UNIVERSITY OF CALIFORNIA
RIVERSIDE

MAC Design for Optical Wireless Communications

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Yiyang Li

August 2011

Dissertation Committee:

Dr. Zhengyuan Xu, Chairperson
Dr. Jianlin Liu
Dr. Yingbo Hua

The Dissertation of Yiyang Li is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to sincerely thank my research advisor, Prof. Zhengyuan Xu, for his guidance, patient advising, understanding and continuous support. He guided me through my Ph.D. research by sharing his enthusiasm, knowledge and experience with me. I would also like to heartily express my greatest appreciation to Prof. Srikanth. V. Krishnamurthy, for his continuous guidance, help and support. Without his suggestion, encouragement, patience and conscientiousness, it is impossible for me to complete this thesis. I would like to thank the Department of Electrical Engineering at UC Riverside, especially the members of my dissertation committee, Prof. Yingbo Hua and Prof. Jianlin Liu, for their valuable inputs, discussions and feedbacks on my research.

I would express my appreciation to Jianxia Ning for the joint efforts. She helped me a lot in understanding networking knowledge and OPNET simulator. I would thank all the members in the Wireless Information Technology Laboratory, past and present, from whom I learned a great deal and with whom I shared my happiness. Thanks to Dr. Gang Chen, Dr. Ning Liu, Qunfeng He, Haipeng Ding, Leijie Wang, Kaiyun Cui, Zongyu Dong, Bo Bai, and Li Wang.

I would like to express my heart-felt gratitude to my husband Leijie Wang who is my workmate as well. I have learned a lot from discussions with him on my research. I appreciated his understanding when I got bad moods and his encouragement when I was depressed. His support, encouragement, patience and unwavering love were my most important wealth in the past six years. I thank my parents, Ling Yang and Ximin Li, for their faith in me and unending encouragement and support. Without my family's endless love, this quality research work would not be possible. My dissertation is dedicated to them.

To my family for all the support.

ABSTRACT OF THE DISSERTATION

MAC Design for Optical Wireless Communications

by

Yiyang Li

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, August 2011
Dr. Zhengyuan Xu, Chairperson

This thesis mainly focuses on the higher layer protocol design for optical wireless communication (OWC) networks in two different optical bands: (1) medium access control (MAC) protocol design and neighbor discovery methods for deep ultraviolet (UV) outdoor communications (UVOC), and (2) configuration of indoor visible light communication (VLC) networks.

For UVOC, solar blind and non-line-of-sight (NLOS) operations are attractive. Light beams from UV light-emitting diode (LED) arrays propagate through scattering media, creating spatially different communication links. This unique physical (PHY) layer characteristic was first captured experimentally based on a UV testbed, from which mathematical signal propagation models were developed and their impact on MAC design was realized, i.e., full duplexing and multi-rate transmission. Then we propose a novel contention-based MAC protocol (UVOC-MAC) that inherently accounts for the UV PHY layer and fully exploits multi-fold spatial reuse opportunities. Evaluations via simulation and analysis show that UVOC-MAC effectively mitigates collisions and achieves high throughput. We further develop efficient neighbor discovery protocols by accounting for the varying channel qualities along different scattering directions.

Besides a list of neighbor nodes' identities, a ranked list of node pointing directions in terms of channel qualities was also included in the constructed table to facilitate the process. Utilizing neighbor feedback or alternating a leader node were proved to be able to alleviate the negative effects of random access based collisions and thus expedite neighbor discovery.

VLC by lighting LEDs is gaining popularity, but there is very limited research on the higher layer protocol design. Our extensive channel measurements using a physical layer testbed suggest two effective means to increase data rates, shrinking the beam width and tuning the transmission beam to point towards a target receiver. We design a configuration framework called VICO, by leveraging above PHY features towards achieving the highest throughput while maintaining fairness. VICO tries to schedule transmissions while minimizing conflicts of links. It also opportunistically tunes the idle LEDs to reinforce existing transmissions to increase throughput to the extent possible. Under these proposed treatments, VICO can provide as much as 5-fold increase in throughput as compared to a simple scheduler that does not exploit the possible variations in beamwidth or beam-angle.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Wireless communication is one of the fastest growing industries all over the world. It has revolutionized the way societies function, e.g., it makes broadcasting to the ubiquitous mobile devices possible from satellite, radio station, and cellular base stations. However, with the development of wireless communications, the need for higher data rate in a wireless access network, particularly for wireless multimedia applications, is growing as well. To date, radio technology has been the primary means to offer these services. Many communication organizations are dedicated to increasing communication capacity. Unfortunately, due to the limited bandwidth and increasing traffic, radio spectrum is becoming increasingly congested.

Recently, unlicensed optical wireless communication (OWC) has been considered as an alternative to the existing radio frequency (RF) communications. Optical wireless communications can be, for instance, infrared (IR) communications, visible light (VL) communications and ultraviolet (UV) communications. The whole optical spectrum can be divided into three parts as shown in Fig. 1.1. The ultraviolet cover-

s wavelength range from 100nm to 400nm; visible light is located from wavelength at 400nm to 780nm; infrared ranges from wavelength at 780nm up to 300μm.

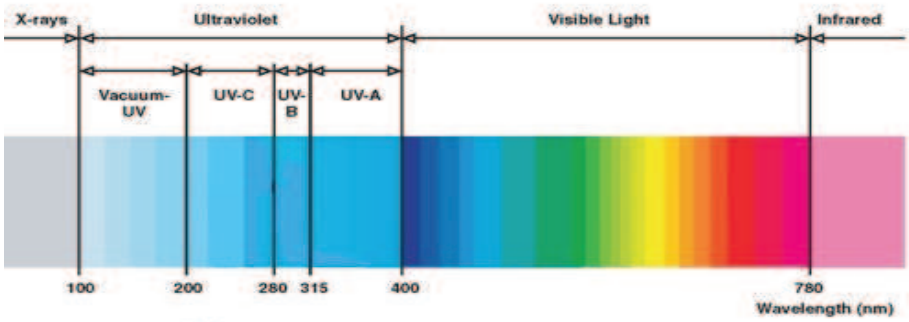

Figure 1.1: Optical spectrum illustration.

Technologies by Infrared Data Association (IrDA), spread widely in applications on notebooks or cell phones. Visible light communication (VLC) promises numerous applications as well. For instance, room lights broadcast alarms, smart-home control messages, or even indoor positioning; billboards transmit advertisements or traffic information; vehicle's head-lights calibrate the distance from its front car. In one word, you are connected with what you have seen and exchange/receive messages. Ultraviolet is another attractive spectrum candidate for both military and commercial applications. Unique interaction of UV signals with the atmosphere makes UV communication a covert and low probability of interception technology.

An optical wireless communication system is composed of a transmitter and a receivers which are physically different and functionally separated. The transmitter modulates information and transfers electronic signal to optical signal with a lighting source, such as light emitting diode (LED). LEDs can be modulated at a relatively high

rate, and thus can serve as a data transmission source. A receiver is a kind of photodiodes. A widely adopted modulation/detection scheme in optical wireless communication is intensity modulation and direct detection (IM/DD). On-off keying (OOK) and pulse-position modulation (PPM) are two typical intensity-based modulation schemes. For example, OOK uses two quantization levels (0 and 1) to represent bit 0 and bit 1, and PPM distributes different pulse positions in a symbol period to represent different signals. The output of a photodetector is proportional to the power of received optical signal. This so-called direct detection cannot capture the phase information of the signal.

1.2 Optical wireless communications

1.2.1 Ultraviolet communications

Given the proliferation of wireless deployments in the industrial, scientific and medical (ISM) bands, the use of LED-based UV communication can be attractive for achieving low cost and high data rate operations. Meanwhile, UV communications typically do not experience interference from commonly used RF systems and equipment. While intense solar energy introduces significant noise to infrared and visible light optical signals, UV-C (the deep UV band operating with wavelength 200-280nm, as shown in Fig. 1.1) waves are immune to solar exposure [4]; this solar blind band makes corresponding UV systems the most attractive from among other bands. In terms of eye and skin safety, UV applications must obey the exposure limits imposed by the International Commission on Non-Ionizing Radiation Protection (ICNIRP) [5] and the International Electrotechnical Commission (IEC). In the deep UV range of the spectrum, the maximum allowable continuous exposure within a one-second time duration occurs at 270 nm at a level as low as 3 mJ/cm^2, increasing to 100 mJ/cm^2 at 200 nm and 3.4 mJ/cm^2

at 280 nm. For a wavelength of 270 nm and a transmitting power of 0.5 mw, it is safe beyond 5cm from an LED.

Another desirable characteristic of UV communications is effective communications via non-line of sight (NLOS) links. When UV waves propagate, there are diverse paths from a transmitter to a receiver because of atmospheric scattering [4],[6],[7]. As long as a certain number of photons sent by a transmitter arrive at a receiver (possibly via NLOS paths), a communication link can be established. A wireless channel established as a NLOS link relies on atmospheric scattering radiation. Therefore, the channel attenuation is severe and the data rate and communication range are limited. Reduction in visible range due to increased aerosol number concentration results in an increase in the number of scatterers in the overlap volume of the transmitter beam cone and receiver field-of-view (FOV) cone. Thus, decreased visible range may result in enhanced UV NLOS communications; this is the opposite of what happens to conventional optical channels. Even in LOS conditions, low background solar radiation and low device dark noise in the UV-C band offer significant advantages. All of the above factors make UV-C attractive for both rural/open and metropolitan/urban outdoor environments. Furthermore, UV-C signals are inherently jamming-resistant and not easy to intercept (low probability of interception) at long distances, due to their unique power decay profile. This makes UV-C especially suitable for tactical applications.

1.2.2 Visible light communications

A VLC system consists of LED arrays (we also refer to these as simply LEDs for brevity) for transmission and photodiodes for detection. It has the following six advantages compared to RF communications: (a) VLC inherently provides increased security against eavesdroppers since it typically requires line-of-sight (LOS) transmissions and

cannot penetrate walls; (b) it is harmless to humans since it operates in the visible light spectrum with much higher tolerable exposure limits; (c) VLC already enjoys ubiquitous infrastructure support. In many cases one can utilize existing lighting infrastructure to build a VLC network; (d) it is cost-efficient because its components (LEDs and photodiodes) can be massively produced and are potentially inexpensive; they can utilize the lighting infrastructure for priority "free" communications; (e) VLC is power-efficient due to sharing of LEDs lighting power; (f) finally, the visible light spectrum is huge and can potentially support very high data rates. In addition, compared to traditional light sources, white LEDs have the advantages of long life expectancy, high lighting efficiency, no out-of-visible-band radiation (unlike incandescent lamps which emit infrared light and fluorescent lamps which emit ultraviolet signals), easy maintenance, and environmental friendliness [3].

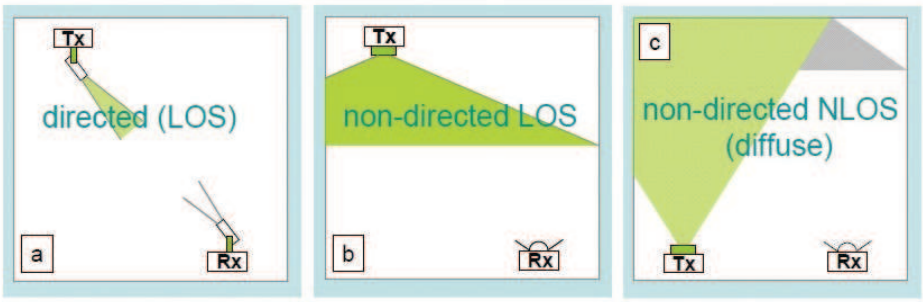

Figure 1.2: Types of communication links: a) Directed line-of-sight (LOS), b) Non-directed LOS and c) non-directed NLOS (diffuse) [1].

This work focuses on an indoor VLC network based on LED lighting systems. Theoretical analysis of an indoor VLC system has been provided in [8] that proves the feasibility of VLC. Indoor VLC links could be classified into line-of-sight (LOS) and non-line-of-sight (NLOS) links (similar to infrared links) [1]. A directed VLC link (Fig. 1.2a) provides LOS communications between a transmitter (Tx) and a receiver (Rx). With

carefully aligned transceivers LOS communications are often possible and can support high transmission rates by focusing the transmission beam and with a narrow receiver field-of-view (FOV). With a non-directed link (Figs. 1.2b and 1.2c), communication between a Tx with a divergent beam and a Rx with a large FOV depends on numerous signal reflections on the surfaces in a room, instead of, or in addition to an LOS link. A link operating in the absence of LOS is defined as a diffuse link. The non-directed link is more robust to shadowing and can inherently support user mobility better. However, multi-path propagation effects may limit the achievable data rate, and may require higher transmission power. In the literature [9],[10], it has been found that the diffuse paths are much weaker than LOS paths. The signal intensity from an LOS path is about 10 times higher than the first reflective path as shown in [9] in a 5m×5m×3m room. Since LOS paths are typically feasible [9], we only consider LOS links in the rest of this thesis.

1.3 Motivations for MAC design

We review some related work in the literature. Note that there has been limited work done on higher layer protocol design for optical wireless communications, particularly in outdoor UV communication (in which we consider ad hoc topology) and indoor VLC (in which we prefer infrastructure topology).

Path loss model for UV communication: For UV communication, there are some prior work on channel modeling including both analytical path loss model [7] and experimental model [6]. Those models capture coplanar NLOS scattering path loss under the motivation of Reilly's common-volume single scattering theory [11]. In a network setting, there is no guarantee of coplanarity of the beam axis and the FOV

axis. The coplanar path loss models presented in [6, 7] require the transmission beam axis and the receiver FOV axis to lie in the same plane. The more recently proposed non-coplanar numerical path loss model in [12] and analytical approximate model in [13] capture the effect of partial alignment. These models are however unable to account for multiple scattering which occurs in practice. In the following chapters, we describe our empirical non-coplanar path loss model developed on the basis of experiments.

MAC design for UV outdoor communication (UVOC): The IEEE 802.11 standard [14] proposes a carrier sense based MAC scheme for indoor IR with LOS communication links. The MAC design in [1] suggests the usage of CSMA/CA in indoor IR ad hoc networks. [15] proposes a new linear contention window selection method, again for IR indoor networks. [16] suggests that TDMA-based MAC protocols could lead to the seamless integration of optical and RF segments. Most recently, a TDMA based MAC is proposed in [17] for an indoor IR infrastructure network.

There is very limited work in the literature on outdoor OWC MAC. The infrastructure-based MACs are not suitable for ad hoc outdoor scenarios or battlefields. Most importantly, the existing approaches are based on carrier sensing; given the modulation strategies and channel asymmetry those are not appropriate for UVOC. To elaborate, it is unclear whether it is appropriate to assume possible full-duplex communications on both directions of a UV link. Since PPM is used, carrier sensing cannot be used to infer whether or not transmissions are ongoing. Finally, transmissions at lower rates may interfere with transmissions at higher rates but not vice versa.

Given the narrow beam angle of wireless optics, the communications are directional. There exist many MAC protocols for use with directional antennas in RF [18, 19, 20, 21, 22, 23, 24, 25]. The natural question that arises is whether these protocols can be directly applied in our considered context (although they are designed for

RF). Due to the nature of UVOC, they are not directly applicable. First, physical carrier sensing is not possible with UVOC. Second, they consider spatial reuse only in directions; here, we consider additional dimensions (pointing angle and data rate). DMAC [18] uses directional RTS/CTS to increase communication range, but suffers from deafness. Tone-DMAC [19] uses out-of-band signaling to solve the deafness problem. Out of band signaling requires the use of a different wavelength in OWC. In optics, switching across different wavelengths requires expensive filters and low-speed mechanical operation; further, it is difficult to implement multiple filters in a small portable device. Protocols such as [20] also use carrier sensing. CRM, CRCM, MDA and DMAC/DA [21, 22, 24, 25] use circular RTS/CTS to avoid the coverage asymmetry problem. A circular RTS/CTS is unnecessary in the context considered. A transmitter can simply turn on all LEDs as shown in all directions at the same time in Fig. 3.1 without causing the asymmetry problem.

It can be observed that none of the existing protocols addresses non-line-of-sight scattering based optical transmission and reception. In this work, our objective is to design suitable MAC protocols and neighbor discovery methods for free space communications in the UV band.

Neighbor discovery schemes for UV communication: There have been a number of efforts on neighbor discovery for RF wireless networks assuming either directional or omni-directional antennas. Vasudevan et al. classify neighbor discovery algorithms into two categories: direct-discovery and gossip-based [26]. They analyze these two possibilities in synchronous and asynchronous scenarios and determine the frequency with which each node should send control packets to maximize the discovery probability within a certain time period. They find that the transmission probability is related to the number of neighbors and transmission beamwidth. The gossip-based

algorithm outperforms the direct-discovery algorithm. Moreover, the performance of the gossip-based algorithm is not sensitive to the density of the nodes. A handshake-based neighbor discovery algorithm is proposed with a time-division-multiple-access (TDMA) based media-access-control (MAC) protocol in 3D space in [27]. With an assumption that each node is equipped with GPS or inertial navigation system (INS), the authors suggest that the transmission power can be reduced to first find neighbors nearby and then increased gradually such that the probability of detection is increased. Luo et al. analyzed neighbor discovery in a code-division-multiple-access (CDMA)-like system and assumed each node is aware of its neighbors' names and signatures [28]. It is shown good performance is achieved by applying multiuser detection techniques; however, the assumptions are unrealistic for an ad hoc network. In [29], methods to save energy in neighbor discovery were researched and a flood-like procedure to achieve neighbor discovery was also suggested. Jakllari et al. proposed an integrated neighbor discovery and MAC protocol for ad hoc networks equipped with directional antennas [30]. The algorithm effectively utilizes directional antennas and accounts for the mobility of the nodes while performing neighbor discovery and maintenance. Vasudevan et al. comprehensively analyzed ALOHA-like neighbor discovery algorithms and proposed a collision detection mechanism to improve the performance in [31]. The algorithm enables each node to know when to terminate the neighbor discovery process without any a priori knowledge of neighbors.

Note that the above algorithms are not directly suitable for a UV ad hoc network. The unique UV scattering channel is fundamentally different from a RF channel. The scenarios are different from both directional and omni-directional antenna cases. The flexibility in both directional and omni-directional transmissions by a UV source (such as a UV LED) provides degrees of freedom in system design and operation that

are not possible with RF. It is possible for a node to know not only the existence of the neighbors but also information about their locations without assistance of GPS or additional devices, contrary to most RF approaches. Thereby, we have to design a neighbor discovery scheme accounting for the unique characteristics of UV communications.

Configuration of indoor VLC networks: As for visible light communications, although the feasibility and advantages of indoor VLC have been previously demonstrated, higher layer protocol design for VLC is very limited. The task group under IEEE 802.15.7 [32] has been working on VLC and has drafted a MAC protocol which inherits most features of the one designed for wireless personal networks (WPANs); however the unique PHY layer characteristics of indoor VLC impact higher layer protocol design. High throughput gains are only feasible if one can account for these physical characteristics better.

In the IEEE standard draft, peer-to-peer, infrastructure to client and broadcast communications are all supported. The draft also considers visibility support across all topologies. Unfortunately, the draft does not account for some physical characteristics of VLC; the VLC properties (as we show) provide opportunities for increasing throughput via appropriate beam tuning. Wu et al. present a new indoor VLC system which achieves satisfactory data rates while supporting multiple access under LOS constraints [33]. The approach however is for peer-to-peer systems: It is unlikely that such communications will be commonplace in an indoor setting, more likely a local area network akin to a WLAN will be employed.

Due to similarities in propagation properties, studies on infrared (IR) are valuable for configuring VLC networks. In [34], the authors show that as long as the receiver FOV is larger than 25° there is very little chance for a communication link to be blocked in an indoor setting. Based on a shadowing analysis versus ceiling height (where lights

are mounted), the authors conclude that the probability of shadowing is low; in particular they show that the likelihood of link shadowing is lower in rooms with larger ceiling heights. Given these results, we assume that shadowing problems can be ignored in the setting considered.

Charoen Tangtrongbenchasil designs an indoor ubiquitous optical free space communication system in [35]. This system is composed of a Tx array board on which each LED emits light through a lens to cover an area (called a cell) on the ground; overlaps across cells can be controlled by an electronic switching matrix hardware. The hardware also contains a co-located Rx array which can appropriately collect received light. Therefore, highly accurate and well controlled directionality can be achieved. This system is able to obtain knowledge of user devices' locations so that a coordinator can stimulate LEDs to transmit signals to desired receiver devices. In this thesis, we adopt a similar configuration for our system as depicted in Fig. 1.3. In order to cover the entire room, we enforce overlaps between each pair of adjacent cells in the default setting. Intuitively, the users located in such overlapped areas have a high possibility of being interfered with. Therefore, when we allocate resources, the interference effect should be carefully taken into account.

1.4 Contributions and organization of the thesis

Based on the PHY layer properties assessed by our experiments, we propose UVOC-MAC, a MAC protocol that is tightly dependent on UVOC PHY properties. UVOC-MAC fully exploits multi-fold spatial reuse opportunities with appropriate configurations.

We design our neighbor discovery mechanisms. First, we consider a well-

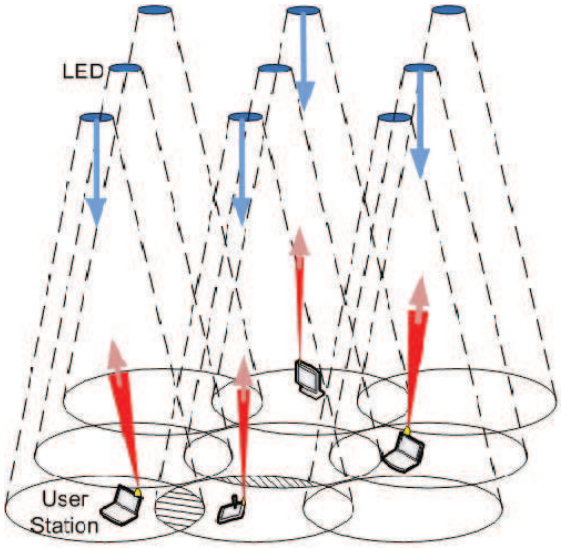

Figure 1.3: An indoor VLC Network: LED arrays are mounted on the ceiling. There are 9 cells in the realization. The clients are all at desk-top height.

planned deployment, wherein nodes are assigned a fixed set of synchronized directions in which they can point their beams. In other words, the sense of direction is global; when a node receives a message from a neighbor, it can immediately determine the direction in which that neighbor transmitted its packet. We refer to this as neighbor discovery with direction synchronization. Second, we propose a more general neighbor discovery scheme, wherein nodes cannot determine a neighbor's direction by simply receiving the latter's transmission. The approach incorporates a novel handshake mechanism that is tightly dependent on the UV propagation characteristics. However, since this approach uses random access as the basis for neighbor discovery, we propose a much more efficient protocol, that can significantly speed up the neighbor discovery process.

For VLC, we first perform extensive channel measurements and verifications in the visible light band in an indoor scenario on a PHY layer testbed. Based on the experiments, we make the following observations: in order to increase data rate **(1)** the transmission beamwidth can be shrunk, and/or **(2)** the transmission beam can be tuned to point to target area. We further simulate the illumination variations using a Matlab

simulation tool. Subsequently we validate that the human eye is able to accommodate the changes in intensity brought about by shrinking the beam or by tuning the beam direction.

After verifying the feasibility of the two degrees of freedom (variable beamwidth and orientation of a transmitting LED), we develop a novel configuration framework (VICO for VIsible light COnfiguration framework) for an indoor VLC network. VICO schedules transmissions, and appropriately chooses the beamwidth and LED orientation, while taking into account the indoor VLC PHY characteristics.

