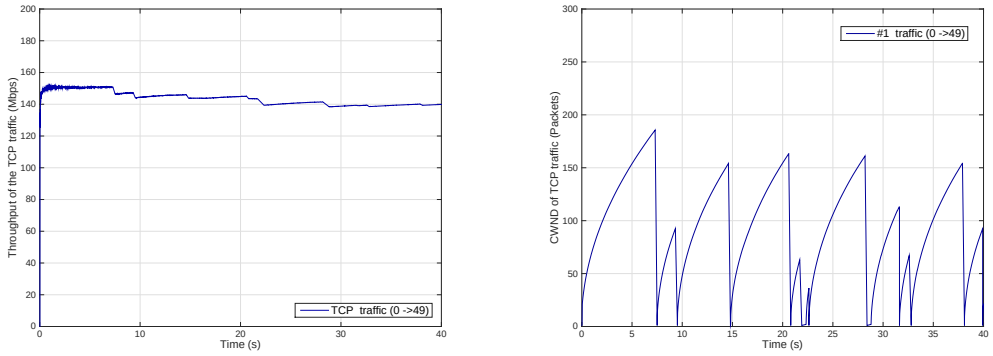
(a) TCP throughput

(b) Congestion Window Size

Figure 7.10: The performance of the baseline scheme when 10 pedestrians are in the area

in Chapter 6. The destination will also reply all RREQs it receives to the source and thus the source can store multiple paths discovered in its route cache as discussed in Chapter 6. After the route discovery process, the source will choose the shortest path among all paths in its route cache to transmit TCP data packets. When receiving a dead link notification message generated by a node on the path, the source will remove all routes that contain the link from the route cache. The source will initiate a new route discovery process when there is no route in its route cache. Figure 7.10 shows the performance of the baseline scheme. In Figure 7.10(a), the throughput stays same before pedestrians are dropped in the simulation area. The throughput then degrades when links are blocked. During the 40 seconds simulation, the source node initiates 11 route discovery processes including the first one at time 0. The link blockages also trigger one fast retransmission process and 16 TCP timeouts, which reset the TCP congestion window to 1 for 17 times in total using TCP Tahoe as shown in Figure 7.10(b) .

A fast retransmission happens when the source receive three duplicate acknowledgements (ACKs) with the same ACK number. The following description shows an example that triggers a fast retransmission event. The source sends the first packet to the destination via a route in the route cache. Before sending the second packet, the source may receives a dead link notification message since the route is blocked. Therefore, the source will send all packets via another unblocked route in the cache and receive three duplicate ACKs that trigger a fast retransmission event via the unblocked route.



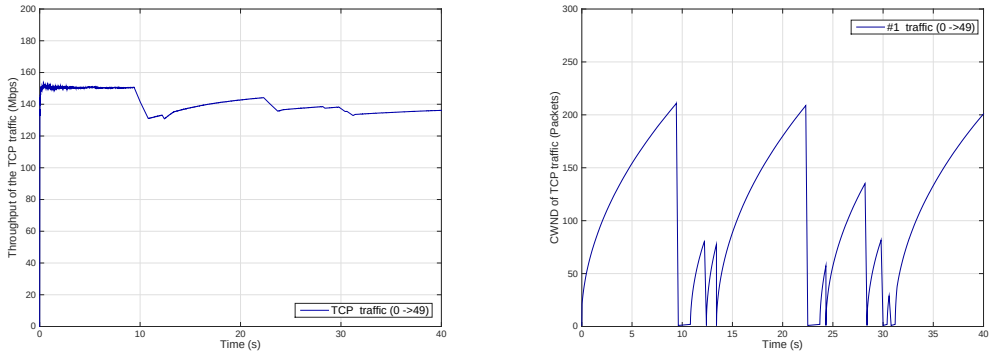
(a) TCP throughput

(b) Congestion Window Size

Figure 7.11: The performance of the scheme with $T_{\text{bt}} = 2$ seconds when 10 pedestrians are in the area

A TCP timeout occurs when a transmitted data packet is not acknowledged before the retransmission timeout expires. If the source sends a packet to the destination when all the routes in the routes in the route cache are broken, the source will not receive an acknowledgment before the retransmission timeout expires.

Figure 7.11 shows the performance of the scheme using the proposed blockage timer solution in Section 7.4.1. The length of the blockage timer T_{bt} is 2 seconds. In other words, a route in the source's route cache will not be used during the next 2 seconds when the node is notified that one of links on the route is blocked. Compared to the baseline scheme in Figure 7.10(a), Figure 7.11(a) shows the throughput is the same before pedestrians are added to the simulation. However, the throughput degrades less compared to the baseline scheme when links are blocked by pedestrians. The aggregate throughput at the end of the simulation is 140 Mbps, which is also slightly higher than the 131 Mbps of the baseline scheme. This is because using the proposed blockage timer scheme, the source node only initiates 4 route discovery processes, compared to 11 route discovery processes in the baseline scheme and each route discovery processes introduces extra overhead into the network. This result also demonstrates that employing a blockage timer can reduce the route discovery rate since nodes do not delete routes that contain a broken link. The proposed blockage timer scheme also reduces the number of slow starts triggered from 17 to 13 as shown in Figure 7.11(b), which includes 3 fast retransmissions and 10 TCP timeouts.

