HETEROGENEOUS WIRELESS AND VISIBLE LIGHT COMMUNICATION FOR THE INTERNET OF THINGS

A Dissertation Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Shengrong Yin

May 2018

HETEROGENEOUS WIRELESS AND VISIBLE LIGHT COMMUNICATION FOR THE INTERNET OF THINGS

Shengrong Yin

APPROVED:

Omprakash Gnawali, Chairman
Dept. of Computer Science

Edgar Gabriel
Dept. of Computer Science

Carlos Ordonez
Dept. of Computer Science

Gopal Pandurangan
Dept. of Computer Science

Marcos Augusto Menezes Vieira
UFMG Brazil

Dean, College of Natural Sciences and Mathematics

Acknowledgment

This dissertation would not have been possible without the guidance of my advisor Prof. Omprakash Gnawali. I would like to first thank him for being an awesome adviser. I learned tons of research and time management skills from him. His humble attitude being an adviser always inspired me to do more with what I can offer. The most important aspect that I learned from him is being confident that I can do better research. There were both happy and frustrated times during this journey, but the best times I had at UH were the motivating discussions on ideas and the process that we collaborated in for scientific publications. I would also like to thank my dissertation committee: Edgar Gabriel, Carlos Ordonez, Gopal Pandurangan, and Marcos Vieira. Their experience and comments helped improve the dissertation significantly.

I would like to thank my collaborators from networked systems laboratory: Qiang Li, Dong Han, Hessam Mohammadmoradi, Milad Heydariaan, Nour Smaoui, and Nacer Khalil. Without them, much work from the dissertation would not have been possible as well. Qiang made a significant contribution on our CTC paper in DCOSS 2015. He implemented high-rate UDP packet generation via prototyping with iperf source code within two weeks. Nour and Milad also contributed significantly on our PurpleVLC work in EWSN 2018. I would also like to thank Philipp Sommer and Branislav Kusy for the collaboration on the study of dual-band wireless networks.

Meanwhile, I would like to thank Qing Wang and Domenico Giustiniano for the collaboration on OpenVLC1.0. I also want to thank Kevin Long for the ideas and dissemination of the classroom workshops using OpenVLC1.0 and PurpleVLC.

I won't be able to finish my dissertation without the support from my mom, dad, wife, family, and friends. Thanks to them for always being my strongest support.

Previously Published Material

Chapter 2 revises a previous publication [106]: S. Yin, O. Gnawali, P. Sommer, and B. Kusy. Multi Channel Performance of Dual Band Low Power Wireless Network, Proc. IEEE MASS, 2014.

Chapter 3 revises a previous publication [84]: S. Yin, Q. Li, O. Gnawali. Interconnecting WiFi Devices with IEEE 802.15.4 Devices Without Using a Gateway, Proc. IEEE DCOSS, 2015.

Chapter 5 revises a previous publication [104]: S. Yin, O. Gnawali. Towards Embedded Visible Light Communication Robust to Dynamic Ambient Light, Proc. IEEE GLOBECOM, 2016.

Chapter 6 revises a previous publication [107]: S. Yin, N. Smaoui, M. Heydariaan, and O. Gnawali. Purple VLC: Accelerating Visible Light Communication in Room-Area through PRU Offloading, Proc. ACM EWSN, 2018.

HETEROGENEOUS WIRELESS AND VISIBLE LIGHT COMMUNICATION FOR THE INTERNET OF THINGS

An Abstract of a Dissertation
Presented to
the Faculty of the Department of Computer Science
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

By
Shengrong Yin
May 2018

Abstract

Connecting sensor, control, and edge devices to the Internet in a reliable and robust way is critical to the success of many big data and IoT applications. Wireless technologies enable such connectivity but have come under increasing challenge due to the proliferation of devices and increase in data requirements. Devices with wireless connectivity compete with each other in the limited spectrum resources, causing spectrum crunch and interference, which significantly hampers the IoT vision. In this dissertation, we study how serious the problem of interference is in wireless networks for IoT, and then propose two solutions to solve this problem. Our goal is to connect IoT devices to the Internet with reliability, robustness, and adaptiveness using edge computing algorithms and methodologies in a practical manner.

One solution is to leverage the wireless interference across various IoT devices. We transformed the interference into a communication channel between these devices and evaluated its feasibility in practical environments. The communication channel was established based on the spectrum sharing by various wireless devices that are using different wireless technologies, such as WiFi, Zigbee, or Bluetooth. In this work, we have achieved one-way communication from WiFi devices to Zigbee devices. We have demonstrated the feasibility to send control signals utilizing the interference. This validates that interference utilization can be a practical solution to solve the spectrum-crunch problem.

The other solution is to avoid interference by exploring new spectrum resources that can provide wireless connectivity. We adopt visible light as the communication medium since it is ubiquitous and free from wireless interference. Existing embedded

LED-to-LED communication is considered a promising technique for IoT connectivity. However, low-cost embedded visible light communication (VLC) has been largely restricted by its reliability, robustness, and speed. In this work, we propose adaptive ambient light cancellation to improve the robustness of embedded VLC, we also design, implement, and open-source a novel embedded VLC platform with a 6-7x performance gain compared to state-of-the-art.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless connectivity has been an inevitable component of the digital gadgets in our daily lives. Bluetooth [6], Zigbee [20], Wi-Fi [17], and LTE technologies [1] are all around us providing streaming service, news feed, as well as notifications to enrich our lives. Devices with wireless connectivity have been shipped to every corner of this planet. As a reliable and robust way to connect users to the Internet, different wireless technologies are working together to bring end users the best experience for streaming/controlling applications. Compared to a wired approach, wireless connectivity offers great convenience in terms of mobility.