More specifically, we make the following contributions:

- **Experimentation on a UV testbed:** We perform extensive experiments on an outdoor UV PHY testbed. We gain insights into physical aspects that impact the MAC design. First, we find that due to NLOS links there can be multiple possible links (in different configurations) between a communicating pair. Second, different configurations result in different delay spreads, which in turn determine the achievable data rates. Third, full-duplex communications are feasible in some specific configurations; if appropriately invoked this can in essence double the capacity of the link. (See Chapter 2)

- **Design of UVOC-MAC:** We design UVOC-MAC to account for the above factors. In a nutshell UVOC MAC effectively exploits NLOS communication links and full-duplex possibilities. It also mitigates some of the side-effects that arise due to these functions (discussed later). It also alleviates problems such as deafness and hidden/exposed node problems, commonly encountered in directional RF communications. We provide a theoretical analysis and also perform simulations to validate the efficacy of designed UVOC-MAC in delivering high throughput

and yielding low collision. The collision probability is decreased by about 50% as compared to a MAC that does not take the UV PHY properties into account. In some cases, the throughput can increase by up to 4 times compared to a protocol that is agnostic of the UV-PHY. (See Chapter 3)

- **Neighbor discovery for UVOC:** A receiver can immediately determine the direction using which a transmitter sent its packets if direction is synchronized (introduced later). Based on this, it can infer the correct direction in which it should send a response back to the transmitter. Since we do not assume any multiuser detection technique, it is impossible to distinguish the desired signal from interference or background noise. We adopt a credit collection method to solve this problem. We design a mechanism for neighbor discovery given those features and analyze the process via extensive simulations. In most cases where the time to set up a network is limited or the cost is prohibitive, direction synchronization may be infeasible. We design the second neighbor discovery algorithm for applicability in a network without restrictions on direction synchronization. Our approach includes a handshaking mechanism that is tightly integrated with UV properties. In a nutshell, each node sends a request containing information about itself and receives feedback from its neighbors. It then counts *collected credits* for each direction and for each neighbor. It uses these credits to create a table in which, the directions of communication with each neighbor are ranked. (See Chapter 4)

- **A more efficient scheme for neighbor discovery:** We design and evaluate an efficient neighbor discovery protocol that (a) allows the fast and efficient discovery of neighbors, (b) can be applied without any prior knowledge of the network topology or configuration, and (c) allow each node to determine the various direc-

tions that can be used to communicate with each of its neighbors and *rank* these directions in terms of their effectiveness. (See Chapter 5)

- **VICO: A framework for configuring indoor visible light communication networks:** Our measurements suggest that in order to increase data rates with VLC (1) the beam width of a communicating link can be shrunk, and (2) the transmission beam can be tuned to point towards the target recipient. We then perform Matlab simulations to verify that the human eye is able to accommodate the changes brought by shrinking a beam or by tuning the beam direction. We design a novel framework VICO, for configuring an indoor VLC network. Specifically, the goal of VICO is to schedule transmissions and tune beamwidths and orientations to achieve the highest throughput under conditions of max-min fairness. We show that this problem, unfortunately is NP-hard. Thus, VICO uses a very effective heuristic to schedule transmissions. It also tunes the beamwidths and orientations to the extent possible to opportunistically increase the achieved throughput. We evaluate VICO via extensive simulations using the OPNET simulator. The PHY layer characteristics are integrated to accurately characterize the VLC links. Our simulations demonstrate that VICO provides as much as a 5-fold increase in throughput compared to a scheduler that does not opportunistically exploit VLC characteristics (tuning of beamwidth and beam-angle). (See Chapter 6)

Chapter 2

UV physical layer characteristics

Towards our goal, we first perform extensive experiments in UV-C band; we make the following observations: (1) NLOS links are likely to exist with various configurations[1]. (2) A transceiver can work in a full-duplex mode with certain configurations. (3) The channel delay spread is different with different configurations; thus the possible data rates will also differ. From the experimental data, we also derive an empirical path loss model for NLOS communications to capture the extent of signal attenuation [36].

In the following, we discuss three physical characteristics of UVOC that impact MAC design. They are (1) the existence of NLOS communication paths, (2) the impact of delay spread on the achievable data rates, and (3) the ability to establish full-duplex communications. In addition, we also describe the empirical non-coplanar path loss model derived from experiments for UV communications.

[1] A configuration refers to a combination of the transmission direction and pointing angle of the transceiver. We define these terms formally later.

2.1 Experimental setup

The testbed used for experiments is shown in Fig. 2.1. In our experiments, we employ LEDs, a solar blind filter, and a photon detector that operates at 259 nm wavelength. Our experiments are conducted in an outdoor open area. The transmitter uses a signal generator to feed a current driver circuit that powers an array of 2 ball-lens UV LEDs. The radiated optical power is 0.4 mW in total. The beam divergence angle is measured to be 17°. At the receiver, a solar blind filter is mounted on top of the circular sensing window of a Perkin-Elmer photomultiplier tube (PMT) module MP1922. The PMT output current is fed to a high speed preamplifier, whose output is further sent to a photon counter for photon detection. The transmitter also has a Q-switched solid state UV laser which can be used as necessary instead of the LEDs. The transceivers employ energy-based modulation (e.g., pulse position modulation - PPM, or on-off keying - OOK). At the receiver an energy detector is employed.

Figure 2.1: The UV testbed (Left cart: photodetector and oscilloscope; right cart: LED/laser transmitter).

2.2 NLOS scattering communication channel

UVOC adheres to the commonly used signal-to-interference-plus-noise-ratio (SINR) model. In order to compute the SINR, one needs to characterize the attenuation that a UV signal experiences. Prior work on UV [4, 6, 7] models coplanar NLOS scattering path loss under the motivation of Reilly's common-volume single scattering theory [11]. From Reilly's theory, if there is a common volume (the overlap volume shown in Fig. 2.2) between the transmission beam and the receiver's FOV, the transmitted signal could be potentially detected. Thus, the transmitter and the receiver can successfully communicate not only via the LOS path but also via *NLOS paths*. NLOS links allow a node to have the choice of more than one configuration to communicate with a neighbor node. However, the increase in communication opportunities also results in increased interference in a transmitter's neighborhood. We analytically assess the tradeoffs in Section 3.2.

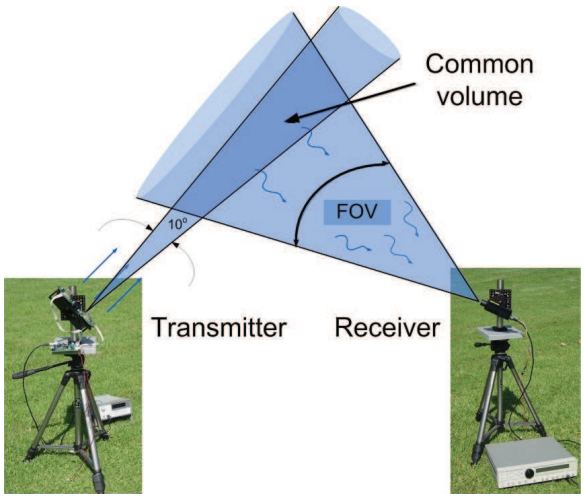

Figure 2.2: Common (overlap) volume between transmission beam and receiving field of view (FOV) to form a NLOS link.

To date, there is no easy-to-use path loss model for non-coplanar geometries that captures the multiple scattering effect. In a network setting, there is no guarantee

Table 2.1: Experimental settings.

The filter transmission efficiency	0.1
The PMT detection efficiency	0.2
LED power	0.2mW
Wavelength	259nm
Noise	16 photons/s
Mismatch	0.05
Beam angle	17 °
Field of view (FOV)	30 °
Receiving pointing angle	90 °

of coplanarity of the beam axis and the FOV axis. The coplanar path loss models presented in [6, 7] require the transmission beam axis and the receiver FOV axis to lie in the same plane. The more recently proposed non-coplanar numerical path loss model in [12] and analytical approximate model in [13] capture the effect of partial alignment. Those models are however unable to account for multiple scattering which occurs in practice. In the following, we describe our empirical non-coplanar path loss model developed on the basis of experiments.

2.2.1 Non-coplanarity and path loss

First we measure the non-coplanar impact on path loss. The transmitter and receiver baseline separation is up to $50m$. Using a curve-fitting method, a path loss model which depends on transmission pointing angle, off-axis angle and baseline separation is developed. The parameter settings are listed in Table 2.1. The values of filter

transmission efficiency and PMT detection efficiency are hardware parameters that affect the received energy. The LED power refers to that of a single LED; we use two LEDs (bound together) in all of our experiments. Given the LED power and actual receiving energy per counting interval, we can derive the path loss from their ratio. The background noise and device dark noise are negligible during experiment.

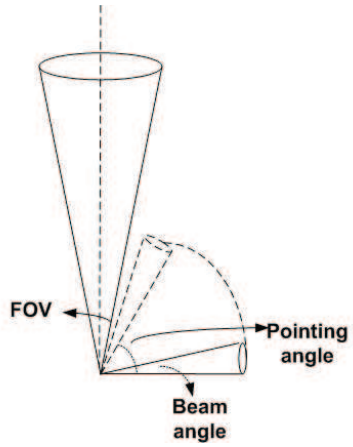

Figure 2.3: Illustration of FOV, beam angle and pointing angle.

The beam angle defines the transmission full beamwidth [7]. Together with the receiver's FOV, it is shown in Fig. 2.3. We fix their values as in Table 2.1. *Off-axis and pointing angles:* The off-axis angle φ is depicted in Fig. 2.4 and derived in

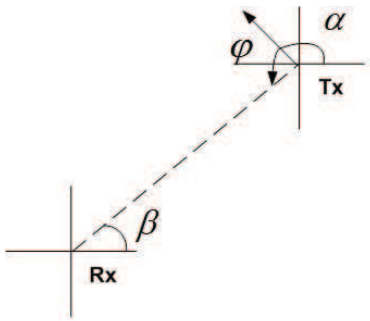

Figure 2.4: Illustration of off-axis angle φ from atop.

Eq. (2.1). It specifies the horizontal deviation from the coplanar axis of two nodes. The vertical deviation from the coplanar axis is referred to as the *pointing angle* [12]. The transmitting angle $\alpha = \frac{\pi}{K} + n\frac{2\pi}{K}$ $(n = 0, 1, 2, ..., K-1)$; K is the total number of directions and in Fig. 2.4, $K = 4$. With fixed transmitter and receiver coordinates, angle β is determined (see Fig. 2.4).

$$\begin{aligned} \gamma &= (\pi - \alpha + \beta)\ mod\ 2\pi \\ \varphi &= \begin{cases} \gamma & 0 < \gamma < \pi \\ 2\pi - \gamma & \pi < \gamma < 2\pi \end{cases} . \end{aligned} \tag{2.1}$$

2.2.2 Non-coplanar path loss model

We leverage the following coplanar UV path loss model developed in [6]

$$L_{coplanar} = \frac{p_t}{p_r} = \xi r^a, \tag{2.2}$$

where L denotes the path loss, p_t is the transmission power, p_r is the received power at the detector, r is the horizontal separation distance of the transmitter and receiver, ξ is the path loss factor and a is the path loss exponent. ξ and a are functions of the transmitter and receiver pointing angles [6]. If the off-axis angle between the transmitter and the receiver is non-zero, the received power decreases and the aforementioned coplanar UV path loss model is no longer valid. Our experimental results (see Fig. 2.5) suggest that the coplanar path loss model can be extended to accommodate such cases. We collect photon counts every second for 20 times at every 2° off-axis angle increment until the signal is overwhelmed by the noise. Using the process of curve-fitting on the experimental raw data in Fig. 2.5, we see that the path loss increases almost exponentially with increasing off-axis angles and can be described by the following model:

$$L_{non-coplanar} = \xi r^a \exp(b\varphi), \tag{2.3}$$

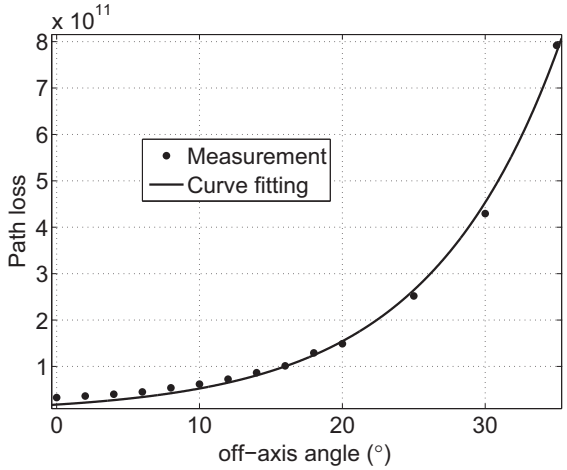

Figure 2.5: Exponential curve fitting at scenario of pointing angle = 15 °, and distance = 15m.

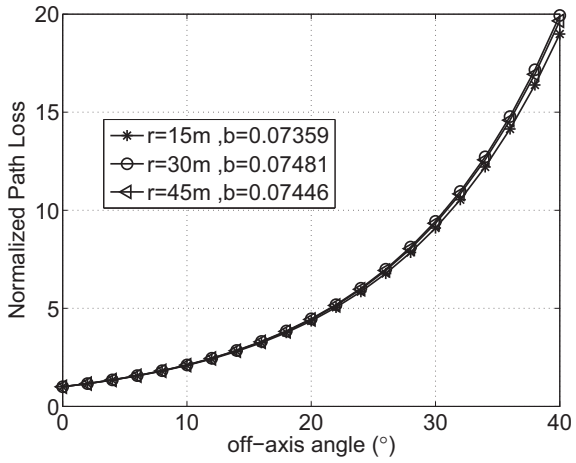

Figure 2.6: The impact of distance on the exponent factor b (pointing angle = 30°).

where the off-axis angle φ is given by Eq. (2.1). The exponent factor b is not (significantly) affected by varying distance r. Experiments show that b is 0.07359, 0.07481, 0.07446 for distances of 15m, 30m and 45m, respectively (see Fig. 2.6). But we observe that b increases as the pointing angle decreases in Fig. 2.7. This is not surprising due to the common volume theory if single scattering is dominant; the overlapped volume shrinks rapidly with increasing off-axis angle at lower pointing angles. The path loss caused by off-axis angle increases faster when the pointing angle is smaller. To ensure SNR above

certain threshold, the maximum tolerable off-axis angle varies with the pointing angle; it increases with increased pointing angle, and takes values of $5°, 7°, 15°$ and $20°$ with pointing angles of $10°, 30°, 40°$ and $60°$, respectively.

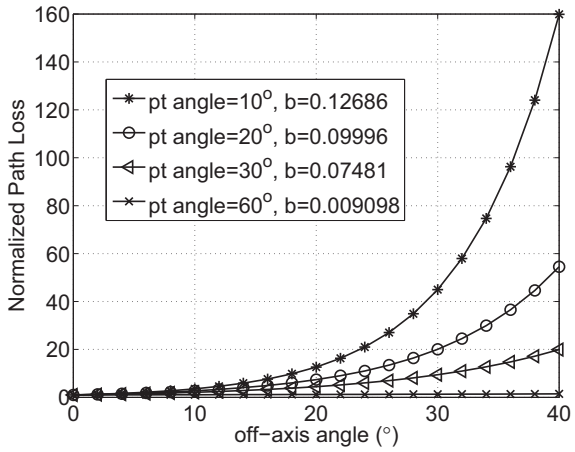

Figure 2.7: The impact of the pointing angle on the exponent factor b (distance $= 30m$).

2.2.3 Spatial reuse with NLOS links

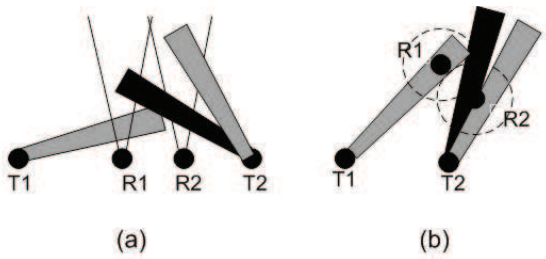

Figure 2.8: NLOS links (a) front-view; (b) top-view (T_1, T_2 are transmitters and R_1, R_2 are the corresponding receivers).

Nodes can increase spatial reuse by tuning the pointing and off-axis angles to create appropriate NLOS links. In Fig. 2.8(a), transmitter T_2 should not use pointing angle in solid dark color to communicate with receiver R_2 in order to avoid interfering with the ongoing communication between T_1 and R_1. Instead, T_2 can choose the grey pointing angle to transmit. In the scenario in Fig. 2.8(b), tuning the pointing angle does

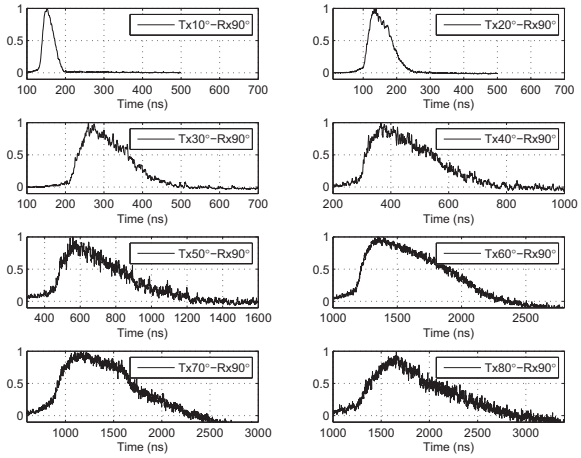

Figure 2.9: Normalized impulse response with various transmission pointing angles in the coplanar case (the transmitter is $100m$ apart from the receiver) [2].

not solve the collision problem with $T_1 \rightarrow R_1$ (dark direction). T_2 should use a different off-axis angle (grey direction) to communicate with R_2.

2.3 Impact of delay spread

The impulse response characterizes the received energy temporal distribution. We observe that *the UV channel delay spread varies when the transmission pointing angle and off-axis angle are tuned.* The measurement results in Fig. 2.9 show that *delay spread increases with the pointing angle.* This is because the common volume increases as the pointing angle increases (with other parameters fixed), and thus leads to addition of scattering paths of significant length, equivalently larger delay spread. We also see from Fig. 2.10 that the off-axis angle does not significantly impact delay spread; this is because the common volume does not change significantly when the off-axis angle is varied within certain range. We consider that one symbol is encoded per pulse and our objective is to avoid inter-symbol interference or ISI (meaning that we wish to ensure that there is no overlap between consecutively received symbols). This would

imply that the pulse width will differ for different point angles. Since the pulse width dictates the achievable data rate, the data rates will vary with pointing angles. Based on our measurements, it is shown that we can employ four data rates corresponding to four pointing angles. The data rates are $8R_t$, $4R_t$, $2R_t$ and R_t (lowest rate) when the pointing angle is 10 °, 30 °, 40 °, and 60 °, respectively. Note that when the data rate is doubled, the slot size (or pulse width) is halved. This feature has to be accounted for when designing a MAC protocol. We also observe that first three data rates (with the corresponding pointing angles) support full-duplex communications (as seen later); the last pointing angle with the lowest rate can only use half-duplex communications.

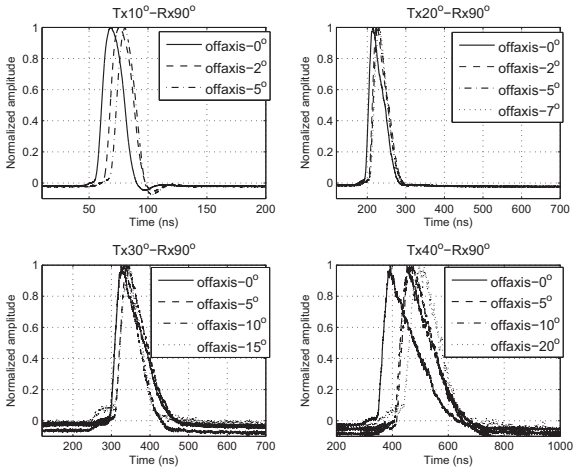

Figure 2.10: Normalized impulse response with various off-axis angles (the transmitter is $50m$ apart from the receiver).

2.4 Full-duplex communications

According to Reilly's common volume theory, simultaneous transmission and reception at the same node is theoretically possible, as long as there is no intersection between its own transmission beam and its collocated receiver's FOV. As an example, with a 30° FOV facing upwards and a 30° beam angle facing the horizontal axis, full-

duplex communications are achievable. In fact, it is easy to see that this is possible as long as the pointing angle is less than 60° (see Fig. 2.3) in theory. In reality, the multiple scattering needs to be taken into account.

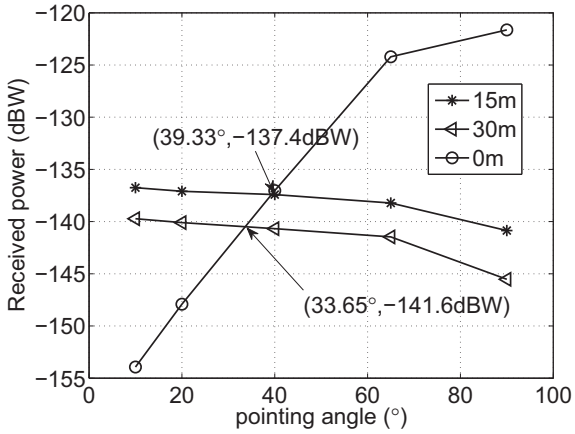

Figure 2.11: Illustration of the feasibility of full-duplex.

We performed experiments to test the feasibility of full-duplex communications in practice. We separate the transmitter and the receiver by distances of $30m$, $15m$ and $0m$ (collocated case), respectively, and measure the received power. The experimental data in terms of the received power versus the pointing angle is plotted in Fig. 2.11. We generally find that small pointing angles result in lower interference power to the collocated receiver and thus, are better suited for full-duplex communications. We observe that a pointing angle of about 40° with a receiver transmission separation of $15m$ (and 34° with a separation of $30m$) can support decodable SINR. Larger pointing angles possibly result in overlapped volumes and cause increased interference level; thus, full-duplex communications are not feasible in these regimes.

2.5 Summary

In summary, small pointing angles can help achieve full-duplex communications and small delay spreads (as seen earlier). However, at these angles, signals are more likely to be blocked by obstacles. Thus, a MAC protocol should opportunistically use small pointing angles, but at the same time, fall back to using larger angles as/when necessary.

Chapter 3

UVOC-MAC protocol design and evaluation

We design UVOC-MAC to account for the aforementioned factors in Chapter 2. In a nutshell, UVOC MAC effectively exploits NLOS communication links and full-duplex possibilities. It also mitigates some of the side-effects that arise due to these functions (discussed later). It also alleviates problems such as deafness and hidden/exposed node problems, commonly encountered in directional RF communications. In this chapter, we also provide a theoretical analysis and perform simulations to validate the efficacy of designed UVOC-MAC in delivering high throughput and yielding low collision. The collision probability is decreased by about 50% as compared to a MAC that does not take the UV PHY properties into account. In some cases, the throughput can increase by up to 4 times compared to a protocol that is agnostic of the UV-PHY [36].

3.1 UVOC-MAC design

In this section, we present our UVOC-MAC protocol. The design of UVOC-MAC is tied to the UV PHY properties: exploitation of *NLOS links*, the use of *multiple rates* and opportunistic use of *full-duplex* communications.

3.1.1 Design of the UVOC transceiver

Our system design is shown in Fig. 3.1. A similar design has been proposed for an indoor visible light communication system in [37]. The rationale behind our design is as follows.

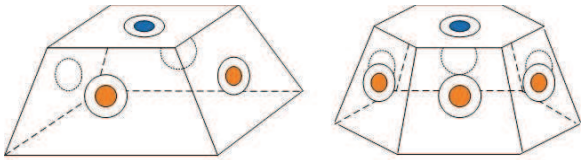

Figure 3.1: System design of the UVOC transceiver with 4 (left) and 6 (right) transmission directions (Dots on the side facets are the directional LED transmitters; dots on the top facet is the omni-directional photon receiver).

Since the photon detector is of high cost but the LEDs are of low cost, it is economical to implement multiple transmitters and a single receiver on a node. Given the dependencies between the pointing angle and full-duplex (introduced in Chapter 2) communications, a good design will be for a node to have multiple surrounding transmitters and a receiver on top. This configuration provides favors to easy installation and deployment on the helmet of a soldier or on the top of a vehicle. In Fig. 3.1, the dots on the side facets represent the LED transmitters (a transmitter could consist of multiple simultaneously emitting LEDs) and the dot on the top facet represents the omni-directional photon detecting receiver. With this configuration, the node[1] is a

[1]Without loss of generality, we also refer to a node as a transceiver.

"directional transmitting" and an "omni-directional receiving" device.

Note that the real implementation is not limited to the shape or the number of facets. The number of directions or facets is determined by the transmitting beam angle. For example, if the beam angle of an LED array on a side facet is 18°, then this device can have up to 20 side facets. In other words, the number of directions is a tunable system parameter. Implementing these nodes is complex and beyond the scope of this work; there are systems that have been conceived in prior work [32] and we use this structure in our simulations. Our physical layer measurements were performed with the apparatus described in Chapter 2.

3.1.2 UVOC MAC description

UVOC-MAC is a random access slot-based protocol. A single UV pulse is transmitted in each slot. The size of a generic slot is a multiple of what we call the *unit* slot size; the unit slot size corresponds to the highest data rate $8R_t$.

The key feature of UVOC-MAC is that it exploits spatial reuse by adaptively choosing: (1) the direction, (2) the pointing angle and, (3) the data rate. Recall that the data rate is determined *only* by the delay spread which is different for different pointing angles, as discussed in Section 2.3.

We first present a brief overview and then discuss operations of the protocol in detail.

3.1.2.1 UVOC-MAC overview

In the default case, nodes are in the idle state where they simply decode received control signals. Once there is new data from the upper layers (network), a node first checks certain tables (discussed later) to see if the desired communication is possible

at that moment and if so, to decide on the best configuration for use. It then chooses a full-duplex mode or a half-duplex mode depending upon whether or not the selected pointing angle is less than or equal to 40°. Based on our experiments in Chapter 2, we assume that with pointing angles less than or equal to 40°, full-duplex communications are possible; if the pointing angle is higher, then only half-duplex communications are possible.

The source and its destination exchange request-to-send (RTS) and clear-to-send (CTS) messages at a base rate R_t (we use similar nomenclature as in 802.11 for ease of presentation; however, the RTS and CTS messages are different here). The RTS message indicates the selected pointing angle; this also implicitly conveys the selected data rate (as discussed earlier, different data rates are possible with different pointing angles). Upon receiving the RTS, the receiver performs certain checks and replies with a CTS indicating the highest possible data rate (based on its interference patterns). The CTS is omni-directional by default and allows neighbors to update their neighborhood information. Then DATA is however sent directionally. There is a data rate announcement (DA) portion at the beginning of DATA frame. It is sent at the base rate and used to confirm the rate at which the remainder of the DATA is sent.

3.1.2.2 UVOC-MAC in detail

Next, we describe the UVOC-MAC in greater detail. We first describe a few data structures that are maintained for various purposes.