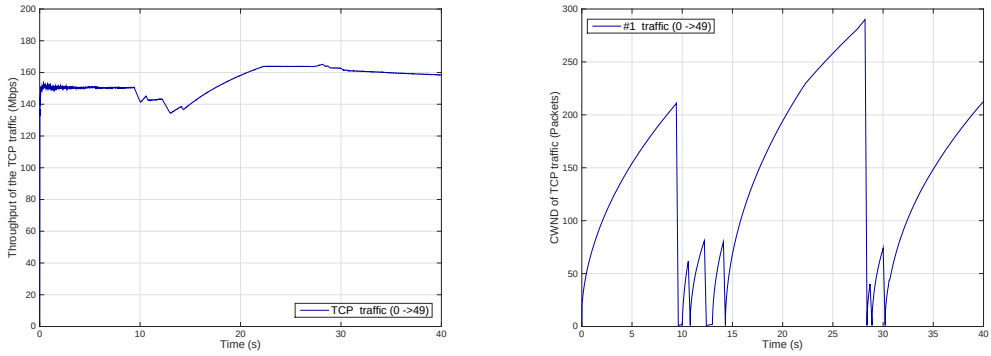


(a) TCP throughput

(b) Congestion Window Size

Figure 7.12: The performance of the scheme with more routes in the cache when 10 pedestrians are in the area

Figure 7.12 shows the performance of the scheme that discovers and stores more routes in a larger route cache. In order to discover more routes, nodes use the "Not longer" RREQ forwarding scheme instead of the "First Only" RREQ forwarding scheme that is used in the baseline scheme as discussed in Chapter 6. As an example in Chapter 6, 121 routes can be discovered using this scheme, compared to 4 routes that can be discovered by the baseline scheme. By storing more routes in the route cache, no route discovery process is triggered after the initial one at time $t = 0$. This indicates that not all the routes discovered are blocked and then removed during the 35 seconds simulation. Comparing with the baseline scheme, discovering more routes in the cache also decreases the number of slow start events triggered from 17 to 12, which includes 2 fast retransmissions and 10 TCP timeouts. However, as shown in Figure 7.12(a), the aggregate throughput does not increase much compared to the baseline scheme. This is because as shown in Figure 7.12(b), when the TCP congestion window is reset to 1 after a TCP timeout happens, it will stay at 1 for a short period, which decreases the aggregated TCP throughput. When a TCP timeout happens, the source will choose another route in the route cache to retransmit the packet. However, some routes in the route cache may be broken at the time of the transmission, and the source does not know if a route in the route cache is broken unless it sends a packet along the route and receives a dead link notification message from a node on the route. If the source chooses a route that is broken at the time of the transmission, it will not start a new retransmission until another timeout occurs. In TCP, the length of the timeout is doubled for each retransmission, with an upper limit of



(a) TCP throughput

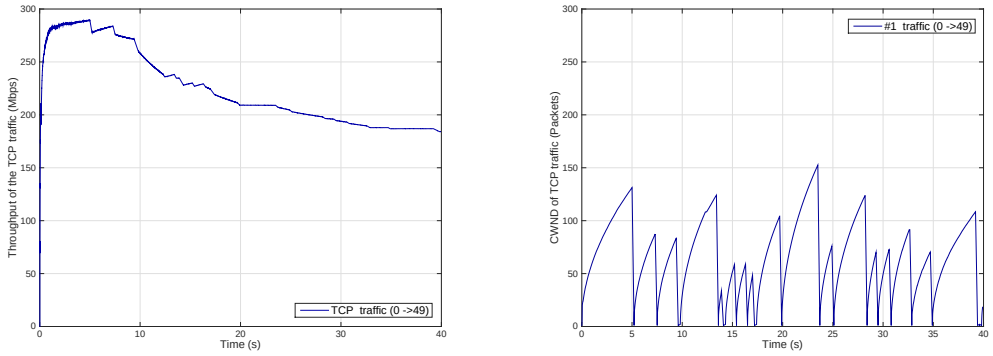
(b) Congestion Window Size

Figure 7.13: The performance of the scheme with more routes in the cache and fixed Retransmission Time Out when 10 pedestrians are in the area

64 seconds [51]. Therefore, the congestion window will stay at 1 for a short period if several consecutive retransmissions happens.

In order to fix this problem, we employ the fixed Retransmission Time Out (RTO) scheme that is used to handle the frequent link failures in MANET as in [48]. In the fixed RTO scheme, the timeout value is not doubled for each retransmission. Figure 7.13 shows the performance of the scheme with more routes in the cache and fixed RTO. By using the fixed RTO scheme, the TCP congestion window stays at 1 for a shorter period as shown in Figure 7.13(b). Therefore, the aggregated TCP throughput increases from 138 Mbps to 158Mbps as shown in Figure 7.13.