Connection table: Each node maintains a separate *Connection table* per neighbor. The table specifies the possible directions and pointing angles the node can use to communicate with this neighbor. The entries (an entry defines a combination of a direction and pointing angle) in the table are *ranked* in ascending order. There

is more than one way to rank the directions and pointing angles. In Section 3.3, we will compare two methods that can be used to rank entries in the *Connection table*. The space complexity of *Connection tables* is $O\{(\#\ of\ neighbors) \cdot (\#\ of\ directions) \cdot (\#\ of\ pointing\ angles)\}$.

Occupancy table: This table is maintained at each node and is updated when the node realizes that a new transmission is going to take place in its neighborhood. To elaborate, when a CTS is received from a neighbor node y, a node first checks the *Connection table* associated with y. It then sets timers in the *Occupancy table* corresponding to those entries which define transmission configurations that would interfere with y's reception. The space complexity of this table is $O\{(\#\ of\ directions) \cdot (\#\ of\ pointing\ angles)\}$.

Rate table: This table stores a record of the data rates in use in the owner's neighborhood. As we discuss later, this information is used by the node to appropriately choose a transmission rate (pulse width) when it acts as a transmitter. The table is updated when the owner checks the first part of every directional DATA packet that it overhears; we refer to this part of the data packet as the "DA" (data rate announcement). The space complexity of this table is $O\{\#\ of\ pointing\ angles\}$.

Receiving nodes table: This table enlists those neighbor nodes that are currently receivers and the time up to which they remain in the receiving mode. The contents of this table are disseminated using signaling messages and are used to combat effects such as deafness (explained later). The space complexity of this table is $O\{\#\ of\ neighbor\ nodes\}$.

Next, we look at specific functions that a node invokes with UVOC-MAC.

Check table: A procedure *check table* is invoked in two cases: (a) when there is new data to send and (b) when a CTS is to be sent. In the first case, the source

node checks the *Connection table* and *Occupancy table* to find the best combination of direction and pointing angle for use. If the best configuration (with the highest rank) is not available, it has to select an available configuration with a lower rank (it checks each possibility in descending order of rank). If a CTS message is to be sent in response to an RTS, the node checks the *Occupancy table* and transmits the CTS in all available directions with the lowest available pointing angle. The available directions are defined as those directions that are interference-free; this means sending from these directions will not cause interference to other ongoing communications.

Figure 3.2: MAC frame format

Frame control	Source and direction ID	Destination ID	**Duration**	**Selected Angle**	Frame body	**Time to wait**	FCS	**Gossip signaling**

Update tables: Every node has to update its aforementioned tables. The tables are updated with different stimuli and periods. The *Connection table* is constructed during the neighbor discovery process and updated by a neighbor maintenance procedure. Most neighbor discovery and neighbor maintenance schemes applied in networks with directional antennas (for example [26, 27]) can also be applied with minor changes in UVOC networks; thus, we can simply leverage these approaches. The implementation of specific neighbor discovery mechanisms for UV is out of scope of this thesis. In this chapter, we assume that the neighbors of a node are known for simplicity. We explore this in more detail in an associated Chapter 4. The *Occupancy table* and *Receiving nodes table* are updated upon receiving appropriate control signals and DATA packets. A node examines the source ID in a CTS message. It also check the IDs of other receiving nodes from what we call the gossip portion (discussed later) in both the RTS and CTS messages. Finally, it checks the duration fields in the CTS, RTS and DATA messages.

The *Rate table* is updated upon decoding the DA portion of every overheard DATA packet.

Gossip signaling: In order to mitigate the hidden node and deafness problems, nodes include information on receiving nodes in their neighborhoods, in the RTS and CTS messages. Note that only partial information can be sent with each message; only a subset of the nodes in the *Receiving nodes table* can be specified. There is a trade-off between the overhead incurred from the information (amount of information) and the efficacy in combating the aforementioned problems.

3.1.3 Frame structure

The frame structure with UVOC-MAC is shown in Fig. 3.2. The *Duration* field specifies the time duration of the communication. The *Selected angle* field specifies the pointing angle suggested by the receiver. The *Time to wait* field is used in relation to full-duplex communications. For instance, if node A is sending RTS to B and receiving from another node simultaneously, A will experience a collision if B sends the CTS right after receiving the RTS. Thus, A informs B using the *Time to wait* field, that it needs to wait for a certain time before sending the CTS. The *Gossip signaling* field contains information about neighborhood receivers as discussed in Subsection 3.1.2.2.

3.1.4 UVOC-MAC states of operation

The state transition diagram in Fig. 3.3 provides a complete depiction of UVOC-MAC. In the ***Idle*** state, nodes simply decode received messages and update the corresponding tables. Upon receiving an RTS, a receiver checks the *Time to wait* field and sets a timer. It sends CTS upon timer reaching zero. The direction to send

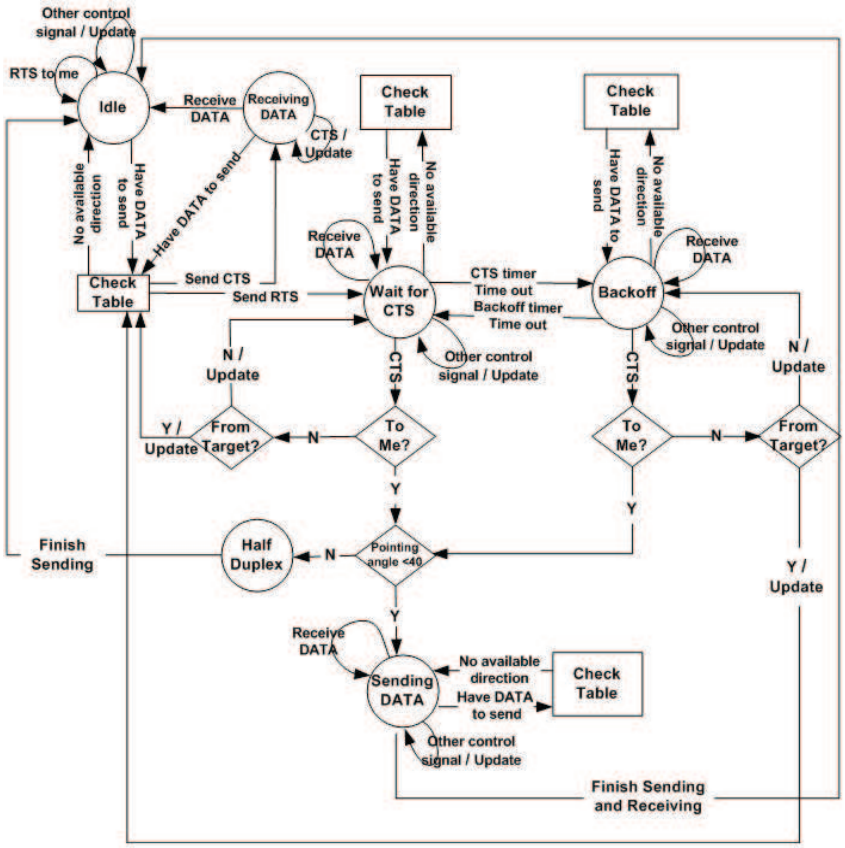

Figure 3.3: Detailed state transition diagram of UVOC-MAC.

the CTS is determined by the *Occupancy table*. At this point, the node transits to the ***Receiving DATA*** state.

If a node has new data to send in ***Idle*** state, it runs the *check table* procedure and sends an RTS (directionally with the best available configuration) and transits state ***Wait for CTS***. In this state, a node still decodes signals that it receives and update tables until it experiences a CTS timeout. If the node receives the expected CTS, it enters the state ***Sending DATA***; if not, upon timeout, it transits into state ***Backoff***.

When a node receives a CTS it checks the *Selected angle* field to identify the direction of transmission and the corresponding rate. It then sends DATA directionally. While sending DATA, the node can receive other intended RTS messages depending on

whether full-duplex communications are possible (the selected pointing angle is less than 40°). After having sent the DATA, the node stays in this state if it is still receiving from other nodes; otherwise it returns to the ***Idle*** state.

3.1.5 Preliminary evaluation of UVOC-MAC

Handling problems with directionality: The use of an omni-directional CTS transmission alleviates the problem of hidden nodes and deafness to a large extent. However, a receiver may sometimes have to use a directional CTS transmission since otherwise, it interferes with an ongoing communication. In such cases, these problems can arise. As discussed in Subsection 3.1.2.2, our gossip signaling successfully alleviates these problems to almost insignificant levels (as we observe with our simulations).

The exposed node problem is not an issue with UVOC-MAC since in most cases, full-duplex communications are possible.

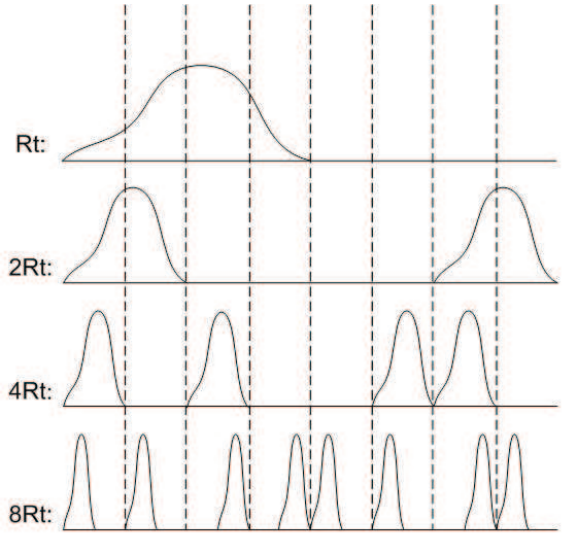

Figure 3.4: Dependencies between different data rates.

The use of multiple-transmission rates: UVOC-MAC is slotted and adap-

tively employs four data rates denoted by R_t, $2R_t$, $4R_t$, $8R_t$. The use of shorter pulse widths with PPM leads to higher data rates; recall that the choice of the pulse width is dictated by the delay spread which is in turn influenced by the pointing angle. The possibilities are shown in Fig. 3.4. A slot can accommodate two pulses. With binary PPM, a pulse in the first half of the slot (a sub-slot) indicates a '0' and a pulse in the second sub-slot indicates a '1'. From the figure we see that when a high rate projects interference on a low rate, there is practically no impact. This is because the energy-based detector simply computes the difference in the energy between the two sub-slots of a slot and in each lower rate sub-slot, a similar interference energy is added. Note here, those results are for a given distance. As an example, when an $8R_t$ rate transmission interferes with a $4R_t$ rate transmission, an interference pulse of identical magnitude is added to each sub-slot of the lower rate transmission. However, as one can immediately see, interference from lower rate transmissions impact higher rate transmissions. Thus, UVOC-MAC disallows lower rate transmissions if they project interference at the receiver of a higher rate transmission; it allows the initiation of lower rate transmissions as long as there is no such impact. This is the necessity of *Rate table* as well.

3.2 Analysis of collision probabilities

Our goal in this section is to show that the collision probabilities in UVOC networks are high if no regulation is used (like with ALOHA) via a simple analysis.

Let us assume that the distance between the intended transmitter and receiver is r_0 and the corresponding off-axis angle is φ_0. The interferer (say i) is located at distances r_i and has corresponding off-axis angles φ_i. We assume that interferers are uniformly distributed in the area with radius R centered at the intended receiver.

The φ_i is uniformly distributed in $\{0, \pi\}$; the analysis can be easily modified for other distributions of r_i and φ_i.

We consider two popular models for characterizing a successful reception: (a) the protocol model and (b) the physical model. With the two models, a collision occurs if:

$$P(collision) = \begin{cases} P(\max_k(p_k) \geq \frac{p_0}{T}) & (protocol\,model) \\ P(\sum_{k=1}^{m} p_k \geq \frac{p_0}{T}) & (physical\,model) \end{cases}, \tag{3.1}$$

where m is the total number of interfering nodes around the desired receiver; p_k is the received power from k^{th} interfering node; p_0 is the received power from the intended transmitter; T is the SINR threshold.

Recall that with UVOC signal detection is typically energy-based. Thus, the additive interference model is a good fit. In other words, the physical model can capture the behaviors with UVOC better. However, the protocol model offers simplicity and hence we consider it as well. With the protocol model we only consider the interference from the strongest interferer; thus, the performance with this model is somewhat optimistic.

The received signal power is given by $p_0 = p_t c r_0^{-a} \exp(-b\varphi_0)$ (refer to Eq. (2.3)) and the interference power by $p_i = p_t c r_i^{-a} \exp(-b\varphi_i)$. Here, the coefficient $c = 1/\xi$. Let $z = r_i^{-a} \exp(-b\varphi_i)$. The probability density function (PDF) of z can be easily computed given the PDFs of r_i and φ_i [38]. The pdf of z is given by (we omit the derivation due to space constraints):

$$f(z) = \begin{cases} \frac{1}{zb\pi} - \frac{\exp(-2b\pi/a)}{b\pi R^2} \cdot z^{(-\frac{2}{a}-1)} & (\frac{\exp(-b\pi)}{R^a} < z < \frac{1}{R^a}) \\ (\frac{1}{b\pi R^2} - \frac{\exp(-2b\pi/a)}{b\pi R^2}) \cdot z^{(-\frac{2}{a}-1)} & (\frac{1}{R^a} < z < \xi) \end{cases}. \tag{3.2}$$

Note here that path loss cannot be smaller than 1, z has an upper bound of ξ so that PDF of z needs normalization. We can then derive the probability that the

interference power exceeds the desired threshold (as per the protocol model) to be:

$$P_c = P(p_i \geq \frac{p_0}{T}) = P(r_i^{-a}\exp(-b\varphi_i) \geq \frac{r_0^{-a}\exp(-b\varphi_0)}{T})$$
$$= P(z \geq \frac{r_0^{-a}\exp(-b\varphi_0)}{T}) \tag{3.3}$$

$$= \begin{cases} \frac{2a\ln\frac{r_0}{R}+2b\varphi_0+2\ln T+a}{2b\pi} - \frac{a}{2b\pi}\frac{r_o^2}{R^2}\exp(\frac{-2b\pi}{a})\exp(\frac{-2b\varphi_0}{a})T^{\frac{2}{a}} \\ \qquad\qquad \text{when } \frac{r_0^{-a}\exp(-b\varphi_0)}{T} \leq \frac{1}{R^a} \\ \frac{a}{2b\pi}\frac{r_o^2}{R^2}(1-\exp(\frac{-2b\pi}{a}))\exp(\frac{-2b\varphi_0}{a})T^{\frac{2}{a}} \\ \qquad\qquad \text{when } \frac{r_0^{-a}\exp(-b\varphi_0)}{T} > \frac{1}{R^a} \end{cases}.$$

The final expression is simply obtained by integrating $f(z)$ over the desired interval. P_c is the probability that one of the interferers (say i) causes a collision at a particular target. With the protocol model, a collision occurs if at least one of these m interferers causes a collision. Thus,

$$P_{protocol}(collision) = 1 - (1 - P_c)^m. \tag{3.4}$$

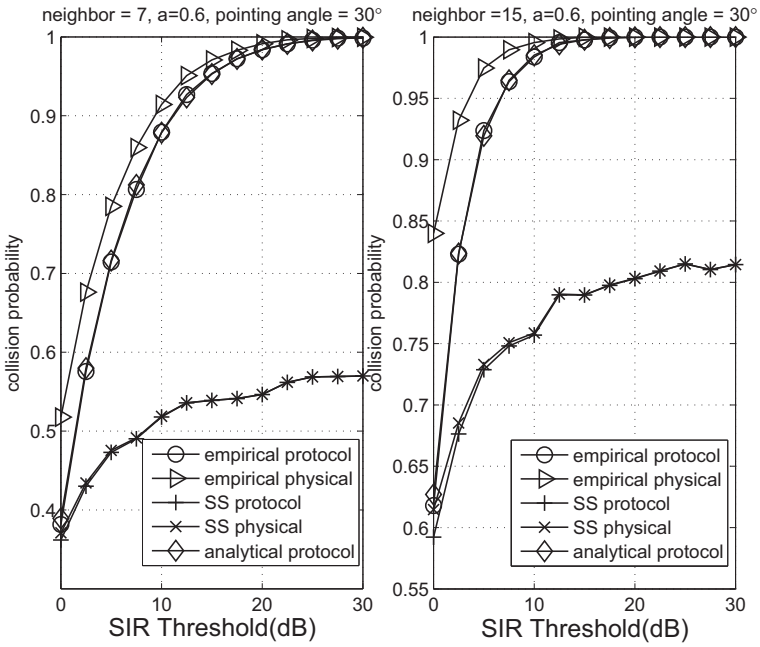

Figure 3.5: Collision probability for a given pair of nodes with distance r_0=67m and off-axis angle φ_0=10 ° within the area R=100m; the pointing angle is 30 °. Left figure shows the scenario with 7 interferers around and right figure with 15 interferers around.

Numerical results based on the above analysis are compared with Monte Carlo simulation results using the proposed non-coplanar path loss model (referred to as the empirical model) and the single scattering non-coplanar path loss model from [12] (referred to as the SS model). With each, we consider both the protocol and physical collision models. r_0 and φ_0 are set to $67m$ and 10°, respectively. r_i is distributed with probability density $2r/R^2$ in $\{0, 100\}m$ and φ_i is uniformly chosen from $\{0, 180\}^\circ$. Note here that the mean value of r_i is $67m$; r_0 is chosen to be this value. The comparisons are presented in Fig. 3.5. We observe that the analytical results match the results from the Monte Carlo simulations using the empirical model quite well. This is expected since the analysis uses the same path loss model as that of the empirical model. However, since the channel attenuation is different with the SS model, the results differ.

Second, we observe that the larger the number of interfering nodes, the larger the discrepancy between the analytical protocol model and empirical physical model. This is because as the number of interferers increases, the protocol model (accounts for only one dominant interferer) becomes less and less accurate.

Third, the discrepancy between protocol and physical models with the SS path loss model, is negligible. This is because the SS model inherently assumes that received power is zero if the interfering node has no common volume with the target FOV. With this, the accumulated interference is very close to the maximum interference. Recall however, this model is inaccurate as we see in our experiments.

Fourth, the SS model yields smaller collision probabilities than the empirical non-coplanar model as the SINR threshold increases. This is because, the multiple scattering is captured by the empirical model (Eq. (2.3)) and this accurately captures higher levels of interference thus, increased collision probabilities. In fact, due to this, the collision probability does not reach '1' for the range of SINR values considered.

Next, we consider the physical model in our analysis. It is extremely difficult to compute the distribution of $\sum_{k=1}^{m} p_k$. Thus, we use the central limit theorem [38] to approximate it as a Gaussian distribution. Towards this, we need to compute the mean $E(z)$ and the variance $Var(z)$ of z. Since the pdf of z is calculated, it is easy to compute these values and we provide them below.

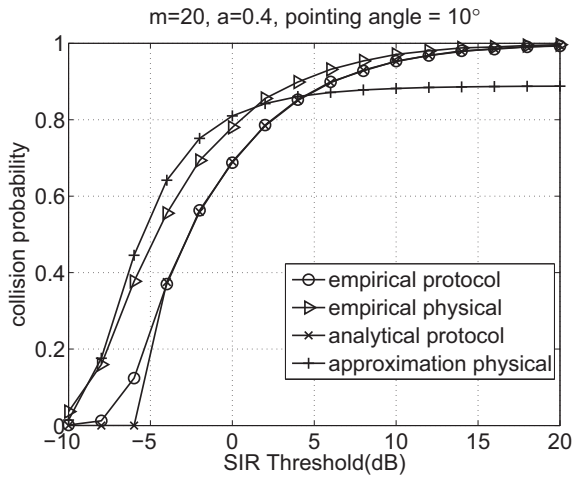

Figure 3.6: Collision probability for a given pair with distance r_0=67m and off-axis angle φ_0=10 °. Interferers number is 20.

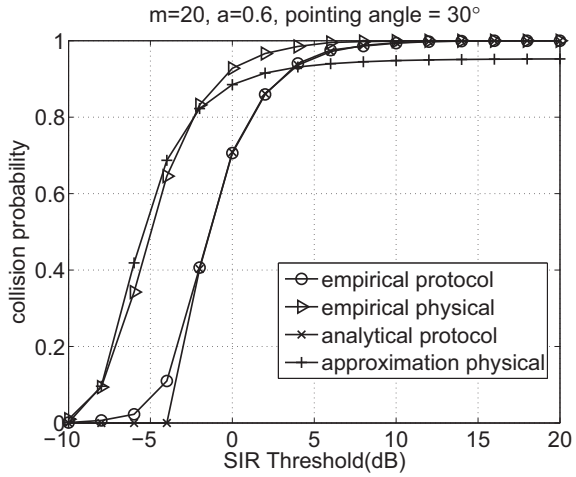

Figure 3.7: Collision probability for a given pair of nodes with distance r_0=67m and off-axis angle φ_0=30 °. Interferers number is 20.

$$E(z) = \frac{2[\exp(-b\pi) - 1]}{\pi ab(1 - \frac{2}{a})R^a} + \frac{1 - \exp(-\frac{2b\pi}{a})}{b\pi R^2(1 - \frac{2}{a})}\xi^{1-\frac{2}{a}} = \mu. \tag{3.5}$$

$$Var(z) = \frac{\exp(-2b\pi) - 1}{2\pi ab(1 - \frac{2}{a})R^{2a}} + \frac{\xi^{2-\frac{2}{a}}}{2b\pi R^2(1 - \frac{1-\exp(-\frac{2b\pi}{a})}{a})} = \sigma^2. \tag{3.6}$$

With μ and σ^2 for z computed as above, the collision probability can be expressed as

$$\begin{aligned} P_{physical}(collision) &= P(\sum_{i=1}^{m} p_i \geq \frac{p_0}{T}) = P(S_m > A) \\ &= P(Z_m > \frac{A - m\mu}{\sqrt{m}\sigma}), \end{aligned} \tag{3.7}$$

where $S_m = \sum_{i=1}^{m} p_i$, $A = \frac{p_0}{T}$, $Z_m = \frac{S_m - m\mu}{\sqrt{m}\sigma}$ and m is the number of interferers. The results are shown in Figs. 3.6 and 3.7. From these two figures, we can see the CLT approximation holds well and there is a match with the Monte Carlo physical protocol results.

Multiple rates: The above analysis (Eq. (3.7)) can be further extended to capture the impact of multiple rates. We can individually calculate the collision probability for transmissions at each data rate knowing that higher rate transmissions have no impact on the lower rate transmissions. For example, transmissions at data rate R_t are only affected by other transmissions at rate R_t. On the other hand, transmissions at rate $8R_t$ are affected by transmissions at all rates. Let r_1 denote $8R_t$, r_2 denote $4R_t$ and so forth. Let the $P(r_1)$, $P(r_2)$ and so on be the probabilities with which interferers transmit with those rates. The corresponding expressions would be:

$$P_{physical}(collision|r_j) = P\left(Z_m > \frac{A - \sum_{i=1}^{5-j} \mu_i P(r_i)}{\sqrt{\sum_{i=1}^{5-j} \sigma_i^2}}\right). \tag{3.8}$$

The collision probability averaged over the set of all possible data rates becomes

$$P_{physical}(collision) = \sum_{i=1}^{4} P_{physical}(collision|r_i) \cdot P(r_i). \tag{3.9}$$

The analytical results produced by Eq. (3.9) yield a collision probability of 70.59% if we were to use the distributions for $P(r_j)$ from simulation data in Fig. 3.9 when a node has 7 neighbors on average and the SINR threshold is 10dB. This extremely high collision probability is due to the lack of coordination in choosing directions, pointing angles and rates. As we see in the following section, the use of UVOC-MAC drastically decreases this collision probability and provides significant throughput improvements.

3.3 Simulation results

In this section, we evaluate the performance of UVOC-MAC with OPNET version 16.0 [39].

Simulation settings: The default simulation settings are listed in Table 3.1. The chosen transmission power corresponds to the typical short UV transmission range (*approx 100m*). A packet generated by a node is targeted towards a randomly chosen neighbor. We use the channel attenuation and delay spread results from our PHY experiments reported in Chapter 2 to characterize signal propagation. All the path propagation parameters are consistent with that in the outdoor experimental settings. The nodes are randomly distributed in the deployment area (network size).

3.3.1 Setting up and maintaining connection tables

*Connection table*s are assumed to be available upon neighbor discovery. There are two ways to rank the entries in *Connection tables*: *high-rate based* method and *path-loss oriented* method. With the former, ranks favor lower pointing angles (thus higher rates). The entries associated with pointing angle leading to half-duplex communications are always ranked lower than those with smaller pointing angles. With the *path-loss*

Table 3.1: Simulation settings

Transmission power	4mW (2mW for each LED)
# of directions	6
SINR threshold	10 dB
Collision model	Physical (accumulative) model
Traffic pattern	200 pkt/sec
Packet size	1024 bits/pkt (data portion)
Network size	100m by 100m
# of nodes	8
RTS length	76 bits (without gossip)
CTS length	80 bits (without gossip)
DATA head length	76 bits
Simulation time	20 seconds

oriented method, ranks are assigned as per the measured path loss. Configurations with smaller path loss are assigned higher ranks.

3.3.2 Comparison of high-rate based and path-loss oriented methods

With UVOC-MAC, we find that the above two maintenance methods result in different *data rate usage distributions, throughput* and *collision probabilities*. Naturally, the *high-rate based* method results in a more aggressive usage of high rates. The distributions with which the different data rates are used (from simulations) are shown in Table 3.2. We see that 93% of the transmissions are at rate $8R_t$. With the *path-loss oriented* method, in contrast, the rate $2R_t$ is used by about 90% of the transmissions. The latter effect is due to the fact that lower pointing angles result in higher path loss for a fixed off-axis angle (seen in our experiments in Chapter 2); thus, a lower path loss

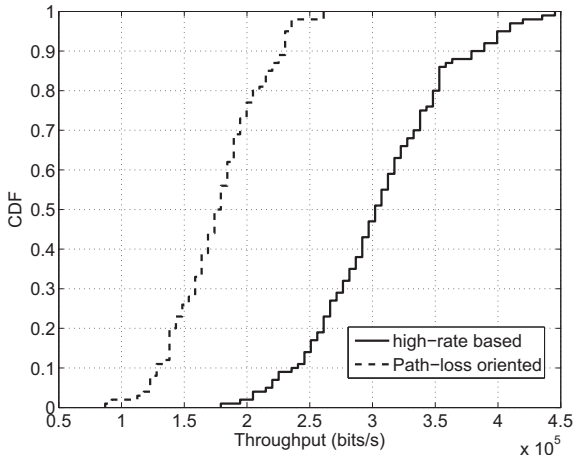

Figure 3.8: Throughput comparison with the two ranking methods.

typically corresponds to a higher pointing angle and thus a lower rate. Note here that in our simulations we find that the lowest rate is only scarcely chosen since it corresponds to a half-duplex mode with the lowest rank; almost always a higher rate is possible.

As seen from Fig. 3.8, the average throughput with the *high-rate based* method is about 74% higher than that with the *path-loss oriented* method. This is because higher rates directly lead to higher throughput.

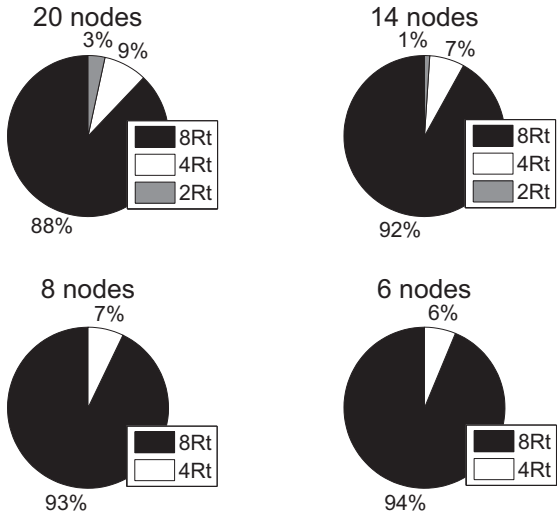

Figure 3.9: Usage of the different data rates with node density.

In Fig. 3.9, we depict the usage of the different rates with varying node densities. The *high-rate based* method is used. We observe that as the node density increases, there is an increased requirement towards using higher pointing angles and correspondingly lower rates. This is because, spatial reuse is more difficult if only the highest rate is used; the pointing angle dimension will have to be more heavily utilized in such cases (higher pointing angle).

3.3.3 Collision probabilities with UVOC-MAC

We observe the packets that are sent/received in a sample duration of 20 seconds. The results indicate that the collision probability (expressed as a percentage) is 19.83% with the *high-rate based* method while it is 23.61% when the *path-loss oriented* method is adopted. Note the drastic reductions that are achieved compared with 70.59% without UVOC-MAC (as discussed in Section 3.2).

3.3.4 ACK transmissions

We observe from simulations that if we transmit ACKs at the base rate with UVOC, collisions might occur during the ACK exchange period. This is primarily because these low rate transmissions affect other high rate transmissions in the node's vicinity. Due to channel asymmetry, it is not a good idea for the receiver to send an ACK using received data rate. It is because in the ACK exchange, the receiver takes the role of the transmitter and the pointing angle and off-axis angles which are key in determining channel quality will have changed. Since nodes already have prior rough knowledge about communications in their neighborhood from updates, we seek to examine if ACKs are indeed needed. Unless stated otherwise, we perform simulation

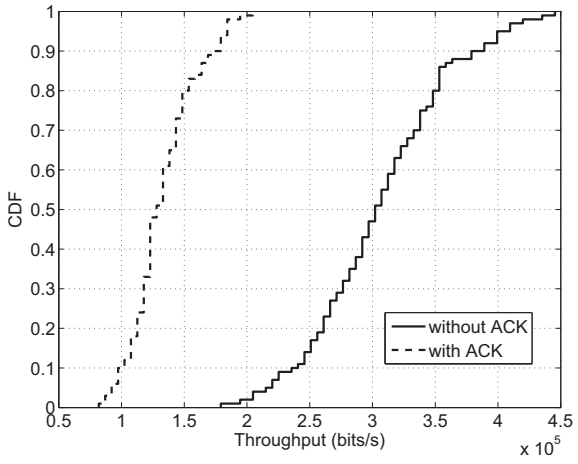

Figure 3.10: ACK impact on throughput.

evaluations to understand if ACKs indeed help in improving throughput.

Fig. 3.10 depicts the performance with or without the ACK process. We see that the ACK procedure brings down the throughput by about 132%; this loss is mainly due to the overhead and collisions with high rate transmissions. Also, as we indicted in the previous paragraph, the collision rate in the DATA part is very low (around 0.5%). Therefore, we conclude that it is better for UVOC to not include ACKs.

Table 3.2: Usage of different data rates

Rates	R_t	$2R_t$	$4R_t$	$8R_t$
High-rate based	0	0	7%	93%
Path-loss oriented	0	89.3%	4.9%	5.8%

3.3.5 Overhead in UVOC-MAC

The RTS-CTS handshake in UVOC-MAC allows nodes to update their tables. This padded information however inevitably introduces overhead. The overhead ratio is related to the packet size and can be calculated as in Eq. (3.10). For a packet size of

40k bits, the overhead ratio is only 1%.

$$Overhead = 1 - \frac{Received\ DATA\ length\ (Pure\ DATA)}{Transmitted\ (RTS + CTS + DATA\ (total))\ length}. \tag{3.10}$$

3.3.6 Impact of tuning the pointing angle

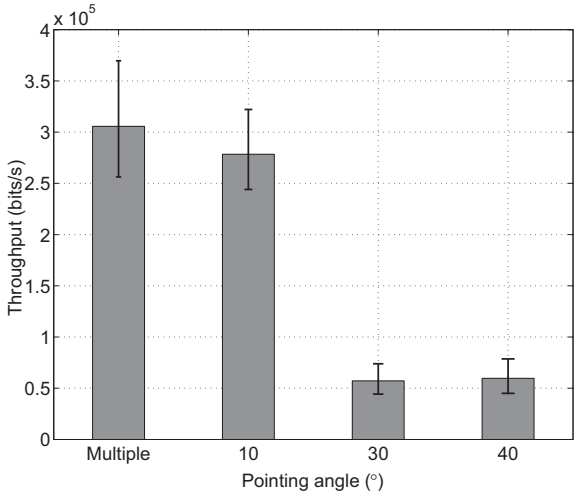

Figure 3.11: Impact of pointing angle on throughput.

One significant advantage of UVOC is the extra dimension of spatial reuse; as discussed this is exploited by varying the *pointing angle*. The additional spatial reuse from exploiting the pointing angle can be expected to lead to higher throughput. We conduct four sets of simulations to quantify the throughput improvements due to varying the pointing angle. The first set of simulations uses UVOC-MAC. In the other three sets, the pointing angles are fixed at 10°, 30° and 40°. Thus, adaptivity in the additional dimension is eliminated. From Fig. 3.11, we see that the usage of multiple pointing angles brings about a 10% throughput increase compared to a case with a fixed 10° pointing angle. Compared with the other two cases, the throughput gains are much higher. The throughput is about 4.3 times that with fixed pointing angles of 30° or 40°. These results first demonstrate lower pointing angles are better (as one might expect).

More importantly, if pointing angles were to be statically assigned, it may turn out that some pairs would have to use higher pointing angles (especially in dense scenarios). This would drastically hurt their throughput. UVOC-MAC provides adaptive tuning to opportunistically exploit the usage of lower pointing angles which in turn, improves throughput.

As we discussed in the end of Chapter 2, the use of links with fixed small pointing angle can be precluded due to interference and thus high pointing angle might be necessary. By varying the pointing angle, UVOC networks bear large spatial reuse opportunities and are capable of producing high throughput.

3.3.7 Exploiting full-duplex communications

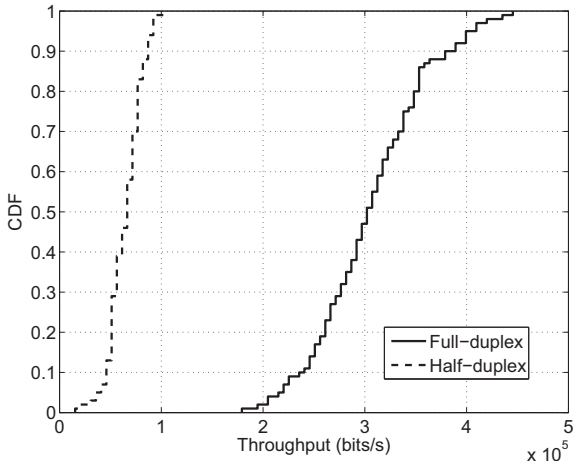

Figure 3.12: Throughput benefits of full-duplex communications.

Recall that UVOC nodes can communicate in full-duplex mode with small pointing angles. As discussed in Chapter 2, a conclusion drawn from the experiments was that pointing angles lower than 40° support full-duplex communications. We perform two sets of simulations to quantify the throughput gains from full-duplex opportunities

with UVOC-MAC. In the first set, nodes can switch between half-duplex mode and full-duplex modes as per UVOC-MAC. In the other set, nodes operate in the traditional half-duplex mode. Fig. 3.12 depicts the throughput from the two sets. A 377.26% increase in throughput is brought by full-duplex opportunities. Again, note that in dense settings, if pointing angles were statically assigned, certain pairs may be forced to just operate in half-duplex modes and thus, achieve much lower throughput than what is possible with UVOC-MAC.

3.3.8 Impact of gossip signaling

If a node is in receiving state, it will most likely not be interfered with UVOC-MAC protection. But it will lose updates about ongoing communications in the network due to this kind of deafness. Gossip signaling increases the awareness with regard to ongoing communications in a node's neighborhood. This can significantly reduce unnecessary RTSs that would interrupt other transmissions and/or fail to establish connections. We evaluate the impact of gossip signalling in a 20-node topology. From the simulation results we observe that the gossip procedure reduces the number of RTS messages initiated by 32.3%. These are those RTS messages that did not result in a successful connection establishment.

3.3.9 Impact of system parameters

Finally, we examine the impact of various system parameters on the performance of UVOC-MAC.

Number of directions: Generally, the more the number of directions the better the spatial reuse. We find that UVOC-MAC effectively exploits an increase in

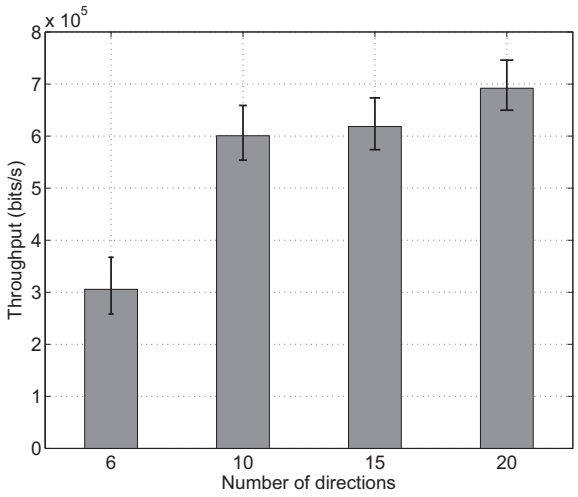

Figure 3.13: Throughput with different numbers of directions.

the number of directions. From Fig. 3.13, we observe the throughput improves by 134.39% as the number of directions increases from 6 to 20.

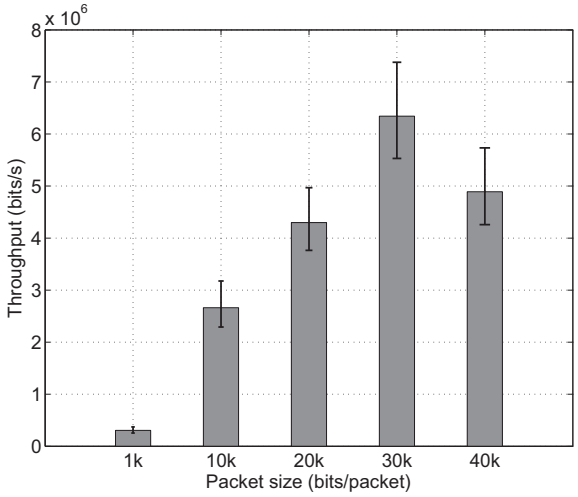

Figure 3.14: Throughput with different packet sizes.

Packet size: Using longer packet sizes can potentially increase throughput due to decrease in the overhead per packet. However, it could also increase the collision probabilities. Fig. 3.14 shows the throughput trend with increasing packet size. To begin with, increasing packet size brings higher throughput. However, when the packet

size increases to beyond a certain extent, the interference dominates and the throughput falls. The collision probability is 62.83% with 40k bits/packet compared with 19.83% for 1k bits/packet.

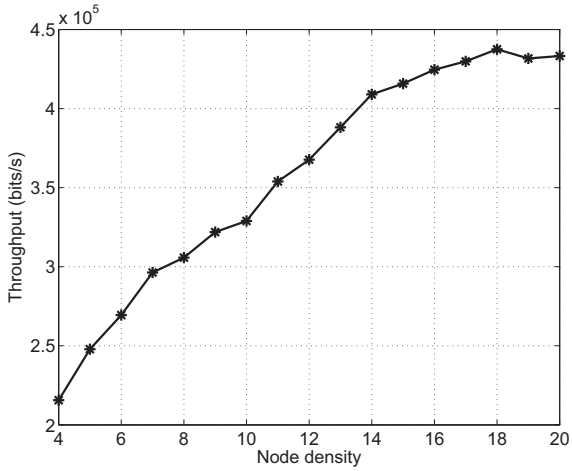

Figure 3.15: Throughput with different node densities.

Node density: To examine the impact of node density, we vary the number of nodes from 4 to 20. The nodes are uniformly distributed in a $100m$ by $100m$ area. Fig. 3.15 shows the network throughput versus node density. We see that the throughput is higher as node density increases. When the network is extremely dense (≈ 20 nodes) the throughput slightly decreases due to increased collisions. These results indicate that UVOC-MAC maintains high throughput even with high node densities.

Impact of mobility: All above mentioned simulations are run in a static network. The *connection tables* are given and not changed in time. We want to examine how UVOC-MAC performs when the nodes are mobile. We expect these networks to be deployed in scenarios with pedestrian motion and thus, we consider low speeds. In Fig. 3.16, there is only one randomly chosen moving node and five different moving speeds are considered. We consider a random walk model where the node moves 1 m

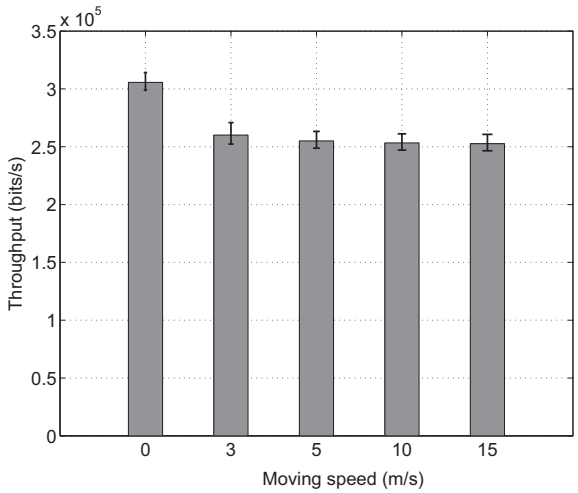

Figure 3.16: Throughput comparison at different moving speeds with a single moving node in network.

forward at each step; the rate at which it moves depends on the speed. The throughput decreases by 14.8% when the moving speed is 3 m/s. Higher speeds further bring reductions but the reduction is much smaller compared to that of the case when the speed changes from 0 m/s to 3 m/s. These reductions are because the connections to moving node are sometimes expired (due to obsolete connection info) after it moves for certain time. We also check the performance when there are multiple nodes moving in the network. Fig. 3.17 shows the throughputs in four cases when one node, two, three or four nodes, respectively, are moving simultaneously. The moving speed for each node is 5 m/s and the moving direction for each node is chosen at random. From this figure, we see that the throughput decreases with an increase in the number of mobile nodes and the largest throughput reduction compared to static case is about 64.2%. We can conclude our UVOC-MAC is capable of tolerating minor changes of neighbor distribution. In order to be deployable in a fast-changing network, some neighbor maintenance techniques should be considered. We will examine this in future work.

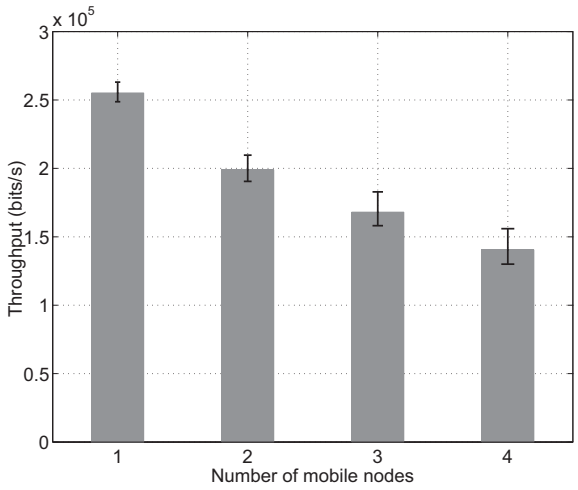

Figure 3.17: Throughput comparison with different numbers of moving nodes. Moving speed is 5 m/s.

3.4 Summary

Communications with UV light is a good alternative to RF in many scenarios. Higher layer design for UV networks will have to conform to UV PHY properties. In this chapter, based on the understanding of the characteristics of UV PHY layer, we design UVOC-MAC, a MAC protocol that inherently accounts for the UV properties. We perform analysis and simulations to demonstrate that UVOC-MAC effectively exploits spatial reuse in extra dimensions. The throughput can improve by up to 4 times compared to a MAC that does not adaptively exploit the properties of the environment.

In the future, effects of UV properties such as back-scattering and wavelength-duplexing can be considered. Back-scattering is similar to the concept of side-lobe in directional antennas and wavelength-duplexing could open another dimension of spatial reuse.

Chapter 4

Credit-collection based neighbor discovery method for UVOC

For most military settings, the use of a centralized infrastructure is infeasible. Rapid reconfigurations and re-deployments are more the norm than the exception. Thus, ad hoc network deployments are typical in these contexts. Although there are some preliminary studies on medium access control in Chapter 3, the network initialization process has not been considered before. This process is crucial for bootstrapping a network prior to operations. Since in typical cases, nodes are randomly deployed, each node will have to determine the existence as well as the locations of the surrounding nodes in order for information sharing. Thus, a distributed neighbor discovery algorithm is desirable for establishing connectivity and for self-organizing the network. Development of such algorithms is the focus of this work.

Each node in the network is assumed to be capable of directional transmissions and omni-directional receptions, given the unique properties of the UV channel in Chapter 2. At the termination of neighbor discovery, a node in the network will have

sufficient knowledge about each of its neighbors.

Based on the propagation properties derived in Chapter 2, we design our neighbor discovery mechanisms. First, we consider a well-planned deployment, wherein nodes are assigned a fixed set of synchronized directions in which they can point their beams. In other words, the sense of direction is global; when a node receives a message from a neighbor, it can immediately determine the direction in which that neighbor transmitted its packet. We refer to this as neighbor discovery with direction synchronization. Second, we propose a more general neighbor discovery scheme, wherein nodes cannot determine a neighbor's direction by simply receiving the latter's transmission. The approach incorporates a novel handshake mechanism that is tightly dependent on the UV propagation characteristics [40].

4.1 Neighbor discovery with direction synchronization

Since NLOS links are feasible for UV communications, we can expect that given several transmission directions from a remote node to the node of interest, more than one direction can be utilized for communication between this pair of nodes. Although the link loss varies significantly with pointing direction, all those links can function well if the background noise and multiuser interference are negligible. In practice, some small angles might not be feasible due to blockage by buildings, trees, vehicles, even people. However, some other directions resulting in links of degraded quality may still be usable. The neighbor discovery process should make all such link information accessible to neighbors. Each node must be made aware of the different NLOS links that it can use to its neighbors and the ranking of these links in terms of channel quality. It suffices if each node is made aware of the existence of its neighbors in a network with omni-

directional antennas (as most prior efforts on neighbor discovery demonstrate). However, if directional antennas are utilized, a node needs to know the direction in which to send information to its neighbors. It is mentioned in [26] that GPS might be necessary or an angle-of-arrival (AOA) estimation technique must be applied to solve this problem. For many reasons including security, GPS might be unavailable in military contexts and low cost is preferable. We choose to develop an algorithm without a requirement for GPS. On the other hand, based on the energy detection and channel scattering characteristics of a UV communication system, AOA estimation is hard to implement. Thus, the issue of how to send information has to be solved by the neighbor discovery algorithm itself.

The neighbor discovery algorithm described in this section is specifically designed for a scenario with *direction synchronization*. Direction synchronization assumes that all the nodes in the network have the same number of fixed directions. The direction labeled "zero" in the direction table of each node should always reflect the same direction. With this assumption, once a receiver node, B receives a message sent by a transmitter node A, containing information with regards to the direction of node A's transmission, it can easily infer the best direction to reply to A, based on geometric symmetry (detailed discussion later). The reciprocal property of the channel is attributed to a homogeneous environment; the transmission pointing angles, the beam divergence angle and the FOV are set to be the same across all nodes. In this case, no handshake mechanism is necessary for direction synchronization. If the restriction of direction synchronization is relaxed, a feedback mechanism should be incorporated. We shall discuss such cases in the next section.

To rank each direction, intuitively, a receiver can record the received energy from a specific transmitter, from each direction. However, since we do not assume any multiuser detection technique, it is impossible to distinguish the desired signal from

interference or background noise. We adopt a credit collection method to solve this problem.

4.1.1 Algorithm description

The number and names of neighbors are unknown to the present node. The only information that the node has is the format of the transmitted message for the purposes of decoding. Only one type of message is transmitted in the network, i.e., the request packet which contains information of *node identification (ID) and direction ID*. Here, we require that each node has unique identification, which could be MAC address or any sequential code for node differentiation.

Every node sends out a request packet of duration of τ and receives all the time (due to the full-duplex capability). The interval between two successive transmissions follows exponential distribution with parameter λ. The node chooses the sending direction randomly.

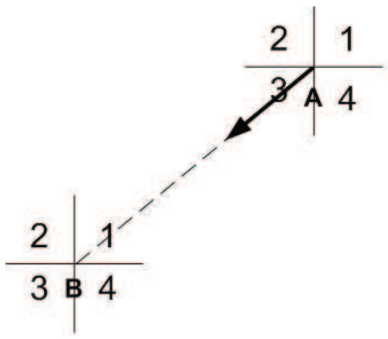

Figure 4.1: Illustration of direction synchronization.

Once a receiver node receives a request from another node, it extracts the information of source ID and the associated direction ID. Because all the nodes are assumed to be direction-synchronized, the receiver can figure out the direction in which

it can respond to the transmitter. As shown in Fig. 4.1, if node A transmits a request to node B using direction "3" and node B successfully receives A's request, node B could conclude that its direction labeled "1" could be used to communicate with node A. Correspondingly, a reception from direction "2" suggests responses in direction "4", and vice versa. Since whether or not a node transmits is a random event, the total interference level the receiver experiences is a random variable (as investigated in Chapter 3). Thus, the successful reception of packets in a given direction follows a random process. Multiple transmissions of the same packet may lead to different outcomes; it may or may not be received depending on the interference levels from other nearby transmitters.

We collect the successfully received neighbor requests (ND_request contains the transmitter's ID and the chosen transmitting direction ID), and create a *credit table* in which the directions of the successful requests are recorded. Each direction is then ranked as follows. A direction to a neighbor is assigned a credit upon correct detection of a packet from that direction. Each node transmits in a randomly chosen direction; the choice of direction is uniformly distributed. Packets sent from a *better* direction are deemed to be received with a higher possibility. Thus, for such directions the credits collected will be higher. Each node will then establish a ranked *connection table* for every neighbor after a sufficiently large number of transmissions are received, based on *credit table*. The more the packets are exchanged, the more reliable is the table. The steps of the process are formalized with the pseudocode below:

K = total no. of directions;

Clear credit_table;

Transmitter:

if timer is up **then**

ND_request.source_ID=self_id;

ND_request.direc_ID=uniformly_choose(K);

send ND_request;

set timer;

end if

Receiver:

if receive ND_request **then**

neighbor_ID=ND_request.source_ID;

prev_direc=ND_request.direc_ID;

current_direc=[prev_direc+ceil(K/2)] mod K;

credit_table.node[neighbor_ID].direction[current_direc]++;

end if

4.1.2 Simulation results

We study the neighbor discovery performance with direction synchronization, via simulations using OPNET version 16.0 [39]. The simulation settings are listed in Table 4.1. The chosen transmission power corresponds to the typical short UV transmission range (approx 100m). We adopt the channel attenuation results from Chapter 2 to characterize signal propagation. The SIR threshold is set to 1dB, indicating that the signal is decodable if the signal power is slightly larger than the total interference power.