As mentioned in Section 7.4.2, multipath routing may be employed to overcome the pedestrian blockage problem and improve the performance of the TCP flow. Figure 7.13 shows the performance of the scheme that employs multipath routing. In this scheme, nodes use the "Neighbor Sweep" RREQ transmission scheme, the "Not longer" RREQ forwarding scheme, the greedy node-disjoint path selection scheme, and the disabled fast retransmission scheme as discussed in Chapter 6. The aggregate throughput decreases after pedestrians are added to the simulation and start moving at time $t = 5$ second, as shown in Figure 7.14(a). Compared to the single path routing algorithm with larger cache in Figure 7.12, three more route discovery processes are triggered by the multipath routing scheme. This is because in the multipath routing scheme, sending packets via multiple routes is able to detect 83 link errors, compared to 36 detected link errors under single path routing. Thus, under multipath routing, routes in the route



(a) TCP throughput

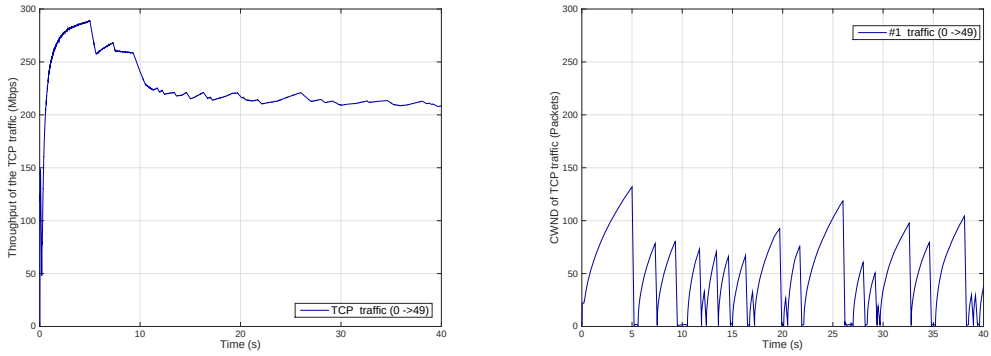
(b) Congestion Window Size

Figure 7.14: The performance of the multipath routing scheme when 10 pedestrians are in the area

cache are removed more quickly and a new route discovery process is triggered once the route cache is empty. However, the aggregate throughput using multipath routing with pedestrians is larger than single path routing with no pedestrian blockage in the network. Therefore, the multipath routing scheme can improve the performance of the TCP flow under the pedestrian blockage due to the high spatial reuse property of directional antennas.

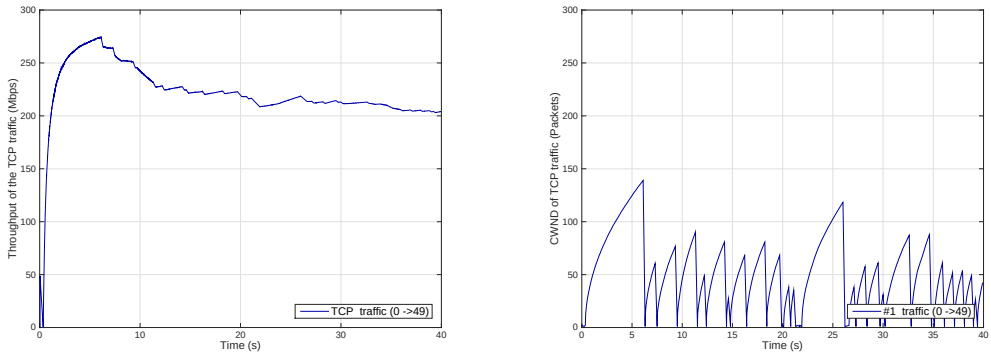
The multipath routing scheme and the blockage timer scheme can be combined together to further improve the performance of TCP under pedestrian blockages. Figure 7.15 and Figure 7.16 show the performance of the combined scheme with $T_{\text{bt}} = 1$ and $T_{\text{bt}} = 2$ seconds respectively. As shown in Figures 7.15(a) and 7.16(a), by choosing the blockage timer duration that is close to the average link blockage time T_b obtained in section 7.2, the combined solution can further increase the aggregate throughput of the TCP flow by 10% over the multipath scheme. This is because by adding the blockage timer, no route discovery processes are triggered after the initial one since the source does not remove a blocked path from its route cache. This reduces the overhead a lot compared to the multipath routing scheme that triggers 3 route discovery processes after the initial one. Although by adding the blockage timer, more than 32 timeouts are trigged instead of 22 timeouts in the multipath scheme, the throughput does not decrease since the fixed RTO scheme is also employed.

In summary, the proposed blockage timer scheme and the multipath routing scheme can



(a) TCP throughput (b) Congestion Window Size

Figure 7.15: The performance of multipath routing scheme with 1 second blockage timer length when 10 pedestrians are in the area



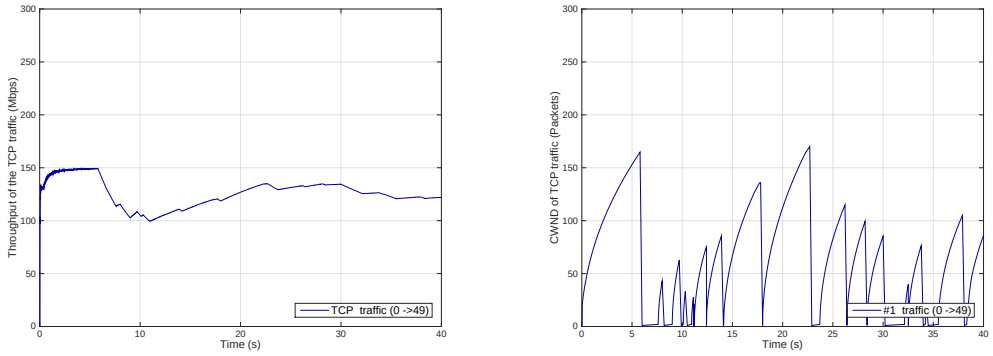
(a) TCP throughput (b) Congestion Window Size