In Fig. 4.2, we consider two criteria for neighbor discovery. First, we impose a somewhat less stringent requirement (loose criterion) wherein a node seeks to establish

Table 4.1: Simulation settings

Transmission power	4 mW
# of directions	6
SIR threshold	1 dB
Collision model	Physical (accumulative) model
Data rate	1 Mbps
Traffic pattern	10000 pkt/sec
Packet size	32 bits/pkt
Network size	100m by 100m
# of nodes	10

only the top one third of all the directions to a neighbor; the ranking among these directions is inconsequential. For example, assume a node can perform transmissions in 6 directions, then we only focus on the top 2 directions between that node and any other node; the relative ordering of the two directions in terms of performance is not considered. We consider a direction to be correctly found if it is accurately classified as being or not being within the top one third of the possible directions.

Next, we consider a strict requirement (strict criterion) wherein, the relative ordering of the directions is necessary. In other words, the direction should be correctly classified in terms of its rank. In the solid curve, we consider the two top directions, now correctly ordered. In other words, a direction is considered to be found correctly if and only if it is exactly at the right position in the *connection table*.

It is observed that with the loose criterion, reaching a discovery probability of 90 % takes less than 0.005s, while the solid curve reaching the same probability takes about 0.05s. When there is a strict temporal constraint on the neighbor discovery

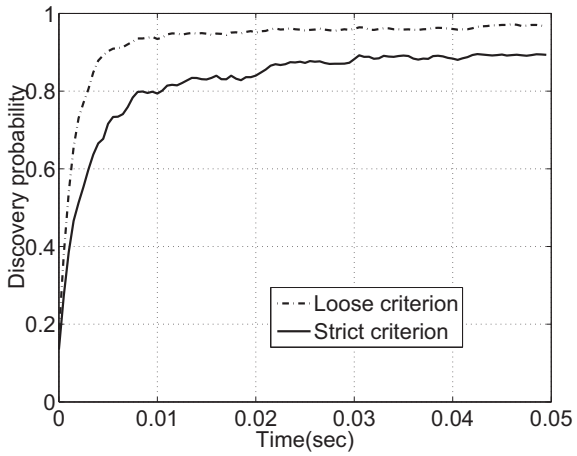

Figure 4.2: The discovery probability with loose and strict criteria.

process, the algorithms can still work well without providing poor directions (the top one third directions can be found) for communications among neighboring nodes.

We observe that while we can discover about 80 % of the directions with high accuracy relatively quickly, ordering among the top directions takes much longer (as evident from the figure). This is because, it is hard to distinguish between the top few directions; this is in some sense to advantageous since it becomes less important as to which direction is ranked higher.

4.2 Neighbor discovery in a general scenario

In Section 4.1 we considered a scenario in which all the nodes in the network were synchronized with regard to their sense of direction. Next, we relax the synchronization constraints to consider a more general case. In military applications, the network deployment typically requires to be accomplished quickly. For example, the nodes might just be thrown from an aircraft on to a sensor field. Thus, direction synchronization is difficult to obtain. Even if direction synchronization is ensured in the initial deployment, it is likely to be lost later due to the dynamics of the environment.

Thus we seek to design an algorithm without the assumption of direction synchronization. With our approach, the receiver resorts to a handshaking process to build what we call a *credit table* (discussed later). The different directions are then ranked based on the *credit table*. As one might expect, the time needed for neighbor discovery grows as compared to the case wherein nodes are synchronized with respect to directions.

4.2.1 Algorithm description

There are two types of packets: (1) the request packet which is identical to the one described in Section 4.1; (2) the feedback packet. Like with the direction synchronization scenarios, a node receives all the time and sends a request packet (see clarification below) after a waiting time which is exponentially distributed with parameter λ. After each successful reception of a request, a feedback packet is expected to be sent back to the sender of the request. The feedback packet contains *the ID of the node that sent the request, the node's own ID, and an identifier of the direction (direction ID) specified in the received request packet*. When it is time to send, each node first checks if any feedback was generated but unsent. If that is the case, then the node chooses to send the feedback and then delete it from the queue of unsent feedback packets. The queuing policy is first-in-first-out (FIFO). If no feedback requests are in the queue, the node sends a request packet (as previously discussed). Note here that, a node does not exactly know the direction using which a request packet sender can be reached; it can only either randomly choose a direction to send or omni-directionally transmit the feedback. Recalling the structural design of the node in Chapter 2, an omni-directional transmission can be realized as long as all the LEDs on the node are powered up and send the same packet simultaneously. If the feedback packet is only sent in one direction (chosen randomly), the likelihood that the original sender receives the feedback is small.

However the downside of sending feedback omni-directionally is that the interference increases. With our approach, we choose to send feedback packet omni-directionally. The overall transmission power of the LEDs on a facet (one direction) is equivalent to the power level with which the request packet is sent. To solve the interference issue, when the feedback packet is being sent, the LEDs could scale down their transmission power levels appropriately. More details and comparisons are deliberated on in a following simulation subsection.

We wish to emphasize here that the node chooses to send feedback whenever such packets are available to send; in other words feedback packets are prioritized. This is because our objective is to finish an existing handshaking process prior to starting new handshaking processes. Only when the feedback is successfully received by the original sender of the request, the handshaking process is considered completed. In some extreme cases, all the transmission opportunities will be occupied by feedback packets for a long time thus, precluding a node from sending its own request packets. When this situation occurs, the node under discussion may keep replying to the same node repetitively, which causes a wastage of available resources. More importantly, the node under consideration is deprived of opportunities to find its neighbors. Therefore, we count the number of successive feedback packets sent and set a predefined threshold beyond which, priority is given to request packets. After sending a request packet, priority is returned to the feedback packets and the counter is re-initialized. We also use a finite buffer for storing queued feedback packets; if this buffer is filled, additional feedback packets are discarded. This allows us to control the maximum number of feedback messages that a node has to respond to.

The steps of our algorithm are succinctly captured by the pseudocode below:

K = total no. of directions;

Clear credit_table;

Transmitter:

if timer is up **then**

  if feedback.head=0‖ send feedback 3 times consecutively **then**

    ND_request.source_ID=self_id;

    ND_request.direc_ID=uniformly_choose(K);

    send ND_request;

    set timer;

  else

    ND_feedback.source_ID=self_id;

    ND_feedback.dest_ID=feedback.head→neighbor_ID;

    ND_feedback.prev_direc=feedback.head→prev_direc;

    ND_feedback.direc_ID=Omni-direction;

    send ND_feedback;

    set timer;

  end if

end if

Receiver:

if receive ND_request **then**

  neighbor_ID=ND_request.source_ID;

  prev_direc=ND_request.direc_ID;

else

  if receive ND_feedback **then**

    neighbor_ID=ND_feedback.source_ID;

prev_direc=ND_feedback.direc_ID;

credit_table.node[neighbor_ID].direction[prev_direc]++;

end if

end if

4.2.2 Simulation results

Table 4.2: Simulation settings

Request packet size	32 bits/pkt
Feedback packet size	80 bits/pkt

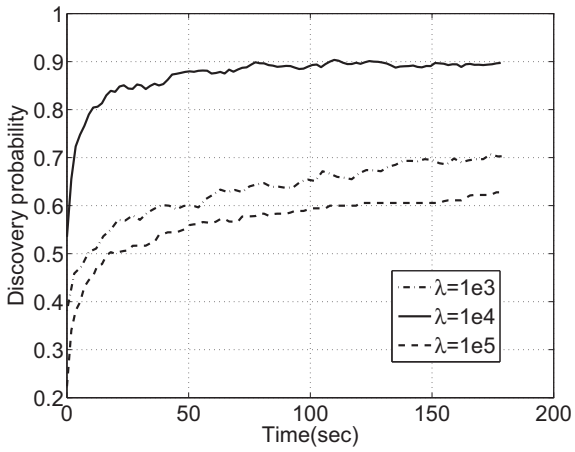

Figure 4.3: The discovery probability in *Algorithm I* with different λ.

We once again perform extensive simulations to evaluate our approach for the general scenarios. The OPNET simulation settings are listed in Tables 4.1 and 4.2. The two kinds of packets exchanged in the communication process are labeled ND_request and ND_feedback (which contains self ID, the request's host-node ID and its chosen transmitting direction ID), respectively. The time interval for which a node should

wait prior to sending packets should be appropriately set to optimize the performance. Shorter waiting intervals cause higher packet volume and consequently, higher interference levels; longer waiting intervals trade temporal efficiency for lower interference levels. We seek to find the value that provides the best trade-off between the two. This value depends on the number of nodes and number of directions used for transmissions by a node. In Fig. 4.3, the cases with different λs (using which the waiting time is characterized) are compared. We see that $\lambda = 10^4$ provides the best performance; it is the sweet spot and yields the fastest discovery speed. We observe that when $\lambda = 10^3$ or $\lambda = 10^5$, the discovery is slower.

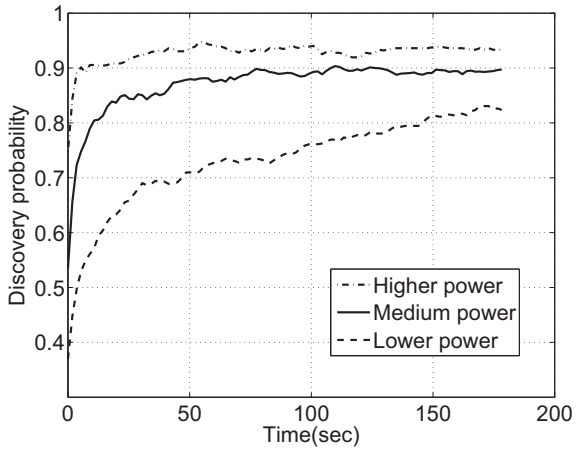

Figure 4.4: The discovery probability in *Algorithm I* with different feedback power.

As mentioned in Section 4.2.1, omni-directional feedback packets increase the interference levels. One method suggested was to lower the power levels at each LED for such feedback packets; however, if the power is reduced to a large extent, the signal quality is compromised. We compare the performance of our algorithm with different power settings for sending the feedback packets. In our first setting, the transmission power of each of the LEDs is equal to the transmission power used for request packets.

This case corresponds to the medium power case in Fig. 4.4. In our second setting we make the total transmission power of all facets (together) to be equal to the transmission power of request packets. This setting is represented by the lower power case in Fig. 4.4. Finally in the third setting, our higher power case corresponds to a case where transmission power on each facet is six times of transmission power of request packets. It appears that the higher power case gives the best performance. This is because the algorithm heavily relies on the number of received feedback packets. As discussed earlier in Chapter 3, the *credit table* tries to capture the number of received request packets. However, the collisions of feedback packets reduce the accuracy of the table entries. Increasing the successful reception rate of feedback packets by using higher transmission powers seemingly assists the discovery probability to approach 1. Motivated by this observation, we next propose an improvement to our algorithm.

4.2.3 Algorithm improvement

4.2.3.1 Algorithm II

We refer to our original algorithm as *Algorithm I* and the new improved version we propose below is named as *Algorithm II*. As more credits are collected, the fraction of discovered directions grows to 1 gradually. To accelerate this process, we have to collect credits as fast as possible. The number of credits a node collects is equal to the number of feedback packets it successfully receives. This is in turn determined by the number of feedback packets sent to the node and the successful reception rate for these feedback packets. The number of feedback packets sent by a neighbor is less than or equal to the number of request received from the node by that neighbor. In the handshaking protocol described above, we used feedback packets to notify the node how

many request packets that it sent have been received by a neighbor. However, due to the loss of feedback packets due to collisions, the number of received request packets by a neighbor is underestimated by the originating node. A solution is proposed here to make credit collection more efficient: when a node (say B) sends a feedback packet to node A as a response to the reception from one of A's directions (say direction 1), it indicates number of request packets that it has received so far from A, sent in direction 1. If A successfully decodes the information in the feedback packet, it can update its *credit table* correspondingly. As discussed before, the number of feedback packets being received from a neighbor is always smaller than the number of request packets that were successfully received by that neighbor. The credits collected with the new protocol will provide much more accurate information and allows the credit table entries to grow faster with increased accuracy than with *Algorithm I*. This decreases the importance of the reception of feedback packets towards accurately updating the credit table. Thus, we can now change the omni-directional feedback transmissions to directional transmissions to decrease interference levels. The direction could be chosen randomly.

The steps of the improved algorithms are described by the pseudocode below:

K = total no. of directions;

Clear credit_table and req_cnt;

Transmitter:

if timer is up **then**

if feedback.head=0|| send feedback 3 times consecutively **then**

ND_request.source_ID=self_id;

ND_request.direc_ID=uniformly_choose(K);

```
    send ND_request;
    set timer;
  else
    ND_feedback.source_ID=self_id;
    ND_feedback.dest_ID=feedback.head→neighbor_ID;
    ND_feedback.prev_direc=feedback.head→prev_direc;

    //Algorithm II:
    ND_feedback.direc_ID=uniformly_choose(K);
    //Algorithm III:
    ND_feedback.direc_ID=Max(credit_table[ND_feedback.
        dest_ID].direc[ND_feedback.prev_direc]);

    ND_feedback.request_count=req_cnt[ND_feedback.
        dest_ID][ND_feedback.prev_direc];
    send ND_feedback;
    set timer;
  end if
end if
Receiver:
if receive ND_request then
  neighbor_ID=ND_request.source_ID;
  prev_direc=ND_request.direc_ID;
  req_cnt[neighbor_ID][prev_direc]++;
else
```

if receive ND_feedback **then**

neighbor_ID=ND_feedback.source_ID;

prev_direc=ND_feedback.direc_ID;

cnt=ND_feedback.request_count;

credit_table.node[neighbor_ID].direction[prev_direc]=cnt;

end if

end if

Compared with ND_feedback packet used in *Algorithm I*, the ND_feedback packet used here has been added in one entry called request_count which records the accumulative counts of received requests from that combination of node and direction. Figure 4.5 shows the variations in the performance of *Algorithm II* with different λ. Again, due to the phenomenon discussed earlier with our initial approach in Fig. 4.3, the case of $\lambda = 10^4$ provides the best performance. In Fig. 4.6, the neighbor discovery performances with different transmission powers employed for feedback packets, are compared. The medium power is six times the lower power and the higher power is six times the medium power, i.e., the configuration is the same as that used to generate Fig. 4.4. Although the higher power case still shows the best performance, the performance difference between the different powers is not as high as with *Algorithm II*. This is because *Algorithm II* does not heavily rely on the number of received feedback packets.

4.2.3.2 Algorithm III

One can easily envision an extension wherein feedback packets can be sent based on the information already available in the *credit table*. If node B needs to respond to node A's request, it checks its *credit table* to determine the current *best*

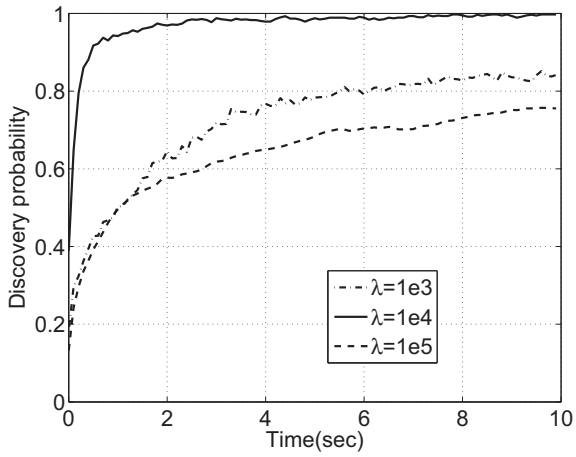

Figure 4.5: The discovery probability in *Algorithm II* with different λ.

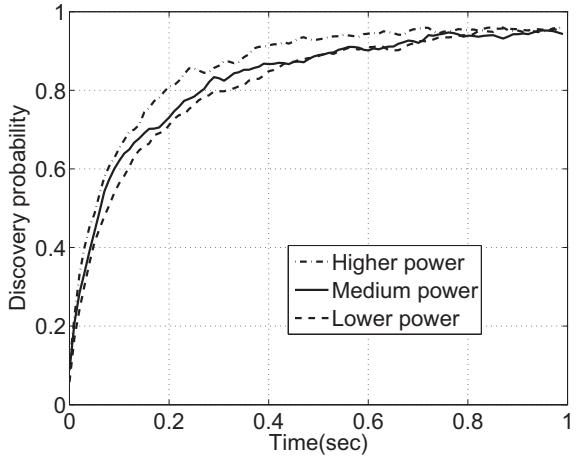

Figure 4.6: The discovery probability in *Algorithm II* with different feedback power.

direction to A and transmits in the corresponding direction.

Our simulations however suggest that this method does not perform as well as *Algorithm II*. In Fig. 4.7 we depict the time taken for neighbor discovery with the three algorithms that we have considered (we set $\lambda = 10^4$ in these simulation experiments). With *Algorithm III* we find that feedback packets experience an increase in collisions from transmissions from other neighbors since in many cases they end up choosing the same direction (the best direction to a common requesting node); this especially hurts

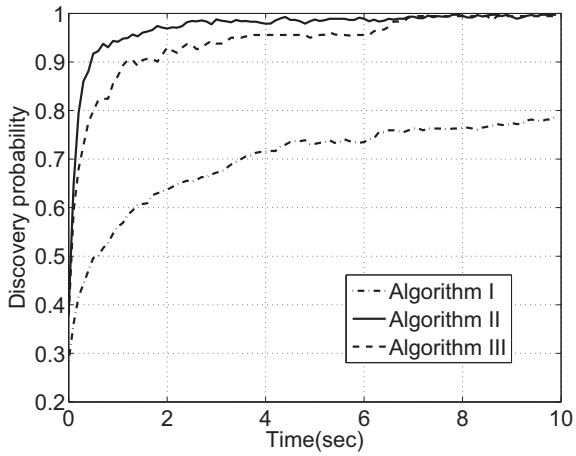

Figure 4.7: Comparison of *Algorithm I*, *Algorithm II* and *Algorithm III*.

neighbors that are further away than those that are closer to the sender. As an example, consider a case where node A sends a request packet in direction 1. Two neighbors, B and C receive the packet and reply to A with their own feedback packets. Although B and C will set timers to wait for an exponentially distributed period, the sending times of the two feedback packets may be very close if $1/\lambda$ is not large compared to τ. When both nodes transmit to A in similar directions, a collision occurs. If one of these nodes (say B) is much closer to A than C, with high probability, B's feedback goes through while C's does not. We like to emphasize that the above effect is a direct artifact of both B and C choosing the same (the best) direction to respond to A. Although the performance of *Algorithm III* is affected due to this (slower time for neighbor discovery), the long term performance is still as good as that of *Algorithm II*.

4.3 Summary

In this chapter, we propose protocols for neighbor discovery in a UV ad hoc network. Due to the unique propagation properties of UV signal, each node may transmit or receive signals through NLOS links in different directions. A node should know

the exact direction in which to transmit to its neighbors for best performance. Towards achieving this, our approach collects "credits" to rank all possible directions. We begin with basic algorithms with and without direction synchronization and then propose various techniques that can significantly enhance the performance of the neighbor discovery process. We perform extensive simulations to not only quantify the performance of our algorithms, but also showcase the effects of various parameters such as the transmission power in use, and the random back-off times that node follow prior to sending neighbor discovery request or feedback packets.

Chapter 5

Leadership based neighbor discovery method for UVOC

As may be evident from the previous description in Chapter 4, a node needs to collect a sufficiently large number of credits in order to accurately rank the directions towards a given neighbor. Since the primary mode of communication is random access, collisions occur. This can skew the results small sample sets may not yield accurate estimates of the best direction. In other words, this results in short term inaccuracies. Convergence to correct results takes a relatively long period of time.

Given this limitation, and the importance of expeditious neighbor discovery, we seek to design an interference-free environment for the process. Our goal here is to enable the discovery of neighbors and the best direction for communications with each neighbor with a *one-time packet exchange*. We later show that our method (proposed below) drastically improves the performance of neighbor discovery in terms of the time taken [41], compared to the method proposed in Chapter 4.

5.1 Our proposed approach:

In a nutshell, our proposed approach relies on sequential neighbor discovery. A single node (the leader) initially performs neighbor discovery; after that node is done, the leadership is passed on to a second node and so no. We call our approach the "leadership-based neighbor discovery" protocol. In more detail, our approach consists of the following steps:

- Initially, all the nodes are in standby status. They only receive (listen) packets but do not transmit anything.
- A pre-chosen leader node (could be randomly chosen) will then initiiate the neighbor discovery process by sending a request packet in a randomly chosen direction. This request packet includes the ID of the sender and the direction in which it is transmitted.
- The leader waits for a fixed time period T, and then, switches to the next direction to send next request packet. It repeats the process until it performs packet transmissions in all possible directions.
- When a neighbor node receives a request packet from the leader node, it calculates the path loss experienced. Note that this is possible because there is no other concurrent transmission. (This is a key difference from the method in Chapter 4; the path loss cannot be obtained since transmissions are subject to interference.)
- Each such neighbor waits for a period t; this period is randomly chosen in $[0, T]$. After this period, the neighbor randomly chooses a direction using which it transmits a feedback packet back to the leader. The feedback packet contains the ID of responding node and the path loss estimated using the request packet. The re-

ception of the feedback packets is not guaranteed because contention exists among different neighbors that attempt sending feedback. However since contention is restricted to only the neighbors of the leader, it is likely to receive many of these feedback packets with a high probability.

- After the process is complete, the leader node can rank the directions for each discovered neighbor found based on the path loss information contained in feedback packets. It then randomly selects a successor from the nodes in its neighbor list. It sends a notifier packet to the selected successor node, using the best direction recorded for this node. The successor then performs neighbor discovery. All other nodes are made aware, that the neighbor discovery *token* has now been passed on to a different node.

It is possible that the notifier packet gets lost; to account for this, if no request packets are heard from the successor for for a period $2T$, the former leader node will re-select a new successor. The process may have to be repeated in extreme cases until a successful leadership transfer is achieved. The steps described above are formalized with the pseudo code below:

K = total no. of directions;

Leader:

while rotation not finished yet **do**

if time T is up **then**

ND_request.source_ID=self_id;

ND_request.direc_ID=uniformly_choose(K);

send ND_request;

set T;

```
    else
        if receive ND_feedback then
            neighbor_ID=ND_feedback.source_ID;
            prev_direc=ND_feedback.prev_direc;
            PathLoss=ND_feedback.path_loss;
            PathLoss.node[neighbor_ID].direc[prev_direc]=PathLoss;
        end if
    end if
end while
while i < # of neighbors do
    order PathLoss.node[i].direc[*];
end while
select leader=uniformly_choose(# of neighbors);
send Notifier packet to next leader;
Other nodes:
if receive ND_request then
    dest_ID=ND_request.source_ID;
    prev_direc=ND_request.direc_ID;
    PathLoss=Tx_power/Rx_power;
    set t=uniformaly_choose(T);
    if time t is up then
        ND_feedback.source_ID=self_id;
        ND_feedback.prev_direc=prev_direc;
        ND_feedback.path_loss=PathLoss;
        send ND_feedback;
```

end if

end if

5.2 Evaluations of our approach:

Table 5.1: Parameters settings in simulations

Transmission power	4 mW
Number of directions	6
Transmission pointing angle	10°
SIR threshold	1 dB
Collision model	Physical (accumulative) model
Traffic pattern	10000 pkt/sec
Request packet size	32 bits/pkt
Feedback packet size	80 bits/pkt
Notifier packet size	32 bits/pkt
Network size	100m by 100m
Number of nodes	10

We evaluate the performance of our neighbor discovery protocol via extensive simulations performed using OPNET version 16.0 [39]. The simulation settings are listed in Table 5.1. The chosen transmission power corresponds to the UV transmission range (approx 50m in Fig. 5.1). In all of the following simulations we assume that the transceiver has 6 directions for transmission operations. The more the directions, the larger the neighbor table maintained and the longer the neighbor discovery process

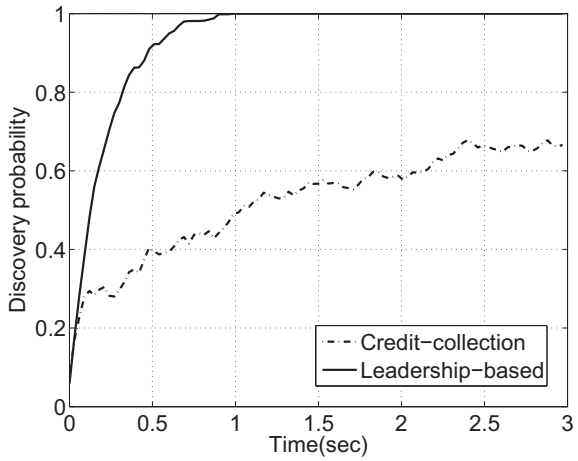

Figure 5.1: Comparison between leadership-based algorithm and credit-collection algorithm proposed in Chapter 4.

takes. The relation between the number of directions and the time consumed by the neighbor discovery process is discussed later. We adopt the channel attenuation results from Chapter 2 to characterize signal propagation. By default, the chosen parameters correspond to the scenario when the transmission pointing angle is equal to 10°. To recall, these parameters are: $\xi = 5 \times 10^9$, $\alpha = 0.4$, $b = 8.7$. The SIR threshold is set to 1dB; the signal is detectable if the power is larger than the total interference power by this value.

The sizes chosen for the three kinds of packets are consistent with that with IEEE 802.11[14]; we choose this since protocols for UV networks are yet to be standardized. Each node is placed randomly in a network area of $1 \times 10^4 \mathrm{m}^2$. The total number of nodes in the network is 10. All the simulation curves are computed as averages of over 10 runs.