Figure 7.16: The performance of multipath routing scheme with 2 seconds blockage timer length when 10 pedestrians are in the area

improve the throughput of the TCP flow by 7% and 40% respectfully. Using both schemes together can further increase the TCP throughput by 55% comparing with the single path routing without the blockage timer scheme.

7.5.2 Performance evaluation when 30 pedestrians are in the area

In this subsection, we evaluate the performance of the proposed blockage timers and multipath routing scheme when there are more pedestrians are in the area. In particular, 30 pedestrians are dropped at the time 5 seconds and start moving with the same parameters as in the previous subsection.



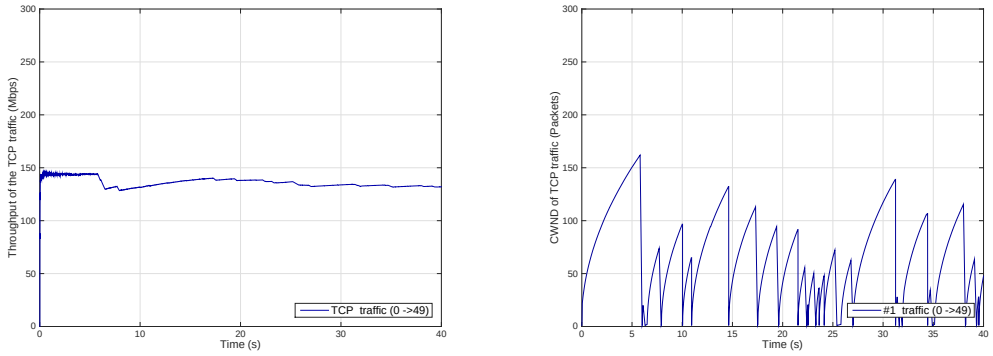
(a) TCP throughput

(b) Congestion Window Size

Figure 7.17: The performance of the scheme with more routes in the cache when 30 pedestrians are in the area

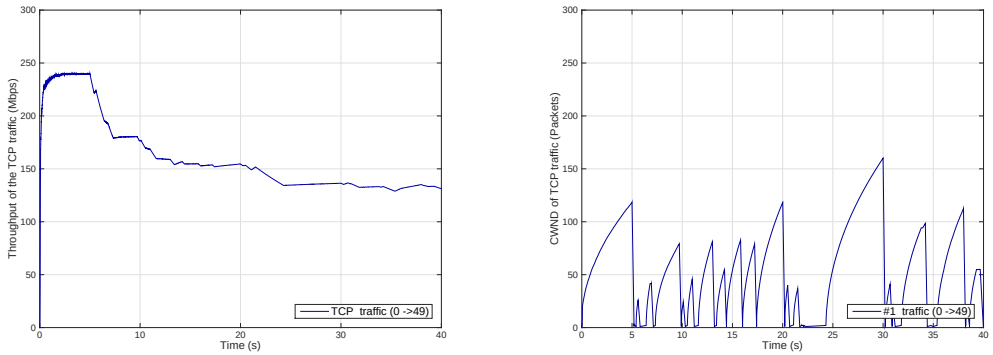
We first study performance of the single path routing with the fixed RTO scheme that discovers and stores more routes in a larger cache, which is the same scheme as studied in Figure 7.17. Compared to the scenario with 10 pedestrians are in the area, the aggregate TCP throughput decreases to 122Mbps as shown in Figure 7.17(a). More link blockages are introduced when more pedestrians are in the area. Thus, the source has to retransmit a packet several times in order to find an unbroken route. As shown in Figure 7.17(b), the congestion window stays close to one for a longer period and 7 more route discovery processes are triggered, which degrades the aggregate TCP throughput.

Based on the observation in Figure 7.17(b) that routes discovered in the route cache may be blocked frequently, a scheme that discovers only one route in each route discovery process is evaluated in Figure 7.18. In particular, in a route discovery process, nodes use the "Neighbor Sweep" RREQ transmitting scheme and "First Only" RREQ forwarding scheme to deliver the RREQs generated by the source to the destination as discussed in Chapter 6. The destination only replies the first RREQ it received, thus, the source can only discover at most one route in a route discovery process. If the only route in the route cache is blocked, a new route discovery process will be triggered. As the result of frequent link blockages, 31 route discovery processes are triggered. Although the TCP congestion window is reset to 1 for 28 times by 10 duplicated acks and 18 timeouts as shown in Figure 7.18(b), the aggregated TCP throughput as shown in Figure 7.18(a) is higher than the scheme that discovers and stores more routes in a larger cache. This is because each route discovery process only takes a short period, e.g. about 30 ms as



(a) TCP throughput (b) Congestion Window Size

Figure 7.18: The performance of the scheme with more routes in the cache when 30 pedestrians are in the area

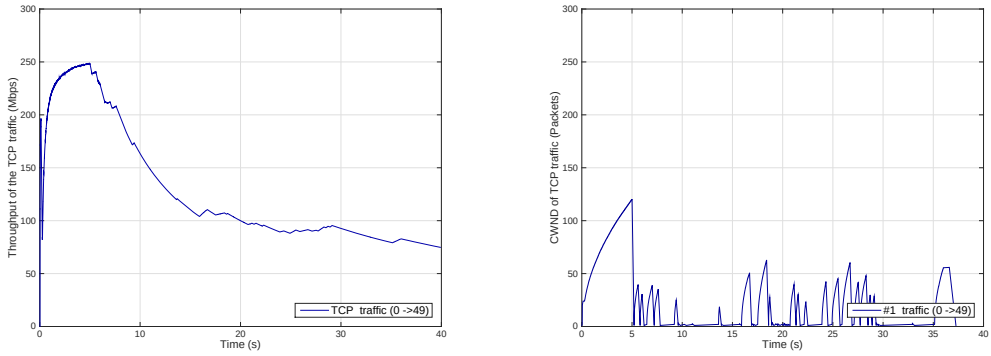


(a) TCP throughput (b) Congestion Window Size

Figure 7.19: The performance of multipath routing scheme when 30 pedestrians are in the area

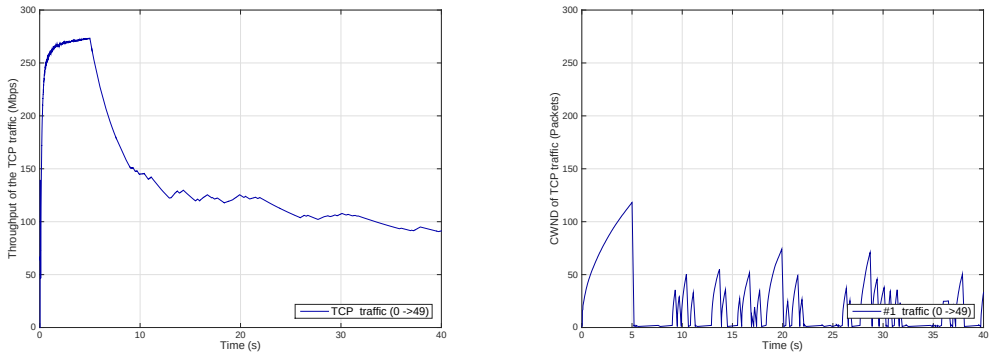
studied in Chapter 6, thus, the TCP congestion window does not stay close to one for a long period.

Figure 7.19 shows the performance of the TCP flow using the same multipath routing scheme as studied in Figure 7.14. Comparing with the scenario in which 10 pedestrians are in the area, the multipath routing scheme does not increase the aggregate TCP throughput much over the single routing scheme when 30 pedestrians are in the area as shown in Figure 7.19(a). This is because 30 pedestrians causes multiple node-disjoint paths to occur infrequently. Since the multipath routing scheme uses a big route cache that stores many discovered routes, as shown in Figure 7.19(b), the TCP congestion window stays close to one for a longer period than the single path routing scheme with a smaller cache.



(a) TCP throughput (b) Congestion Window Size

Figure 7.20: The performance of multipath routing scheme with 1 second blockage timer length when 30 pedestrians are in the area



(a) TCP throughput (b) Congestion Window Size

Figure 7.21: The performance of multipath routing scheme with 2 second blockage timer length when 30 pedestrians are in the area

Different from the scenario with 10 pedestrians, using multiple routes with blockage timers will decrease the performance of TCP as shown in Figures 7.20 and 7.21 when 30 pedestrians are in the area. The blockage timer scheme is employed to keep a temporarily blocked route in the route cache instead of removing it and reduce the number of route discovery process triggered. Since a route in the network is blocked more frequently when 30 pedestrians are moving in the area as studied in Section 7.2, more than 100 timeouts are triggered that keep resetting the TCP congestion window to one as shown in Figure 7.20(b) and 7.21(b), which decreases the aggregated TCP throughput as shown in Figure 7.20(a) and 7.21(a) .

In summary, when 30 pedestrians are in the area, multipath routing does not outperform

single path routing since multiple node-disjoint paths between the source and the destination become rare. Moreover, the scheme with smaller cache size and without blockage timer performs better since links in the network are broken frequently.