We compare the newly proposed protocol (labeled leadership-based) with the previously proposed protocol from Chapter 4 (labeled credit-collection) in Fig. 5.1. We wish to point out here that this comparison is with the fastest version of the algorithm proposed in Chapter 4. The expected value of the waiting interval λ, for credit-collection

is set to be 10^{-3}s. The uniformly distributed waiting interval T with the proposed protocol is set to be 2×10^{-3}s; this ensures that mean is 10^{-3}s and thus, the settings are consistent in both cases. In Fig 5.1, we see the sharp speed up in the neighbor discovery process with the leadership-based approach. The leadership-based algorithm saves about 92% of the time consumed in neighbor discovery, if one were to have a requirement that every node find 70% of its neighbors in the network. It saves about 97% of the time consumed in neighbor discovery if this requirement is stricter and every node has to find about 80% of its neighbors. The credit collection protocol consumes a lot of time to stabilize. These results suggest that interference has a dominant influence on neighbor discovery in UV networks; using a relatively interference-free approach drastically speeds up the neighbor discovery process.

5.3 Improving fairness

Although not explicitly stated earlier, we wish to point out here that a node can assume the role of the leader multiple times. We expect that the neighbor discovery process is a background process that is continuously executed; this is essential since the neighborhoods of nodes could change either due to failures, environmental changes (obstacles causing links to fail) or mobility. In this section, we consider the fairness of our approach in terms of the number of times a node gets an opportunity to being the leader.

To examine the fairness of our approach, we perform a simulation experiment wherein, we run the discovery process until all nodes discover almost all of their neighbors (¿ 99 %). The total simulation time was 3 seconds and node '0' was arbitrarily chosen to be the initial leader. Table 5.2 shows the number of times that each node is chosen to be

Table 5.2: Records of being the leader (time length = 3 secs)

Find 100% neighbors					
Node ID	0	1	2	3	4
Leader counts	22	29	14	13	18
Node ID	5	6	7	8	9
Leader counts	21	22	18	12	25

the leader (referred to as *leader records*) . We immediately see that certain nodes have a better chance at neighbor discovery; in other words, an imbalance is observed. Node 1 is chosen to be the leader 29 times; however, node 8 is only chosen 12 times. Table 5.3 shows the leader records up to the first 0.54 seconds of the simulation. In this period, the percentage of neighbors discovered is on average, 95% and 7 of the nodes had found all of their neighbors. We observe that node 0 and node 4 assumed the role of the leader 6 times each; many other nodes had the opportunity only twice. We also indicate in this table, the percentage of neighbors discovered within this time. We observe that node 8 has discovered all of its neighbors even though it has been a leader just twice. Most nodes find all of their neighbors as long as they become a leader more than twice. This observation is encouraging in that, a node only has to serve as a leader a few times, in order for it to find its neighbors. If a node is chosen as a leader too often, it is likely to waste time and resources. Thus, it appears meaningful to provide equal opportunities of being a leader to all nodes in the network.

Table 5.3: Records of being the leader (time length = 0.54 secs)

Node ID	0	1	2	3	4
Leader counts	6	4	2	3	6
Neighbor percentage	1	1	0.75	0.89	1
Node ID	5	6	7	8	9
Leader counts	5	3	2	2	5
Neighbor percentage	1	1	0.81	1	1

5.3.1 A modified approach to improve fairness

In order to evenly spread the leadership duties, intuitively it would seem that nodes need to have a *count* of how many times each has served as a leader. Then, the leadership role can be passed onto the neighbor who has served the least number of times. This will ensure that the number of times that each node serves as the leader will be roughly equivalent.

To achieve this goal, we incorporate what called *a leader table* in the notifier packet. This enlarges the size of notifier packets compared what was used earlier (as in Table 5.1). Specifically, the size of the packet will increase by $(\#of\ nodes \times 8bits)/packet$. The first leader node (also called trigger node), say node 0, will create this table and mark itself as being the leader once. Once the notifier packet is received by the next leader node (say node 1), a new entry is added into the table to record that node 1 served as the leader once. This leader table is updated at each leadership transfer instance. In this way, we ensure that the current leader node possesses the most up-to-date leader table. Each leader node will choose the next leader that with the least *experience* of being the leader. If multiple nodes are qualified to be the next leader, one

of them is randomly chosen. The pseudo below describes the process in detail.

```
K = total no. of directions;
Leader:
extract Leader_table from Notifier packet;
while rotation not finished yet do
  if time T is up then
    ND_request.source_ID=self_id;
    ND_request.direc_ID=uniformly_choose(K);
    send ND_request;
    set T;
  else
    if receive ND_feedback then
      neighbor_ID=ND_feedback.source_ID;
      prev_direc=ND_feedback.prev_direc;
      PathLoss=ND_feedback.path_loss;
      PathLoss.node[neighbor_ID].direc[prev_direc]=PathLoss;
    end if
  end if
end while
while i < # of neighbors do
  order PathLoss.node[i].direc[*];
end while
select next_leader=Min(Leader_table[neighbor_ID]);
send Notifier packet to next_leader;
```

Other nodes:

if receive ND_request **then**

neighbor_ID=ND_request.source_ID;

prev_direc=ND_request.direc_ID;

ND_feedback.path_loss=Tx_power/Rx_power;

t=uniformly_choose(T);

if t is up **then**

ND_feedback.source_ID=self_id;

ND_feedback.prev_direc=prev_direc;

send ND_feedback;

end if

end if

5.3.2 Evaluating the modified approach

Table 5.4: Records of being the leader (time length = 3 secs)

Find 100% neighbors					
Node ID	0	1	2	3	4
Leader counts	20	19	20	20	18
Node ID	5	6	7	8	9
Leader counts	19	19	19	20	20

Next, we present evaluations of our modified approach. Tables 5.4 and 5.5 show the number of times that each node in the network serves as a leader, when the overall percentage of neighbors discovered is approximately 100% and 95%, respectively.

Table 5.5: Records of being the leader (time length = 0.42 secs)

Node ID	0	1	2	3	4
Leader counts	3	3	3	3	3
Neighbor percentage	0.89	1	1	1	0.94
Node ID	5	6	7	8	9
Leader counts	3	3	3	3	3
Neighbor percentage	1	1	1	0.71	1

The results are from one randomly chosen realization. We observe that the variance in the counts (in Table 5.4) is relatively small; specifically, it is 0.44 compared to 26.84 in Table 5.2 with the original scheme. We see from Table 5.5 that all the nodes have been the leader, the same number of times; this means that the approach is fair. Both Table 5.3 and Table 5.5 reflect scenarios where each node found approximately 95% of its neighbors, successfully. However, the time incurred with the latter algorithm is only 0.42s; this represents a 22% reduction in time compared to the original approach.

This reduction is a direct artifact of the increased efficiency with the modified scheme; nodes that have previously served as leaders and have found their neighbors do not waste time in serving as leaders again.

Figure 5.2 depicts how the modified scheme improves the performance of neighbor discovery. We choose a 20 node scenario and the area of deployment is still 100m×100m. The maximum transmission range and transmission power are 50 m and 0.4 mW, respectively; these are the same as in Table 5.1. From Fig. 5.2, we see that the time needed by every node for finding 95% its neighbors on average, with the basic leadership-based algorithm is about 1.62 seconds; this time is reduced to about 0.9 sec-

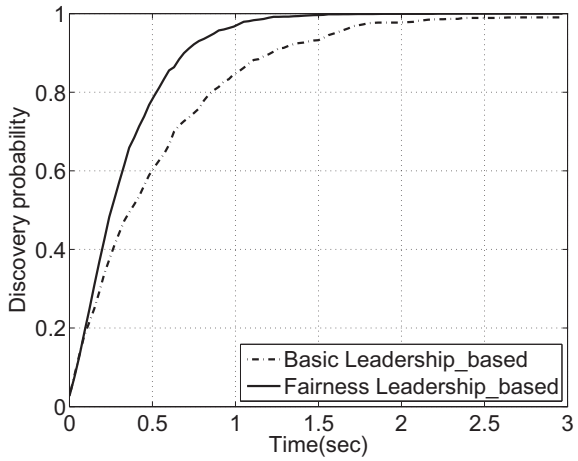

Figure 5.2: Comparison between schemes with or without fairness consideration.

onds with modified fair scheme. The modified scheme brings about a 45% performance improvement *after* taking into account the increase in packet transmission time due to the enlarging of the notifier packet.

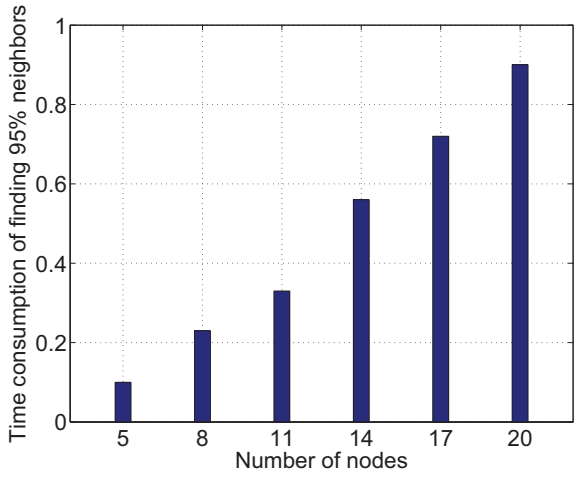

Figure 5.3: The time taken for neighbor discovery varies with number of nodes. The time shown is when 95% of neighbors are discovered.

Finally, we examine how the time for neighbor discovery varies as we vary the number of nodes in the network. Fig. 5.3 shows how the time required to discover 95% of the neighbors changes (on average), as we vary the number of nodes in the network. We

retain the coverage area to be 100m×100m for each simulation point represented in the figure. The notifier packet length is also set to be identical to that in the scenario with 20 nodes; we do this to isolate the impact of node density from the incurred overhead changes from the notifier packets. Fig. 5.3 shows that the more the nodes in the network, the longer it takes for neighbor discovery as one might expect. More importantly, one can conclude that the time consumption is *proportional* to the number of nodes. The relationship is almost linear as seen in the figure.

As described earlier, our leadership-based algorithm attempts to construct an interference-free environment for the leader node to find its neighbors. However, with an increase in node numbers or network size, the time taken for neighbor discovery will increase. We further increase the node density (see Fig. 5.4) and we find that the time taken grows almost linearly versus the the number of nodes. For very dense networks, as compared with credit-collection algorithm, we found that the savings in terms of time get larger with increased node density. We use curve fitting method to obtain extrapolated results for the leadership-based (solid line) and the credit-collection (dotted line) protocols, respectively. The curve fitting expression for the solid line is $0.4264 \times \exp(0.02799n)$ and the expression for the dotted line is $0.4937 \times \exp(0.03880n)$, where n stands for the number of nodes in the network. The slopes for both curves are obtained as $0.4264 \times 0.02799 \times \exp(0.02799n)$ and $0.4937 \times 0.03880 \times \exp(0.03880n)$, respectively (by taking the first derivative with respect to n). For a fixed value of n, the slope of the curve for the credit-collection protocol is always larger than the one for the newly proposed leadership-based protocols. Thus, one might expect the difference between two curves to grow when the number of nodes increases; this can be observed in Fig. 5.4.

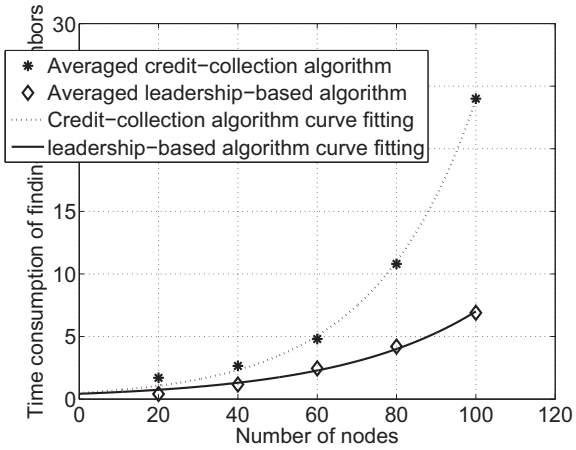

Figure 5.4: The time taken for neighbor discovery varies with number of nodes – a comparison of the leadership-based algorithm and the credit-collection algorithm. The time shown is when 70% of the neighbors are discovered, on average.

In order to cover all the directions seamlessly, the number of transmitters installed should be inversely proportional to the transmission beam divergence of each Tx. For example, if the beam divergence is 60^o, the number of transmitters should at least be 6 to cover all the directions. With more number of transmitters installed, the transmission beam divergence of each Tx can be reduced. Assuming that the transmission power is kept the same, small beam divergence results in longer transmission range and lower interference to other nodes. However, a large number of transmitters also implies that the time necessary for the neighbor discovery increases because the nodes have to search and rank more directions. Fig. 5.5 shows the relationship between the time required and the number of transmitters. Each direction corresponds to one transmitter individually; for e.g. the plot corresponding to 15 directions represents the scenario where the transceiver is equipped with 15 facets and 15 transmitters. From this figure, we conclude that the more directions, the longer the neighbor discovery process. For the four cases compared in Fig. 5.5, in order to find 95% of neighbors on average, it takes 0.9 secs, 1.4 secs, 1.92 secs and 2.39 secs, for the cases with 6 directions, 9

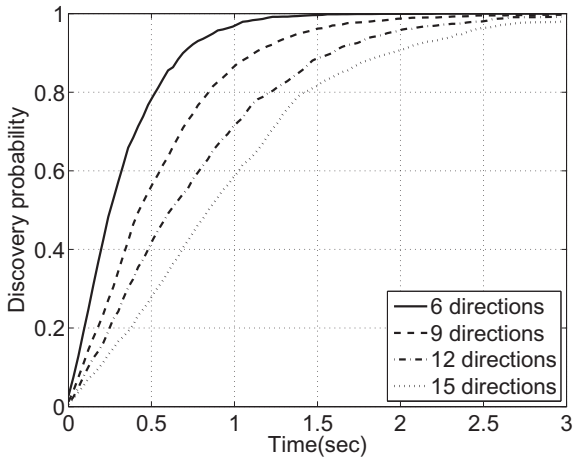

Figure 5.5: Performance with different numbers of directions. 20 nodes in total are distributed in the network.

directions, 12 directions, and 15 directions, respectively .

5.4 Terminating neighbor discovery

In most application scenarios, it is important to bootstrap the network within a relatively short period. It is unlikely that a deployer will be able to pre-configure nodes with the specific topology; moreover, link qualities are likely to vary temporally. Our goal here is to determine when a neighbor discovery process can be considered to have terminated.

5.4.1 The scheme for terminating neighbor discovery

Achieving the above goal is far from trivial. Due to multi-user interference, feedback packets may collide leading to information loss. As a consequence a leader node is unlikely to realize all its neighborhood information by simply serving as the leader once. Thus, as discussed earlier, a node assumes the role of the leader multiple times. At each instance, it updates its neighborhood information with any data that is newly collected. If a node is unable to obtain new information upon becoming the leader, it is

likely that it has already obtained its complete neighborhood information. In fact, the more times this happens, the more confidence one has in this conclusion. Thus, we make a modified rule that if a node does not acquire new information for a certain consecutive number of instances when it serves as a leader, it assumes that it does not need to be a leader anymore. It *marks* itself as being done with neighbor discovery. If all nodes have marked themselves as being done, then the procedure terminates. The information on which of the nodes are marked as above is recorded in a termination table. This table is exchanged in the notifier packets; thus, it is globally disseminated. We discuss this table later.

Each node maintains a counter which is initialized to '0'. Whenever it has acted as a leader but has not obtained any new neighborhood information, it increments this counter. If it does obtain new information, the counter is reset. When the counter values reaches a pre-defined threshold N, the node marks itself. When it transfers leadership, it updates the termination table to indicate that it is done with neighbor discovery. If the total number of nodes and their IDs are known a priori (which is likely), a simple bit corresponding to the nodes' ID can be set to indicate that it is done. In other words, the termination table can simply be a bitmap which is equal in size to the number of nodes in the network.

With the information in the termination table, the current leader has the most up-to-date information and can make a timely and efficient decision on which node to choose as the leader next. Note here that this can be combined with the fairness criterion described earlier; however, we use this as the explicit criteria in the remainder of the section.

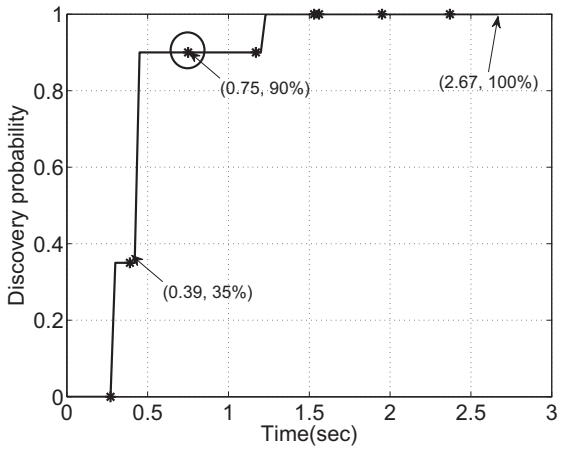

Figure 5.6: Illustrating the termination process. Solid line shows how the percentage of discovered neighbors increases with time; *s represent the instances when the node starts serving as the leader.

5.4.2 Performance evaluation

Figure 5.6 shows the percentage of neighbors discovered with time for a single realization from the perspective of an arbitrarily chosen node in the network. We simulate the 20 node scenario and other simulation settings are the same as described in Section 5.3. In this figure, we see that the node under consideration found 35% of its neighbors after the first instance of being the leader; this percentage increased to 75% after its second instance of being the leader. However, as shown by the circle in the plot, after the node served as the leader for the third time, there was no new information obtained; note however that, the neighbor discovery was not complete for the node. Although the counter increased by 1, it was reset after the node served as a leader for the fourth time, since new information was acquired. The neighbor discovery procedure finally ended at 2.67 seconds because the counter threshold was reached. At that point, the node had indeed discovered all of its neighbors.

Table 5.6 shows the total leader counts *after* every node found 100% of its neighbors. We set the termination counter threshold mentioned above to be 2. Inter-

Table 5.6: Leader records after finding 100% neighbors

Node ID	0	1	2	3	4
Leader counts	6	4	5	7	4
Node ID	5	6	7	8	9
Leader counts	5	4	5	6	6
Node ID	10	11	12	13	14
Leader counts	5	6	6	4	2
Node ID	15	16	17	18	19
Leader counts	7	7	7	6	7

estingly, from the Table 5.6 we observe that all the nodes except the one with node ID of 14 served as the leader without any information update, more than 2 times. This is because, node 14 was the final node in the neighbor discovery process; unless its counter was set to 2, the process could not be terminated. However, node 14 could be assigned to being a leader only by one of its neighbors. In this case, nodes attempted to search for 14 and interim also served as leaders again (the leadership token was passed around). Finally, the token was passed onto node 14 and the process terminated.

Figure 5.7 shows the performance of the protocol with the termination procedure with six scenarios, with different node densities. We see the termination time is proportional to the node density as one may expect.

5.5 Summary

Neighbor discovery in UV networks is an important function that allows nodes to bootstrap the topology and begin operations. While there have been a plethora

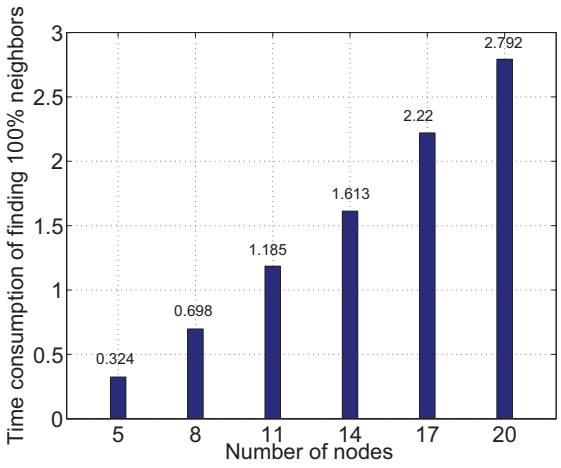

Figure 5.7: Illustration of the performance of the termination process with different node densities. All the nodes in the network find 100% of their neighbors.

of proposals for neighbor discovery in the RF context, they are inapplicable in the UV context given its unique PHY characteristics. In this chapter, we propose a novel neighbor discovery protocol for UV networks. The key idea is to eliminate interference by arbitrating the discovery process by using leaders; a leader is the only nodes that are allowed to perform neighbor discovery at any given time. The leadership role is passed on from node to node. We demonstrate that our protocol is extremely effective in performing neighbor discovery via extensive simulations that incorporate the UV PHY layer. In particular, the time needed for neighbor discovery process is decreased by as much as 90% as compared to the only previously proposed neighbor discovery scheme for UV networks. We also provide extensive evaluations to capture the impact of various network and system parameters on the performance of our protocol.

Chapter 6

VICO: A framework for configuring indoor visible light communication networks

In this chapter, we first perform extensive channel measurements and verifications in the visible light band in an indoor scenario on a real PHY layer testbed. Based on the experiments, we make the following observations: in order to increase data rate **(1)** the transmission beamwidth can be shrunk, and/or **(2)** the transmission beam can be tuned to point to target area. Based on these observations, we simulate the illumination variations using a Matlab simulation tool. With this we validate that the human eye is able to accommodate the changes in intensity brought about by shrinking the beam or by tuning the beam direction.

After verifying the feasibility of the two degrees of freedom (variable beamwidth and orientation of a transmitting LED), we develop a novel configuration framework (VICO for VIsible light COnfiguration framework) for an indoor VLC network. VICO

schedules transmissions, and appropriately chooses the beamwidth and LED orientation, while taking into account the indoor VLC PHY characteristics [42].

Organization: We present our PHY layer experiments and simulations and analyze them in Section 6.1. Our configuration framework, VICO is described in Section 6.2. Our extensive evaluations of VICO via simulations, are presented in Section 6.3. We conclude our work in Section 6.4.

6.1 Physical layer characteristics of VLC

We perform extensive channel measurements and verifications on a PHY layer testbed in indoor scenarios. As alluded earlier, our main findings are: **(1)** the beamwidth can be adjusted and it is inversely proportional to achievable data rate; **(2)** the transmission beam can be tuned to directionally (not vertically) illuminate an adjacent area; the path loss is proportional to the beam-angle[1]. Following our measurements we perform simulations to examine the illumination variations when we change beamwidth or beam-angle. We find that these variations do not hamper visibility to the human eye.

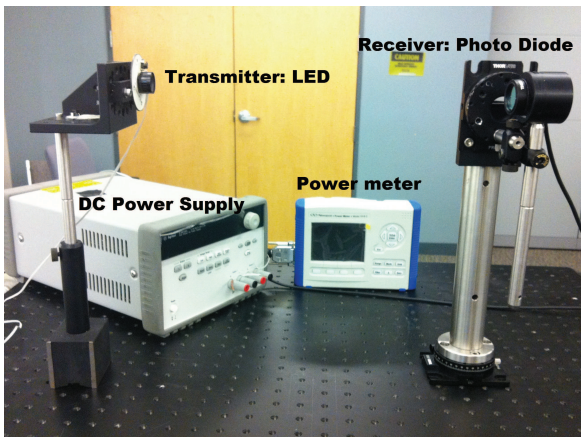

Figure 6.1: Visible light experimental testbed.

[1]We define beam-angle as the angle between the axis of transmission beam and the vertical axis.

Experimental setup: In our experiments, we employ a single LED as the transmitter, a filter, and a photodiode as the receiver as shown in Fig. 6.1. At the transmitter, we use a 1 Watt light spot on "OptoTherm Linear Heat Spreader with Lens" white LED (OPA742W23). The illumination output of the LED is 41lm with a driving current of 350mA and a full beam angle of 23°. The relative radiant power spectral density and the spatial intensity distribution curves of each LED are available in the data-sheets [43]. On the receiver side, a commercial photodiode (PDA10A) with an internal trans-impedance preamplifier is used. It has an active area of 0.8mm^2 and an electrical signal bandwidth of 150 MHz [44]. An optical bandpass filter (Thorlabs FB450-40) is added in front of the photodiode. The filter has a center wavelength of 450nm, a full width at half maximum (FWHM) of 40nm, and transmittance of 70% [45]. These components are jointly used to expand the active receiving area and reduce the ambient light noise. The receiver acts as an energy detector and can facilitate the use of on-off keying (OOK) or pulse position modulation (PPM).