7.6 Conclusion

In this chapter, we study link and route outages introduced by pedestrians in a 60 GHz network. Both analytic models and the MATLAB simulation show that link blocking probability increases linearly when the length of the link and the number of pedestrians increase. However, the link blocking probability does not depend on with the moving speed of the pedestrians. Moreover, the average duration of a pedestrian blockage event is less related to the length of the link and the number of pedestrians, especially in scenarios where there are fewer pedestrians and the length of the link is much larger than the diameter of a human body. However, the average duration of a pedestrian blockage event decreases inversely with the moving speed of the pedestrians. Furthermore, these relationships can also be extended to the characteristics of route blockages. Base on the unique properties of link blockages and directional antennas, the blockage timer and the multipath routing schemes are proposed to solve the link blockage problem in 60 GHz network. In the end, we evaluate the performance of the proposed solutions using ns-2 when 10 and 30 pedestrians are in the network. The simulation results show that when there are fewer pedestrians in the network, e.g. 10 pedestrians, the proposed blockage timer scheme and the multipath routing scheme can improve the TCP throughput by 7% and 40% respectfully and using two schemes together can increase the TCP throughput by 55% compared to single path routing using default DSR protocol. However, when there are more pedestrians in the network, e.g. 30 pedestrians, neither scheme performs well due to frequent link blockages and a lack of multiple node-disjoint paths between the source and the destination. In this case, the original directional DSR scheme with smaller cache size performs better.

Chapter 8

Summary and future work

Recent technology advances are poised to enable low-cost, low-power communications in the 7 GHz of unlicensed spectrum at 60 GHz millimeter wave (mmW) frequencies. However, mmW systems that meet the Gb/s data rate demands of wireless multimedia applications must overcome severe propagation effects, including high path loss and high diffraction loss [4]. Consequently, nodes in the network have to use directional antennas to communicate with each other to overcome high propagation losses and achieve high data rates. The narrow main beam widths of directional antennas introduce many design challenges for Medium Access Control (MAC) protocols, and at the same time, this can provide opportunities for routing protocols to improve the capacity of the network by utilizing the benefits of directional antenna such as high gain and high spatial reuse. Although the small wavelength of 60 GHz can help to achieve high directional antenna gain, 60 GHz electromagnetic waves cannot diffract around humans and furniture. These obstacles penalize the 60 GHz link budget by 20-30dB. Therefore, when people are moving, links in 60 GHz networks are on and off frequently due to human body blockage. This introduces design challenges for both routing and transport protocols. In this dissertation, we proposed solutions at the MAC and network layer to address these challenges. In this chapter, we summarize the contributions of the dissertation and present interesting directions for future research.

8.1 Contributions

We made the following contributions in our dissertation:

1. **Medium Access Control:** We proposed an enhanced directional MAC (EDMAC) to resolve the unfairness and low channel utilization issues of deafness in directional MAC protocols for 60 GHz networks. Under EDMAC, packets destined to the same receiver

from different sources use a common optimized contention window size. The experimental results show that EDMAC outperforms basic DMAC especially in multi hop scenarios. Moreover, EDMAC can be used not only in 60 GHz networks, but also for 2.4/5 GHz networks. We also study the performance of TCP traffic in multi-hop networks using EDMAC and DMAC with different queue models, specifically First-In-First-Out (FIFO), Round Robin (RR) and Deficit Round Robin (DRR). Using EDMAC and Round-Robin based queue can improve the fairness between flows in the network and help to estimate the throughput of a flow given its route and network topology.

2. **Routing**: We study the performance of single path routing for networks using EDMAC and find that shortest path routing does not always fully utilize the high spatial reuse properties of directional antennas. We propose two heuristic routing algorithms named HOP-FP and FP-HOP that combine fattest path and minimum hop together with and without the consideration of interference.Ns-2 simulation results show that without the consideration of interference, both HOP-FP and FP-HOP the HOP-FP algorithms are not guaranteed to perform better than shortest path routing for all topologies. However, the HOP-FP algorithm increases the capacity of the network by 27% on average and the FP-HOP algorithm also outperforms the shortest path on average. With the consideration of interference, both algorithms outperforms shortest path on all topologies. The HOP-FP and FP-HOP algorithms increase the capacity of the network by 30% and 40% respectfully.

   We then propose a dynamic node-disjoint paths routing protocol to improve the performance of TCP flows. In particular, we develop an online node-disjoint path discovery process to find multiple node-disjoint paths between the source and the destination without the knowledge of the global topology. From the ns-2 evaluation of various RREQ transmitting and forwarding schemes, we find that using the "Neighbor Sweep" or the "Neighbor Unicast" transmitting scheme along with the "Not Longer" forwarding scheme can discover a large number of node-disjoint paths in the network.

3. **Link outages by pedestrian blockage:** We study link and route blockages when people are moving in a 60 GHz network. Both analytic models and MATLAB simulation

show that link blocking probability increases linearly when the length of the link and the number of people in motion increase. However, the link blocking probability does not depend on with the velocity of the pedestrians. Moreover, the average duration of a human blockage event is less related to the length of the link and the number of people in motion, especially in scenarios where there are fewer people in motion and the length of the link is much larger than the diameter of a human body. However, the average duration of a human blockage events decreases inversely with the pedestrian velocity. Based on the unique properties of link blockages and directional antennas, blockage timer and multipath routing schemes are proposed to mitigate the link blockage problem in 60 GHz network. Simulation results show that when there are few people in the network, e.g. 10 people, the proposed blockage timer scheme and the multipath routing scheme can improve the TCP throughput by 7% and 40% respectfully and using two schemes together can increase the TCP throughput by 55% compared to single path routing using the default DSR protocol.