6.1.1 Beamwidth adjustment

We conduct experiments to understand the relation between the transmission beamwidth and the area of coverage with LOS communications. The area of coverage will decrease if we reduce the beamwidth and the intensity of the shrunk-down area will increase for a fixed transmission power. Since we do not have access to LEDs with adjustable beamwidths, we vary (reduce or increase) the receiving distance to vary the beamwidth (shrink or expand the illumination area). There are two ways to alter the energy intensity: changing beamwidth or varying the distance between the Tx-Rx pair. Consider as in Fig. 6.2, a LED mounted on the ceiling at point T. If the beamwidth

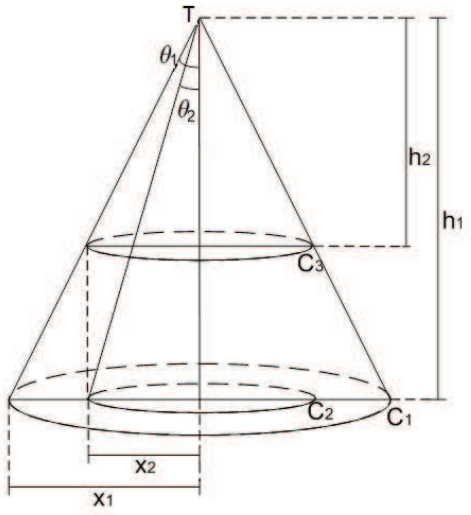

Figure 6.2: The relationship between beamwidth and distance from the LED source.

is θ_1 (θ_i reflects half the beamwidth), the coverage area at h_1 (the distance from the LED) is C_1. If we shrink the beamwidth to θ_2, the coverage area is C_2. The energy is uniformly distributed in the plane when the plane is orthogonal to the transmission beam. The energy intensity is the same on different intersection circles as long as the size of the circle is equivalent. When the area of C_2 and C_3 are the same (C_2 corresponds to beamwidth θ_2 at distance h_1; C_3 corresponds to beamwidth θ_1 at distance h_2; the area of C_2 is identical to the size that of C_3), the energy intensity in C_2 is equivalent to that in C_3; this is true as proved in [46], based on the scaling property of illumination rendering. It was found that the basic illumination pattern at $k \times h$ (k is a constant) is identical to that at h with its beamwidth scaled up by k. This is because with a shrinking beamwidth there is a decrease in path loss; this decrease counterbalances the increase in path loss brought about by the increased distance. The relationships between

the variables in Fig. 6.2 are shown as follows:

$$\begin{aligned} x_1 &= h_1 * tan(\theta_1), \\ x_2 &= h_1 * tan(\theta_2) = h_2 * tan(\theta_1), \\ x_1/x_2 &= tan(\theta_1)/tan(\theta_2). \end{aligned} \tag{6.1}$$

We consider an LED transmission beamwidth specified by $\theta_1 = 18^\circ$ in the measurement settings[2]. Without loss of generality, the term "beamwidth" in the rest of the chapter refers to this half beamwidth angle θ_1. We test three scenarios at distances of 2m, 2.5m and 3m, respectively. The energy intensity is measured by a power meter as: 1.89μW, 1.13μW, and 0.85μW correspondingly. We have $x_1^2/tan^2(\theta_1) = x_2^2/tan^2(\theta_2) = x_3^2/tan^2(\theta_3) = A$, and θ_1=18°. Then, one can derive that θ_2=15.18°, θ_3=12.28°, and $A = 8.91$. For a fixed beamwidth, the farther from the LED, the lower the energy intensity that is achieved. The measurements thus show that the energy intensity is inversely proportional to the area covered [46].

The energy intensity is also directly proportional to the signal-to-noise ratio (SNR) (assuming that the background noise is white Gaussian noise with a constant mean) [3]. Since the data rate is inversely proportional to the SNR, it is proportional to the received energy intensity and inversely proportional to the coverage area size. Assuming that an on-off-keying (OOK) modulation scheme is adopted, we have $R_1/R_2 = E_1/E_2 = x_2^2/x_1^2 = tan^2(\theta_2)/tan^2(\theta_1) = K$, where R_i is the data rate achieved with a coverage area C_i, E_i is the energy or intensity with the same coverage area ($i \in \{1, 2\}$), and K is a constant [46]. If we want to double the data rate, we have to shrink the beamwidth to 12.9° (from 18°). At desk-top height (0.5m)[3], the radius of the coverage

[2]The LED's maximum beamwidth as shown in its data-sheets [43] is 23°. This means that the energy is uniformly distributed within the area covered by this full beamwidth of 23°. However, the beamwidth measured in the experiments suggests total energy loss (i.e., it is totally dark) beyond the edge of the area covered by a beamwidth specified by $\theta_1 = 18^\circ$.

[3]Normally, we assume the room height is 3m; thus the distance from the LED to a node at desk-top

area is 0.8123m when the beamwidth is specified by θ_1 equal to 18°. Therefore, in order to cover a room with 4m×4m, we need at least 9 LEDs with minimal overlaps between each service cell. Adjacent LEDs are separated by a space about 1m. Please note that the specific settings are for OOK modulation and obtained from our experimental testbed in Fig. 6.1.

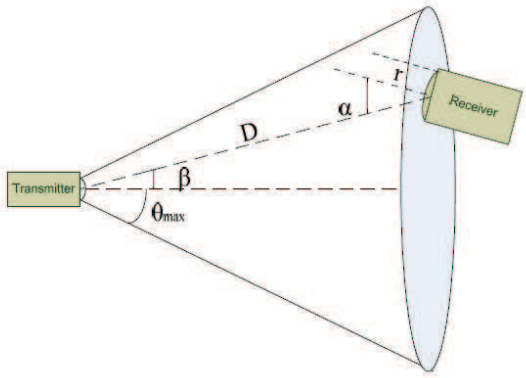

Figure 6.3: LOS channel model (from [3])

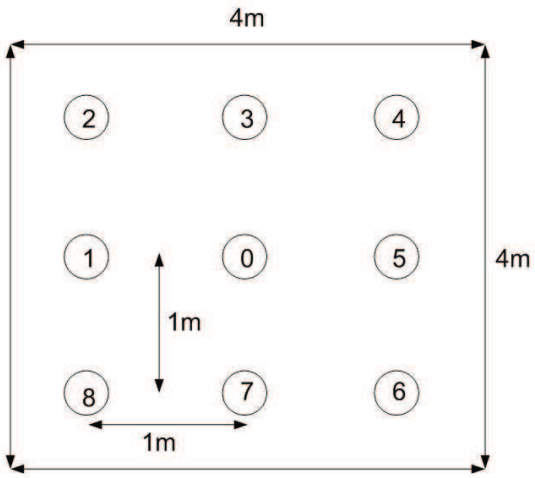

Figure 6.4: LED distributions in a 4m×4m room. Each LED has been labeled; LED 0 is the coordinator.

height is 2.5m.

Matlab simulations: Based on the LOS path loss model in Eq. (6.2) of [3], the illuminance distribution in a 4m×4m room has been simulated using Matlab. In Eq. (6.2), m is the order of Lambertian emission[4] and is related to the semi-angle at the *half illuminance* of an LED $\Phi_{1/2}$; specifically, $m = -ln2/ln(\cos(\Phi_{1/2})$. P_r and P_t denote the received power and the transmitted power, respectively. The angle between the source-receiver line and the receiver orientation (see Fig. 6.3) is θ; the angle between this line and the source beam axis is β (viewing angle). As shown in Fig. 6.3, if θ is larger than the receiver's FOV, the LOS link does not exist and therefore cannot receive from the target LED. θ_{max} is the maximum beamwidth. Ω_r is the Rx solid angle[5] seen by the Tx and A_r is the Rx area. They satisfy the relationship $A_r \cos(\theta) = D^2\Omega_r$. The Rx distance to the source is D. Some of these relationships are illustrated in Fig. 6.3. For more details the reader is referred to [3].

$$Pathloss = \frac{P_r}{P_t} = \frac{(m+1)A_r}{2\pi D^2}\cos(\alpha)\cos^2(\beta). \tag{6.2}$$

The simulation settings are as follows: there are 9 (3×3) LED arrays uniformly mounted on the ceiling. The specific deployments are shown in Fig. 6.4. Each array consists of 100×100 individual LEDs. In the simulation, we set θ_{max}=18°, which is consistent with what was used in our experiments. The receiver's FOV is 60°. The photodiode's receiving area A_r is $1.0mm^2$. Other parameters are set identical to those in [8]. We compute the illumination on the ground using Eq. (6.2), assuming that all the LEDs are facing vertically downwards (towards the ground) and have the same beamwidth. The illuminance distribution is plotted on the left side in Fig. 6.5. If we shrink the middle LED array's beamwidth to 12.9°, the illuminance distribution is shown

[4]Lambertian emitter is a light source whose radiance varies according to Lambert's cosine law.

[5]The solid angle, Ω_r, is the two-dimensional angle in three-dimensional space that an object subtends at a point.

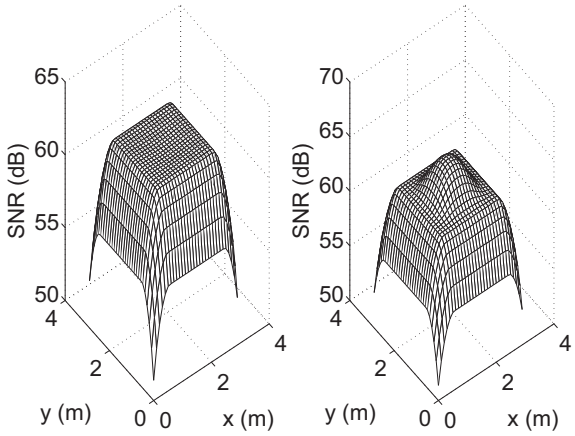

Figure 6.5: Illuminance distribution in a 4m×4m room. Left: all LEDs are facing vertically to the ground with identical beamwidths of 18° (default deployments in which all the LEDs are at default positions); Right: the middle LED array's beamwidth is shrunk to 12.9°.

in the right plot. From the plots, we can see the SNR increase in the middle of the room and the corresponding decrease at the edges. The maximum SNR for left scenario is 63.0062 dB and 66.0712 dB for right one. There is a 3dB difference which is consistent with our experimental derivations.

6.1.2 Beam-angle tuning

Next, we examine if we can tune the pointing direction (beam-angle) of a transmission beam to achieve better performance. For example, in Fig. 6.6, the cell to the bottom left is covered by two LED arrays. If the two LED arrays transmit the same information to the node in this cell simultaneously (assuming OOK), the SNR will increase and so does the achievable data rate. However, if these two LED arrays send different signals at the same time, it would cause severe interference. We have to take these into account in higher layer protocol design.

Consider the case where adjacent LED arrays send the same information. At the target Rx, the time delay between the two paths from two LED arrays depends on

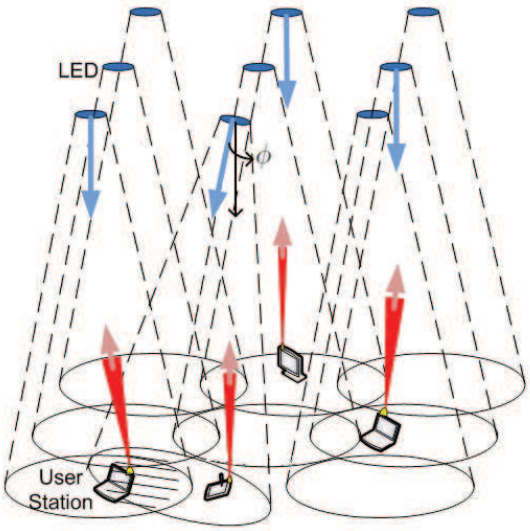

Figure 6.6: An indoor VLC network: One of the 9 LED arrays has been tuned to illuminate the adjacent cell.

the difference in the distances on the two paths. In an indoor VLC system, the distance between two LED arrays is of the order of a few meters; the distance from ceiling to the desk-top height or ground is again of the order of a few meters. Typically the maximum distance difference on the two paths from the pair of LEDs to the user is typically no more than 5 meters and the corresponding time delay is about 1.6×10^{-8}s. Compared to the LED's modulation bandwidth which is 2MHz, the time delay is negligible. Thus, we can ignore the impact of this delay spread with minimum loss in accuracy.

Matlab simulations: We follow similar simulation procedures and use settings as in Subsection 6.1.1. Our goal is to observe the illuminance changes due to the tuning of the beam-angle. In Fig. 6.7, we see that the maximum SNR is increased in the cell towards which the middle LED array is redirected. The SNR variation depends on the extent to which the beam-angle of the LED array is tuned. For example, when the middle LED array's beam-angle is tuned such that the coverage area shifts upwards by 0.1m from the default position (left plot in Fig. 6.7) the maximum SNR increase is

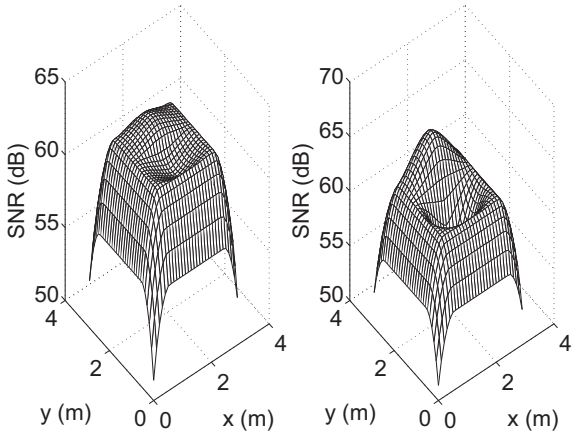

Figure 6.7: Illuminance distribution in a 4m×4m room. The middle LED array is tuned to directionally shift towards the top-middle cell. Left: the center of the coverage of the tuned beam-angle is 0.1m away from default position; Right: 0.5m away from default position.

1dB in the middle cell of the top row. Correspondingly, the SNR in the bottom middle cell decreases by about 2dB. When the middle LED array beam shifts by 0.5m from the original position (right plot in Fig. 6.7), the maximum SNR increase in the top middle cell is 3dB and the decrease in SNR in the bottom middle cell is about 8dB.

6.1.3 Eye sensitivity to the beam adjustments

Table 6.1: Illuminance comparison in a room with beamwidth and beam-angle adjustments

Scenarios	Average SNR (dB)	the minimum SNR (dB) all-over the room
No adjustments (left plot in Fig. 6.5)	61.556	51.3923
Beamwidth adjustment (right plot in Fig. 6.5)	61.9226	51.3923
Beam-angle tuning 1 (left plot in Fig. 6.7)	61.548	51.3923
Beam-angle tuning 2 (right plot in Fig. 6.7)	61.4019	51.3923

The illumination difference brought by beamwidth modifications and beam-

angle tuning directly impacts a human's feeling on the illumination. Since the international LED lighting standards is still under development [47], the sensitivity of human eyes to the illuminance change brought by tuning beamwidth and beam-angle has not been studied extensively. However, it is generally believed that a change of about 1 to 3 dB in the average and minimum SNRs (illumination) is unlikely to significantly affect the perception of a human. We assume that the left plot in Fig. 6.5 which represents the default case (all the lighting sources are distributed uniformly and point vertically to the ground), corresponds to a scenario of acceptable illuminance. Note that in our simulations, the SNR directly corresponds to the illuminance. The maximum SNR in the default case is 63.0062 dB, the average SNR 61.556 dB and the minimum SNR 51.3923 dB. The minimum SNR is observed at the four corners of the room.

6.2 Design of VICO: Our configuration framework

In this section, we present the design of our configuration framework VICO. In a nutshell, the goal of VICO is to schedule transmissions in the VLC network so as to achieve the highest throughput under conditions of max-min fairness. In achieving this goal, VICO attempts to exploit both beamwidth adjustments and beam-angle tuning to opportunistically increase throughput.

In a nutshell, VICO first allows clients to associate with their closest LED array (which are similar to base stations); this allows the LED array to determine the location of the client. The LED array then finds the best beamwidth and beam-angle for each client. The information is then fed to a central controller that then makes the scheduling decisions; it determines which LED arrays can be active simultaneously in a time slot and which clients they serve. Finally, the controller also triggers unused LED arrays to

enhance the signal quality (and hence the rate) at clients that are already being served (send the same information as the default LED array that is already serving the client), if possible. A block diagram depicting the functionalities of VICO is in Fig. 6.8.

Each LED array communicates with a single user (or client) at any given time; note however that clients can receive simultaneously from two arrays if they transmit the same information. If there are multiple users associated with a LED, the transmissions to the users are separated in time (each user is served in a different time-slot). Our goal is to then determine the set of clients, that can be simultaneously served in a single time slot.

We use a *conflict graph* to model the effects of interference. The conflict graph determines the groups of links that mutually interfere and hence, cannot be active simultaneously. Based on the conflict graph, VICO attempts to do the following: (1) create groups of links that are conflict free (these are links that can be simultaneously activated); since groups of links are activated one by one, fairness is ensured (2) maximize the throughput by minimizing the number of groups; (3) opportunistically increase throughput by utilizing inactive LEDs to increase SNR at clients already being served (on a "water-filling" basis).

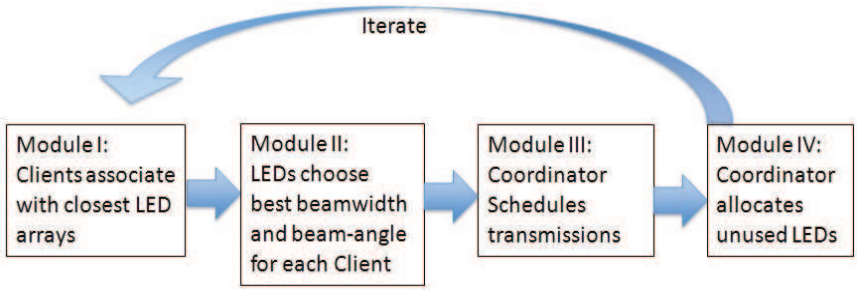

Figure 6.8: A high level overview of VICO.

6.2.1 Formulation of a scheduling problem

In order to avoid conflict, we seek to partition the network into different subgroups. The links in the same group share the medium without interfering with each other. Let us assume that M LED arrays and N clients are in the room. For ease of discussion, we assume that $N > M$ (as typical in a conference room or a hospital lounge); VICO however, is applicable when $N \leq M$. Let x_{ij}=(l_i, c_j) be the link between LED i and client j. Two links, x_{ij} and x_{pq} that interfere with each other cannot be scheduled in the same time slot. Our goal is to create the minimum number of groups, each of which contains a set of non-interfering links. In each time slot, a specific group is activated. If we are able to minimize the number of groups, the maximum throughput under max-min fairness conditions could be achieved. We first show that our problem is NP-hard. Subsequently, we design an efficient heuristic to create the groups.

Theorem 1 *The scheduling problem for our VLC system which has two degrees of freedom (beamwidth and beam-angle of each LED) is NP-hard.*

Proof. We first show that the problem with fixed beamwidths and angles (let us call it Problem 1) is NP-hard. We define the conflict (interference) graph $G = (V, E)$, where the set of vertices V is the set of possible links x_{ij} in the original network graph. An edge $e \in \{E\}$, between two vertices in the interference graph exists if the two links (vertices) interfere with each other. Since the two vertices (links) that are connected with an edge in the interference graph conflict with each other, they need to be scheduled in different time slots and thus be *colored* differently. In other words, we want to color the vertices of graph G in such a way that no two adjacent nodes share the same color. This is the classic vertex coloring problem which is known to be NP-hard [48].

In our general problem, we have two additional degrees of freedom (beamwidth and beam-angle of each LED). This means that we no longer have only one interference graph. Consequently, we need to find a coloring with the minimum chromatic number across all possible interference graphs. Since Problem 1 which is a special case of our problem is NP-hard, we conclude that our problem is also NP-hard. In other words, even if we had a way to find the beamwidths and the orientations of every LED that leads to the "best" throughput, we would not be able to determine the schedule since problem 1 is NP-hard. ■

Given the result in Theorem 1, we next propose a heuristic to group non-conflicting links as efficiently as possible.

6.2.2 LED beaming and orientation

In order to simplify the problem and consider feasibility, we assume that each LED can only have two different beamwidth configurations (denoted as θ_0 and θ_1) and three options for beam orientation (with corresponding beam-angles ϕ_0, ϕ_1, ϕ_2). The beamwidths are 18° and 12.9° respectively, to be consistent with our PHY layer studies. The three different beam orientations can be achieved in four directions (to reach out to possible adjacent neighbor cells). ϕ_0 represents the angle where the LED is facing the ground vertically i.e., $\phi_0 = 0°$. We refer to this pointing position as the original (default) position. The beam-angle ϕ_2 is the offset angle from original position when the coverage area completely overlaps with an adjacent neighbor cell when the latter's beam is set in the original position. The beam-angle ϕ_1 creates a half-way overlap i.e., $\phi_1 = \frac{1}{2}(\phi_2)$.

Applicability: The fixed number of beamwidths and beam-angles are chosen to simplify the description of our solution and our evaluations. Moreover, they also tie

in with our PHY layer studies described in Section 6.1. One would only expect a finite set of beamwidth and beam-angle combinations used since the number of data-rates possible belong to a finite discrete set (based on the modulation and FEC combinations used). VICO is applicable in such more general cases with no modifications. If electronic steering is possible and beamwidths and beam-angles can be tuned to arbitrary values, VICO will still apply.

6.2.3 Details of VICO

The following modules make up our framework VICO. Again, these modules and the sequence of functions in VICO are depicted in Fig. 6.8. Given that the scheduling problem with the optimal beamwidth and beam-angle is NP-hard, we use a heuristic to group the non-conflicting links within the same time slot in the third module.

Module I (Client Association): Each client chooses the nearest LED array to associate with. A LED array may have multiple associated clients. We assume that clients can determine the closest LED based on beacons that are transmitted during a calibration phase. Since we only care about scheduling downlink transmissions in this chapter, it is assumed that an LED array can determine the location of a client with the received signal or from an image sensor [3]. Alternatively, clients may use localization methods (e.g. [49]) to determine their coordinates and this is later communicated to the LED array.

Module II (LED Tuning): Each LED array chooses the best combination of beamwidth and beam-angle from sets $\{\theta_0, \theta_1\}$ and $\{\phi_0, \phi_1, \phi_2\}$ for each associated client. The "best" combination is the one that minimizes path loss. Once an LED array obtains the coordinates of an associated node, it calculates the path loss to the node with different combinations of beamwidth and beam-angle. It then chooses the

combination that yields the lowest attenuation and hence the highest SNR; this in turn allows the LED array to transmit with the highest viable rate to the client. In the case of ties, the default position (ϕ_0) and the smaller beamwidth (θ_0) are used. This is because communications with this combination has the low associated possibility of interfering with neighbor cells.

Module III (Scheduling Transmissions): Next, the information with regards to each client is fed back to a central controller. The controller generates a basic conflict graph with the information provided. This conflict graph $G = (V, E)$ contains vertices that correspond to the links determined in previous module (with the corresponding beamwidth and beam-angle). There is an edge between the vertices x_{ij} and x_{pq} in G if the links x_{ij} and x_{pq} cannot be active simultaneously; i, p denote the LEDs' labels and j, q represent the nodes' labels. To begin with the controller tries to find the group of links S_0, which are to be scheduled in the first slot. Note here, that since each LED can at most serve one client at a time, the maximum value of S_0 is the number of LED arrays in the network. It picks a link from $G = G^{(0)}$ at random (say the link x_{ij}) to be scheduled in this slot i.e., the link is included in S_0. It then picks a second link (not picked before) at random; if this link interferes with the links that are already included in S_0, it is eliminated; else, it is included in S_0. The process is repeated for each link until all links are either included in S_0 or eliminated. The group of links in S_0, are scheduled in the first time slot.

Next, all the links that were included in S_0 are removed from consideration (nodes are removed from the conflict graph) to form $G^{(1)}$. The process in the above paragraph is repeated for $G^{(1)}$ to form the next group S_1. The grouping procedure itself is repeated (for each $G^{(k)}$) by eliminating links that were included in prior groups until no links are left.

One can envision opportunistically scheduling links that were scheduled in previous groups, in later groups if possible. For example, one can envision a link scheduled in S_0 being again scheduled in S_1, after all the previously unscheduled links were considered for S_1. Instead of this, here, we seek to *reinforce* transmissions at the clients that receive information in S_1, using the LEDs that are idle. We find that this allows the clients to receive data at higher SNRs and therefore at better bit rates.

The steps of the process are formalized with the pseudocode below:

Module III: Scheduler

Input: G

Output: S_k, k^{th} subtopology

begin:

$k = 0$;

$G^{(0)}$ = copy of G;

m=size of $G^{(k)}$;

while m>0 **do**

$S_k \leftarrow$ one of links in $G^{(k)}$;

$conflict\_decision(S_k, G^{(k)})$;

m=m-(# of scheduled links in S_k);

k= k+1;

end while

function $conflict\_decision(S_k, G^{(k)})$

if link in $G^{(k)}$ not interfere with ALL links in S_k **then**

$S_k \leftarrow$ link;

link removed from $G^{(k)}$;

$G^{(k+1)}$ = copy of $G^{(k)}$;

end if

Module IV (Reinforcement via Idle LEDs): In each time slot, due to conflict, many LEDs may only illuminate without sending any meaningful information. These silent LEDs can be used to reinforce (provide additional power) active links. An idle LED array would change its beam-angle and choose an appropriate beamwidth to send information to a client that is currently being served by a different LED. The approach is based on the "water-filling" principle, which works if one were to use an energy detector using OOK modulation scheme. Since the data rate is related to the achievable SNR increasing SNR. We first choose the client with the lowest data rate to receive additional power from the nearest silent LED array (if possible). This procedure is repeated until all the silent LED arrays have been assigned to serve one of the active clients (or are categorized as "cannot be used for doing so").

6.3 Evaluations of VICO

We evaluate the performance of VICO via extensive simulations. Specifically, the performance of scheduling transmissions with and without VICO is compared. In the cases without VICO, we set the LED beamwidths to fixed default settings; further, the beam-angle is not tunable. For simplicity, we assume that with GPS or by using prior indoor localization methods each client is aware of its location and coordinates within the network [50, 49]. This information is then sent via explicit control messages to the central coordinator with VICO. In all our simulations, the pathloss model obtained via our PHY layer studies in Section 6.1 is adopted. In addition, we input the LEDs' configurations in the specific scenarios considered into a Matlab program in order to

check the illuminance distributions as in Section 6.1.