8.2 Future work

In this dissertation, we proposed new MAC and network protocols to improve the performance of 60 GHz networks. Our work does have some limitations but also points to several new directions for future research. For the proposed EDMAC protocol, our analytical model assumes all packets have equal payload size. Also the proposed protocol is not particularly energy efficient, since a sender still sends beamforming training packets to the receiver even if the receiver is pointing at other senders. A more energy efficient protocol can be proposed in future work. Moreover, our MAC protocol solution is a contention based solution, a contention free solution, e.g. a time division multiplexed (TDM) solution, can be proposed for a static network.

For the routing schemes, we show that the node-disjoint multipath routing algorithm can increase the aggregate throughput of the network compared to single-path routing schemes when there are one or more flows in the network. However, the performance improvement is not guaranteed because the node-disjoint multipath routing algorithm does not consider inter-flow interference. An enhanced node-disjoint multipath routing algorithm that does consider

inter-flow interference can be proposed in future. Moreover, new forwarding schemes can be proposed in future to further improve the performance of the discovery process, for example, by reducing the number of RREQ transmitted in the network.

When we study link and route blockages when people are moving in a 60 GHz network, we assume a uniform spatial distribution of people. For a non-uniform spatial distribution, a learning based protocol may be proposed to estimate the link blocking time and frequency in future. Moreover, we propose two schemes to overcome the human blockage problem in 60 GHz networks. However, when there are more people in the network, e.g. 30 people, neither scheme performs well due to frequent link blockages and a lack of multiple node-disjoint paths between the source and the destination. Therefore, a hop-by-hop transport layer protocol, such as in [52], may perform better than the TCP protocol studied in this dissertation for 60 GHz networks.

References

[1] D. A. Sobel and R. W. Brodersen, "A 1 gb/s mixed-signal baseband analog front-end for a 60 GHz wireless receiver," *IEEE J. of Solid-State Circuits*, pp. 1281–1289, April 2009.

[2] S. et al., "60 GHz single-chip CMOS digital radios and phased array solutions for gaming and connectivity," *IEEE JSAC*, pp. 1347–1357, October 2009.

[3] I. M. C. (IMEC), "Low-cost low-power 60ghz solutions in digital 45nm cmos," *ScienceDaily*.

[4] *IEEE Std 802.11ad-2012 (Amendment to IEEE Std 802.11)*, 2012.

[5] "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.3: Amendment 2: Millimeter-wave-based alternative physical layer extension," *IEEE Std 802.15.3c-2009 (Amendment to IEEE Std 802.15.3-2003)*, pp. 1–200, Oct 2009.

[6] "Wirelesshd specification overview." [Online]. Available: http://www.wirelesshd.org/

[7] "Ecma tc48 - high rate short range wireless communications." [Online]. Available: http://www.ecma-international.org/memento/TC48-M.htm

[8] S. Kato, "Millimeter-wave standardization updates," in *Proc. of Microwave Workshop and Exhibition*, Nov. 2009.

[9] P. Smulders, "Exploiting the 60 GHz band for local wireless multimedia access: Prospects and future directions," *IEEE Commun. Magazine*, pp. 140–147, Jan. 2002.

[10] S. K. Yong and C. C. Chong, "An overview of multigigabit wireless through millimeter wave technology: Potentials and technical challenges," *EURASIP J. on Wireless Commun. and Networking*, pp. 1–10, 2007.

[11] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 1995.

[12] D. B. Johnson, D. A. Maltz, and J. Broch, "Ad hoc networking." Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001, ch. DSR: the dynamic source routing protocol for multihop wireless ad hoc networks, pp. 139–172.

[13] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, Feb 1999, pp. 90–100.

[14] P. J. T. Clausen, "Optimized link state routing protocol (olsr)," 10 2003, rFC 3626.

[15] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, ser. SIGCOMM '94. New York, NY, USA: ACM, 1994, pp. 234–244. [Online]. Available: http://doi.acm.org/10.1145/190314.190336

[16] J. Chen, "Multipath tcp in lossy wireless environment," in *in Proc. IFIP Third Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 04*, 2004.

[17] C. Bettstetter, "Mobility modeling in wireless networks: Categorization, smooth movement, and border effects," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 3, pp. 55–66, Jul. 2001.

[18] C. A. Balanis, *Antenna Theory: Analysis and Design*. Wiley-Interscience; 3 edition, 2005.

[19] *IEEE Std 802.15.3c-2009 (Amendment to IEEE Std 802.15.3)*, 2009.

[20] K. Son, S. Mao, M. Gong, and Y. Li, "On frame-based scheduling for directional mmwave wpans," *Proc. of IEEE INFOCOM*, 2012.

[21] S. Singh, R. Mudumbai, and U. Madhow, "Distributed coordination with deaf neighbors: Efficient medium access for 60 ghz mesh networks," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –9.

[22] T. ElBatt, T. Anderson, and B. Ryu, "Performance evaluation of multiple access protocols for ad hoc networks using directional antennas," in *WCNC*, vol. 2, march 2003, pp. 982 –987 vol.2.

[23] T. Korakis, G. Jakllari, and L. Tassiulas, "A mac protocol for full exploitation of directional antennas in ad-hoc wireless networks," in *MobiHoc*. New York, NY, USA: ACM, 2003, pp. 98–107.

[24] R. Choudhury and N. Vaidya, "Deafness: a MAC problem in ad hoc networks when using directional antennas," in *ICNP*, oct. 2004.

[25] M. Takata, M. Bandai, and T. Watanabe, "A mac protocol with directional antennas for deafness avoidance in ad hoc networks," in *Proc. of IEEE Global Telecommunications Conference*, Nov. 2007.

[26] F. Calì, M. Conti, and E. Gregori, "Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Netw.*, vol. 8, no. 6, pp. 785–799, Dec. 2000.

[27] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535 –547, mar 2000.

[28] L. Kleinrock and F. Tobagi, "Packet switching in radio channels: Part i–carrier sense multiple-access modes and their throughput-delay characteristics," *Communications, IEEE Transactions on*, vol. 23, no. 12, pp. 1400 – 1416, dec 1975.

[29] D. Goodman and A. Saleh, "The near/far effect in local aloha radio communications," *Vehicular Technology, IEEE Transactions on*, vol. 36, no. 1, pp. 19 – 27, feb 1987.

[30] B. Ramamurthi, A. Saleh, and D. Goodman, "Perfect-capture aloha for local radio communications," *Selected Areas in Communications, IEEE Journal on*, vol. 5, no. 5, pp. 806 –814, june 1987.

[31] J. Li, C. Blake, D. S. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, ser. MobiCom '01. New York, NY, USA: ACM, 2001, pp. 61–69.

[32] K. Thompson, G. Miller, and R. Wilder, "Wide-area internet traffic patterns and characteristics," *Network, IEEE*, vol. 11, no. 6, pp. 10–23, 1997.

[33] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on tcp throughput and loss," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, 2003, pp. 1744–1753 vol.3.

[34] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, ser. SIGCOMM '95. New York, NY, USA: ACM, 1995, pp. 231–242.

[35] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. pp. 712–716, 1971.

[36] Z. Chen, R. D. Yates, and D. Raychaudhuri, "EDMAC: An enhanced directional medium access control protocol for 60 ghz networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, Sept 2013, pp. 1726–1730.

[37] S. Mueller, R. Tsang, and D. Ghosal, "Multipath routing in mobile ad hoc networks: Issues and challenges," in *Performance Tools and Applications to Networked Systems*, ser. Lecture Notes in Computer Science, M. Calzarossa and E. Gelenbe, Eds. Springer Berlin Heidelberg, 2004, vol. 2965, pp. 209–234.

[38] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 10, 2001, pp. 3201–3205 vol.10.

[39] R. R. Choudhury and N. H. Vaidya, "Performance of ad hoc routing using directional antennas," *Ad Hoc Networks*, vol. 3, no. 2, pp. 157 – 173, 2005, ad Hoc Networking for Pervasive Systems.

[40] Y. Li, H. Man, J. Yu, and Y.-D. Yao, "Multipath routing in ad hoc networks using directional antennas," in *Advances in Wired and Wireless Communication, 2004 IEEE/Sarnoff Symposium on*, Apr 2004.

[41] H. Gharavi and B. Hu, "Directional antenna for multipath ad hoc routing," in *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, Jan 2009, pp. 1–5.

[42] S. McCanne and S. Floyd, "ns network simulator." [Online]. Available: http://www.isi.edu/nsnam/ns/

[43] K.-C. Leung, V. O. K. Li, and D. Yang, "An overview of packet reordering in transmission control protocol (tcp): Problems, solutions, and challenges," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 522–535, Apr. 2007.

[44] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbor discovery in wireless networks with directional antennas," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, March 2005, pp. 2502–2512 vol. 4.

[45] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.

[46] A. Al Hanbali, E. Altman, and P. Nain, "A survey of tcp over ad hoc networks," vol. 7, no. 3, Third 2005, pp. 22–36.

[47] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ser. MobiCom '99. New York, NY, USA: ACM, 1999, pp. 219–230. [Online]. Available: http://doi.acm.org/10.1145/313451.313540

[48] T. D. Dyer and R. V. Boppana, "A comparison of tcp performance over three routing protocols for mobile ad hoc networks," in *Proceedings of the 2Nd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, ser. MobiHoc '01. New York, NY, USA: ACM, 2001, pp. 56–66. [Online]. Available: http://doi.acm.org/10.1145/501422.501425

[49] H. Xie and D. Goodman, "Mobility models and biased sampling problem," in *Universal Personal Communications, 1993. Personal Communications: Gateway to the 21st Century. Conference Record., 2nd International Conference on*, vol. 2, Oct 1993, pp. 803–807 vol.2.

[50] Y.-C. Hu and D. B. Johnson, "Caching strategies in on-demand routing protocols for wireless ad hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 231–242. [Online]. Available: http://doi.acm.org/10.1145/345910.345952

[51] W. R. Stevens, *TCP/IP Illustrated (Vol. 1): The Protocols*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1993.

[52] M. Li, D. Agrawal, D. Ganesan, and A. Venkataramani, "Block-switched networks: A new paradigm for wireless transport," in *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'09. Berkeley, CA, USA: USENIX Association, 2009, pp. 423–436. [Online]. Available: http://dl.acm.org/citation.cfm?id=1558977.1559006