6.3.1 Simulation setup

We implement VICO using OPNET 16.0.A [39], as detailed below.

Phases of operations: We implement all the functional modules within VICO. Specifically, we divide the operations into four phases. Each of the phases roughly corresponds to a specific module within VICO.

- Phase 1: Each client identifies the LEDs that it can communicate with. It then sends an association message indicating its location, to the closest LED.
- Phase 2: Each LED array sends the information with regards to its associated clients and their locations to the coordinator. The coordinator determines the optimal transmission beamwidth and beam-angle for each client.
- Phase 3: The coordinator assigns links to time slots as per the heuristic-based approach of VICO discussed in Section 6.2. It also identifies idle LEDs and determines if they can reinforce transmissions to clients that are already being served.
- Phase 4: The coordinator notifies each LED about the transmission schedule. Then, each LED sends information to each client as per the schedule; the beamwidth and beam-angle are appropriately tuned specific to the client to which the transmission is performed.

Parameter settings: The experimental parameters in Section 6.1 are also used in the simulations. As mentioned, we adopt the channel model in Eqn.(6.2). The specific settings are provided in Table 6.2. The simulation setting is a room which is 4 meters long, 4 meters wide and 3 meters high. All the users are located at a height of 0.5

meters. Each adjustable LED can use two beamwidths, viz., 12.9° and 18°; it can also use four pointing directions and two steps in each direction (to define the beam-angle as discussed in Section 6.2). We use a basic data rate of 1 Mbps. The data rate is expected to increase in proportion to the SNR in stages (if the SNR is doubled, the data rate increases to 2 Mbps and so on) with OOK modulation [46].

Table 6.2: Simulation settings

# of directions	4
# of beamwidth	2
Collision model	Physical (accumulative) model
Basic data rate	1 Mbps
Traffic pattern	10000 pkt/sec
Packet size (association)	48 bits/pkt
Packet size (data and control)	32 bits/pkt
Network size	4m×4m×2.5m

Scenarios: We primarily compare the case with VICO with the cases where (a) one degree of freedom (only beamwidth or beam-angle) can be adjusted and (b) where no degree of freedom can be adjusted. We vary the client density in the network to examine the performance of VICO. In a small scenario, we perform an exhaustive search with all possible beamwidths and beam-angles and find the maximum throughput achieved (by consuming the minimum number of time slots). We compare that with the performance with VICO to examine how close the configuration of VICO is likely to be with respect to the optimal. Finally, we also examine the impact of reinforcement with idle LEDs.

Metrics: We evaluate VICO in terms of network throughput, number of time slots that are consumed to accommodate all links and the illuminance distributions. Each scenario is simulated at least 20 times and a statistical average is computed. An exception is the case where we examine the illuminance; here, we select a single scenario randomly.

6.3.2 Results and discussion

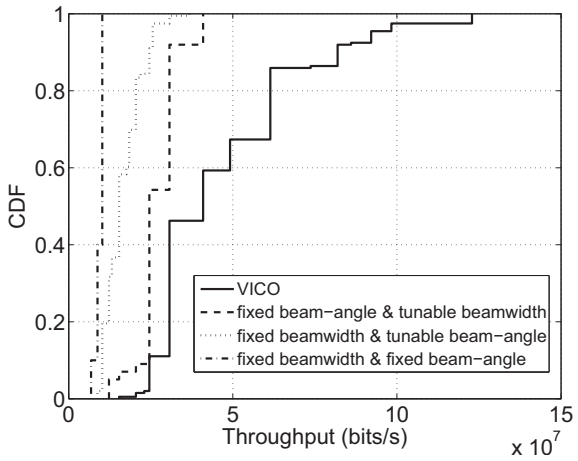

Figure 6.9: CDFs of the throughputs with the four approaches: VICO provides significant gains over other approaches.

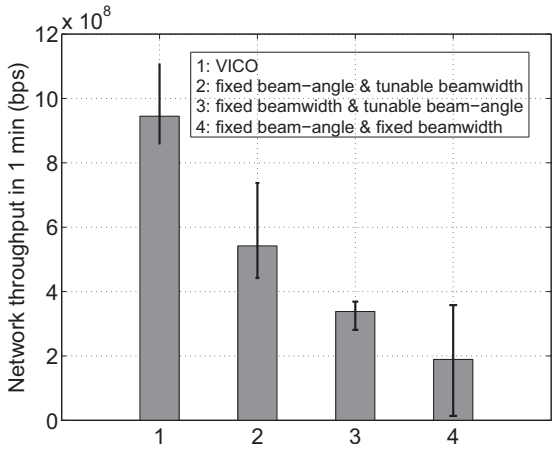

Figure 6.10: Averaged throughputs with error bars with the four approaches.

VICO is efficient in terms of network throughput and the number of time slots consumed for scheduling links: We compare the total network throughput with four different scenarios; (1) With VICO where we exploit both degrees of freedom (tunable beamwidth and beam-angle); (2) With one degree of freedom (tunable beamwidth but fixed beam-angle); (3) With one degree of freedom (tunable beam-angle but fixed beamwidth); (4) With fixed beamwidth and beam-angle. The total number of clients within the network is 20.

In Fig. 6.9, we plot the CDFs of the throughputs with the four scenarios. The average throughputs are shown in Fig. 6.10. It is easy to see that VICO almost doubles the median throughput compared to a case where only the beamwidth is adjustable. There is about a three-fold increase in the average throughput compared to the case where we only tune the beam-angle and the beamwidth is fixed at 18°. The improvement is about 5 fold if the LED has no freedom in terms of adjustment i.e., compared to a vanilla scheduler. These results demonstrate the efficacy of VICO in exploiting the degrees of freedom to provide a drastically increased throughput.

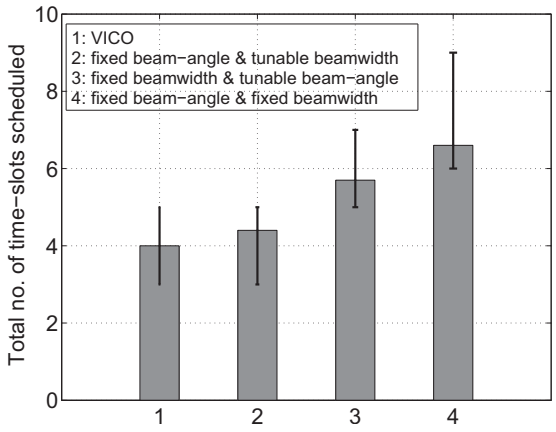

Figure 6.11: Total number of time slots required for scheduling all links with the four scenarios.

Next, we examine the number of time slots consumed in order to schedule all

the links in the network with the above four approaches. We again perform 20 simulation runs for each case and compute the average number of time slots used. For the four cases, the average number of consumed time slots are 4, 4.4, 5.7, and 6.6, respectively, as shown in Fig. 6.11. Intuitively, the fewer the time slots the more the clients that are included in each possible group. This implies that a higher number of simultaneous active transmissions are possible and this again reflects the higher throughput with VICO. We have also plotted the error bars (showing the maximum and minimum number of time slots consumed over the 20 runs with different client locations). Also notice that in the worst case, almost twice as many slots as with VICO, are needed with the simple scheduler that does not tune beamwidths or beam-angles.

Discussion: The reason for VICO's performance gains are the tuning of the beamwidth and beam-angle. Reducing the beamwidth provides the most gain since it improves the data rate while simultaneously improving spatial reuse. Simply tuning the beam-angle does provide an additional gain in spatial reuse; however, by itself it does not provide an increase in intensity (SNR) at the client and thus, an increase in data rate. By combining the two, VICO is able to offer significant benefits compared to the simple scheduler.

Comparing with the optimal: Next, we compare the performance of VICO relative to the optimal. Due to the complexity involved in finding the optimal (with an exhaustive search) we consider a small scenario where we cast 4 LEDs and 8 clients in a room of size 3m×3m×3m. We perform an exhaustive search with all possible beamwidths and beam-angles and find the best configuration that yields the minimum number of time slots. We then run VICO with the same client locations and find the number of slots consumed by the scheduler. We present the results with six such

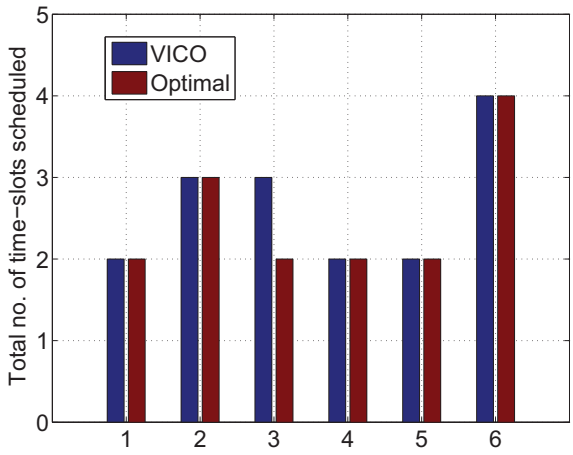

Figure 6.12: Comparing the VICO scheduler with an optimal scheduler.

scenarios with randomly chosen client locations in Fig. 6.12. The results demonstrate that VICO is able to schedule the LEDs with the minimum number of slots in five of the six scenarios considered. In Scenario 3, it needs one additional slot compared to the optimal scheduler. We do recognize that this is a fairly small setting and inconclusive with regards to larger scenarios; however the exponential complexity of finding the optimal makes a study in larger settings prohibitive.

We wish to point out that even if the scheduling with VICO is sub-optimal, the reinforcement process wherein idle LEDs are used to increase the SNR at active clients restores (discussed next) some of the throughput lost due to sub-optimal scheduling.

The contribution from idle LEDs: In the last functional module of VICO (Fig. 6.8), idle LEDs which are not transmitting in a time slot to avoid conflict to others try to reinforce the transmissions of adjacent LEDs to increase the SNR at a client. If using such reinforcement, an increase in SNR is possible, the data rate can potentially be increased at the corresponding clients. We perform 20 simulation runs with and without this reinforcement. (Note that the results reported earlier included the reinforcement component). We observe that the use of reinforcement with idle LEDs increases the

network throughput by 31.58%. Specifically, the throughput is 7.18×10^8 bps for the case without VICO's final module, and is 9.45×10^8 bps with the module turned on.

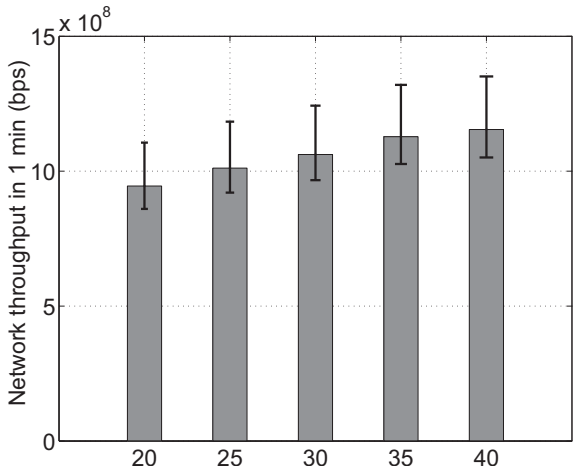

Figure 6.13: Throughput variations with client density.

VICO scales as the number of clients in the network increases: Next, we assess the performance of VICO with increased client (or node) density. We consider 5 scenarios wherein, the average node densities are 20, 25, 30, 35 and 40, respectively. The room size is identical to what was considered earlier i.e., 4m×4m×3m. From the plot of the average throughputs shown in Fig. 6.13, we see that the network throughput increases with node density when we increase the number of nodes. This is because VICO is able to increase the number of simultaneously active clients in these scenarios. In other words, the number of links in each group is increased. As one might expect, if the densities are increased much further, the throughput does not increase anymore and a saturation point is reached. In fact, this point of diminishing returns is seen when we change the number of nodes from 35 to 40.

The illumination variations with VICO are tolerable to the human eyes: Although we have earlier demonstrated that the variations in beamwidth and

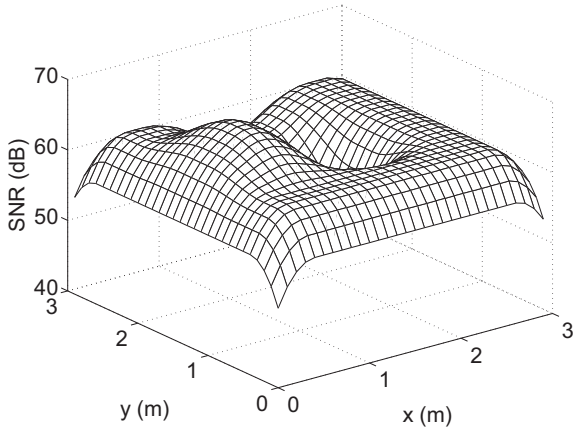

Figure 6.14: Illuminance distribution with a specific scenario created by VICO in our OPNET simulations.

beam-angle considered results in illumination variations that are tolerable to the human eyes, we now verify this with VICO, in the settings considered in the simulations. While we have verified that all the scenarios lead to tolerable illuminations, we present a sample result from our OPNET simulations for a single scenario here. In this scenario, we find that the LED 0 directs itself towards the LED 1's cell. LED 5 uses the larger beamwidth while the others use the smaller one (LED labels are shown in Fig. 6.4 for illustration). The results are in Fig. 6.14. We see from the figure that the minimum SNR is 49.2576dB and while the average SNR is 61.0447dB in the default setting. Looking back at Table 6.1 in Section 6.1, we find that these are within acceptable limits of human tolerance.

6.4 Summary

We design VICO, a framework for configuring indoor VLC networks. This is one of the very first work to consider higher layer protocol design and configuration issues in VLC settings. We first perform experiments on our PHY testbed which demonstrates that tuning the LED beamwidths and beam-angles can effectively result in increased SNR and thus, data rates. We design a framework VICO that efficiently schedules

LEDs to client transmissions while exploiting the ability to tune beamwidths and beam-angles. Via extensive simulations, we demonstrate the efficacy of VICO compared to schemes that do not exploit either or both of these degrees of freedom. VICO offers a 5-fold increase in throughput compared to a scheduler that does not exploit beamwidth adjustments or beam-angle tuning.

Chapter 7

Conclusions

This thesis explored multiple aspects of optical wireless communication systems and networks, including (1) MAC design for outdoor ultraviolet communications; (2) neighbor discovery methods for outdoor ultraviolet communications; (3) configuring indoor visible light communication networks.

Higher layer design for UV networks has been studied in this thesis. The medium access control schemes will have to conform to UV PHY properties. In this thesis, we first perform extensive experiments to understand and characterize the UV PHY layer. Secondly, we design UVOC-MAC, a MAC protocol that inherently accounts for the UV properties. This MAC scheme is suitable for full-duplex, four dimensional spatial reuse and multiple data rate adaptation. To study the performance of UVOC-MAC, we also provided collision probability analysis and network simulations. Results show that UVOC-MAC brings reduced collision and enhanced throughput.

To construct and maintain a network, neighbor discovery is an essential process. Since UV communication is mostly attractive in military usage, the current neighbor discovery methods are not appropriate. Therefore, we propose protocols for neighbor

discovery in a UV ad hoc network. We begin with basic algorithms with and without direction synchronization and then propose various techniques that can significantly enhance the performance of the neighbor discovery process.

In order to accelerate the network establishment, we propose a leadership based neighbor discovery protocol for UV networks. This algorithm drastically reduce the time required by neighbor discovery procedure by creating an interference-free broadcast environment. The nodes are able to terminate the neighbor discovery process as well.

Finally, we design VICO, a framework for configuring indoor VLC networks. After verifying that tuning the LED beamwidths and beam-angles can effectively result in increased SNR and thus, data rates, we design a framework VICO that efficiently schedules LEDs to client transmissions. The extensive simulations demonstrate the efficacy of VICO compared to schemes that do not exploit either or both of these degrees of freedom.

Bibliography

[1] J.M. Kahn and J.R. Barry. Wireless infrared communications. In *Proc. of the IEEE*, 1997.

[2] G. Chen, Z. Xu, and B.M. Sadler. Experimental demonstration of ultraviolet pulse broadening in short-range non-line-of-sight communicaiton channels. *Optics Express*, 2010.

[3] K. Cui, G. Chen, Z. Xu, and R. D. Roberts. Line-of-sight visible light communication system design and demonstration. In *7th IEEE, IET International Symposium on Communication Systems, Networks and Digital Signal Processing: 2nd Colloquium on Optical Wireless Communications*, Newcastle, UK, July 21-23, 2010.

[4] Z. Xu and B.M. Sadler. Ultraviolet communications: potential and state-of-the-art. *IEEE Commun. Magazine*, 46(5), May 2008.

[5] International commission on non-lonizing radiation protection (ICNIRP): Guidelines on limits of exposure to ultraviolet radiation of wavelengths between 180 nm and 400 nm. republished in Health Physics, 2004.

[6] G. Chen, Z. Xu, H. Ding, and B.M. Sadler. Path loss modeling and performance trade-off study for short-range none-line-of-sight ultraviolet communication. *Optics Express*, 17(5), March 2009.

[7] Z. Xu, H. Ding, B.M. Sadler, and G. Chen. Analytical performance study of solar blind non-line-of-sight ultraviolet short-range communication links. *Optics Letters*, 33(16), Augest 2008.

[8] T. Komine and M. Nakagawa. Fundamental analysis for visible light communication system using LED lights. *IEEE Trans. on Consumer Electronics*, 50(100-107), 2004.

[9] F.J. Lopez-Hernandez, R. Perez-Jimenez, and A. Santamaria. Ray-tracing algorithms for fast calculation of the channel impulse response on diffuse IR wireless indoor channels. *Optical Engineering*, 39(10), 2000.

[10] J. Grubor, O.C. Gaete Jamett, J.W. Walewski, S. Randel, and K.D. Langer. High-speed wireless indoor communication via visible light. In *ITG Fachbericht*, Berlin und Offenbach,2007.

[11] D.M. Reilly. Atmospheric optical communications in the middle ultraviolet. M.S. Thesis, MIT, Cambridge, MA, 1976.

[12] L. Wang, Z. Xu, and B.M. Sadler. Non-line-of-sight ultraviolet link loss in non-coplanar geometry. *Optics Letters*, 35(8), 2010.

[13] L. Wang, Z. Xu, and B. M. Sadler. An approximate closed-form link loss model for non-line-of-sight ultraviolet communication in non-colanar geometry. *Optical Letters*, 36(7), April 2011.

[14] IEEE Std 802.11 2007. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. http://standards.ieee.org/getieee802/download/802.11-2007.pdf.

[15] V. Vitsas and A.C. Boucouvalas. Performance analysis of the advanced infrared CSMA/CA MAC protocol for wireless LANs. *Wireless Communications*, 9(5), 2003.

[16] P. Chowdhury, B. Mukherjee, S. Sarkar, G. Kramer, and S. Dixit. Hybrid wireless-optical broadband access network(WOBAN): prototype development and research challenges. *IEEE Network*, 23(3), 2008.

[17] N. Agrawal, S.D. Milner, and C.C. Davis. Free space optical sensor network for fixed infrastructure sensing. In *SPIE*, 2009.

[18] R.R. Choudhury, X. Yang, R. Ramanathan, and N.H. Vaidya. Using directional antennas for medium access control in ad hoc networks. In *ACM MobiCom*, September 2002.

[19] R.R. Choudhury and N.H. Vaidya. Deafness: A MAC problem in ad hoc netwroks when using directional antennas. In *IEEE ICNP*, October 2004.

[20] A. Nasipuri, S. Ye, J. You, and R.E. Hiromoto. A MAC protocol for mobile ad hoc networks using directional antennas. *IEEE WCNC*, September 2000.

[21] C.M. Cordeiro, H. Gossain, and D.P. Agrawal. A directional antenna medium access control protocol for wireless ad hoc networks. In *Brazilian Telecommunication Society*, 2005.

[22] T. Korakis, G. Jakllari, and L. Tassiulas. A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks. In *MobiHoc*, 2003.

[23] J. Wang, Y. Fang, and D. Wu. SYN-DMAC: A directional MAC protocol for ad hoc networks with synchronization. In *MILCOM*, 2005.

[24] H. Gossain, C. Cordeiro, and D.P. Agrawal. MDA: An efficient directional MAC scheme for wireless ad hoc networks. In *GLOBECOM*, 2005.

[25] M. Takata, M. Bandai, and T. Watanabe. A directional MAC protocol with deafness avoidance in ad hoc networks. *IEICE trans on Communications*, E90-B(4), 2007.

[26] S. Vasudevan, J. Kurose, and D. Towsley. On neighbor discovery in wireless networks with directional antennas. In *IEEE INFOCOM*, March 2005.

[27] G. Pei, M. Albuquerque, J. Kim, D. Nast, and P. Norris. A neighbor discovery protocol for directional antenna networks. In *MILCOM*, October 2005.

[28] J. Luo and D. Guo. newblock Neighbor discovery in wireless ad hoc networks based on group testing. In *46th Annual Alerton Conference*, September 2008.

[29] M.J. McGlynn and S.A. Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In *Mobihoc*, September 2001.

[30] G. Jakllari, W. Luo, and S.V. Krishnamurthy. An integrated neighbor discovery and MAC protocol for ad hoc networks using directional antennas. *IEEE trans on wireless communications*, 6(3), 2007.

[31] S. Vasudevan, D. Towsley, R. Khalili, and D. Goeckel. Neighbor discovery in wireless networks and the coupon collector's problem. In *ACM Mobicom*, 2009.

[32] IEEE 802.15 WPAN task group 7 (TG7). Visible light communication. http://www.ieee802.org/15/pub/TG7.html.

[33] Z. Wu and T.D.C. Little. Network solutions for the LOS problem of new indoor free space opticle system. In *7th IEEE, IET international symposium on communications systems, networks, and digital signal processing*, Newcastle, UK, July 21-23, 2010.

[34] S. Jivkova, B.A. Hristov, and M. Kavehrad. Power-efficient multispot-diffuse multiple-input-multiple-output approach to broad-band optical wireless communications. *IEEE trans on vehicular technology*, 53(3), May, 2004.

[35] C. Tangtrongbenchasil. Ubiquitous optical wireless communication using optical micro-cell system. Ph.D. thesis, kochi university of technology, Japan, 2008.

[36] Y. Li, J. Ning, Z. Xu, S.V. Krishnamurthy, and G. Chen. UVOC-MAC: A MAC protocol for outdoor ultraviolet networks. In *18th IEEE ICNP conference, Kyoto, Japan*, October 2010.

[37] T. Matsumura. 10Mbps Visible Light Transmission System. IEEE 802.15 WPAN Task Group 7 (TG7),http://www.ieee802.org/15/pub/TG7.html, 2008.

[38] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw–Hill, New York, 1991.

[39] OPNET User's Documentation. http://www.opnet.com.

[40] Y. Li, L. Wang, Z. Xu, and S.V. Krishnamurthy. Neighbor discovery for ultraviolet ad hoc networks. *IEEE Journal on Selected Areas in Communications: Special Issue on Advances in Military Networking and Communications*, 2011.

[41] L. Wang, Y. Li, Z. Xu, and S.V. Krishnamurthy. A novel neighbor discovery protocol for ultraviolet wireless networks. In *ACM MSWiM*, Miami, Florida, USA, November 2011.

[42] Y. Li, L. Wang, J. Ning, K. Pelechrinis, S.V. Krishnamurthy, and Z. Xu. VICO: A Framework for Configuring Indoor Visible Light Communication Networks. In *submitted to IEEE INFOCOM*, Orlando, Florida, USA, March 2012.

[43] Datasheets of LED. http://www.optekinc.com/datasheets/OPA742.PDF.

[44] Datasheets of Photodiode. http://www.thorlabs.com/thorcat/13000/13054-S01.pdf.

[45] Datasheets of Lense. http://www.thorlabs.us/thorProduct.cfm.partNumber=FB450-40.

[46] H. Yang, J.W.M Bergmans, T.C.W. Schenk, J. Linnartz, and R. Rietman. Uniform illumination rendering using an array of LEDs: a signal processing perspective. *IEEE Trans on signal processing*, 57(3), March 2009.

[47] Nonresidential compliance manual, 2005.

[48] T. Jensen and B. Toft. Graph Coloring Problems. *Wiley Interscience*, 1995.

[49] K. Chintalapudi, A.P. Iyer, and V.N. Padmanabhan. Indoor localization without the Pain. In *ACM MobiCom*, Chicago, Illinois, USA, September 20-24,2010.

[50] N. Swangmuang and P. Krishnamurthy. Location fingerprint analyses towards efficient indoor positioning. In *IEEE PERCOM*, Hong Kong, 17-21 March 2008.

[51] A. Shrestharinis. Visible-light communication demonstrator: system modeling and analogue distribution network. Master thesis,Jacobs university, Bremen, Germany, 2009.

[52] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of inteference on multi-hop wireless network performance. In *ACM Mobicom*, San Diego, California, USA, September 14-19, 2003.

[53] E. Gelal, K. Pelechrinis, T.S. Kim, I. Broustis, S.V. Krishnamurthy, and B. Rao. Topology control for effective interference cancellation in multi-user MIMO networks. In *IEEE INFOCOM*, San Diego, CA, USA, 2010.

[54] G. Jakllari, W. Luo, and S. Krishnamurthy. An integrated neighbor discovery and mac protocol for ad hoc networks using directional antennas. *IEEE Trans. on Wireless Communications*, 6(3), 2007.