However, today's wireless devices have been sharing a limited unlicensed spectrum, causing spectrum crunch when many devices are competing for the channel resources in a limited space. This becomes more problematic due to the growing number of connected devices. Today's wireless technologies such as Wi-Fi, Bluetooth, and Zigbee are sharing 2.4 GHz spectrum band, which is unlicensed and globally accessible. Industry's leading low-power wireless-mesh networking technology - Thread [15] is also using 2.4 GHz as its communication band. Researchers and industry practitioners proposed various approaches to avoid interference between various wireless technologies that are sharing the same frequency band. However, there is a lack of knowledge when multiple wireless technologies are coexisting in the same place. Some of the existing work tried to solve this problem by proposing solutions allowing various wireless technologies to coexist [49]. In this dissertation, we first improve existing study by designing innovative wireless testbeds and evaluating performance issues while various wireless technologies are existing in real-world scenarios. We then propose solutions on how to interconnect various wireless technology as a benefit from the interference. Last but not the least, we propose a low-power embedded visible light communication system as a complimentary wireless technology for today's IoT (Internet of Things) devices. The goal is to expand the horizon of wireless connectivity options and enhance communication performance in challenging environments where RF-based technology may not work well.

1.1 Dissertation Overview

Low-power wireless communication is the basic building block of the Internet of Things (IoT). Applications for the Internet of Things apply to human daily lives ranging from medical care to smart agriculture and have been developing at an extremely fast pace. The ever more crowded wireless spectrum and the associated communication reliability and robustness problems, however, have started to have a significant impact on the uptake of IoT platforms by the industry and academia. Srinivasan *et al.* [85] state that a mismatch between abstraction and reality in low-power wireless has been a tremendous impediment for protocol designers. One of the main reasons is that link-layer performance of radios, conditional to different modulation schemes, transmission powers, and operating frequencies, is rarely understood in sufficient detail. The network and link-layer decisions of when, at what power, and with which neighbor to communicate is a complex decision due to the dynamic and unstable characteristics of the low-power wireless links. Nevertheless, network protocols can greatly benefit from utilizing high-quality wireless links in terms of energy efficiency of the network as a whole, improved network lifetime, and robustness of the IoT network. As a first step towards building a reliable and robust wireless system, we argue that a better understanding of low-power wireless, based on detailed physical and link-layer measurements in a real-world scenario simultaneously, can provide insights for improving network reliability.

Understanding the wireless interference is the first step to provide concrete solutions to build a reliable and robust wireless system. Based on this study, we proposed the first solution to alleviate the spectrum-crunch problem. We called this solution crosstalk-based communication (CTC). CTC helps to utilize interference between multiple wireless technologies. Packets transmitted from the WiFi radio can be served for two purposes. One is for normal WiFi transmission. Second is to create a channel that can transmit information to devices with other wireless technology such as Bluetooth or Zigbee.

Recently, commoditization of low-power wireless technology has led to a decrease in the cost of hardware components. Meanwhile, low-cost IoT devices with wireless connectivity have been shipped to market at an increasing quantity. Connecting these devices on the Internet requires additional gateways to convert message type from low-power wireless to an existing TCP/IP stack. The existing solution to enable such a communication is through a gateway or a bridging device. At a high level, the device is a router. It has one 802.15.4 interface to communicate with the 802.15.4 network. The other interface may be WiFi or wired, which is used to provide Internet access. The device shuttles traffic between the two interfaces. A control message (e.g., to turn a light on) coming from a smartphone app, travels to the gateway (possibly through the Internet), is translated appropriately for 15.4 network, and is transmitted by the 15.4 radio. On the software side, the bridging may happen at the application layer (with custom application-specific messages) or at the network layer (with standardized network-layer protocols). Recent IETF standards such as 6LoWPAN [83] support development of sensor networks with this architecture. This architecture, which we call gateway-oriented architecture, has served us well as evidenced by the vibrant ecosystem of smart home devices and companies that sell those products. Despite some application deployments using WiFi-based sensors and controllers, 802.15.4 or low-power low-rate radios occupy a unique point in the price and design space that they are likely to be a radio of choice for many years to come. However, the core idea in our approach is to have WiFi devices transmit packets with special patterns representing the information to be conveyed to the 15.4 network. The transmission is done on a WiFi channel overlaps with the 15.4 channel on which 15.4 devices are listening. The 15.4 devices sample the signal on the channel due to WiFi transmissions (which we would typically call crosstalk or interference and try hard to avoid) and interpret the information in the pattern.

The second solution is to avoid interference by exploring a new spectrum frequency that can provide connectivity as well, which is visible light communication. Visible Light Communication (VLC) has been proposed not only as an alternative wireless channel for communication for IoT but also as one of the ways to address spectrum crunch. Many research projects have tried to advance different flavors of VLC systems in the last few years. Most discussed VLC systems are perhaps the ones that use light bulbs at homes (e.g., LiFi [39]), however, there is a body of work on low-cost low-power embedded VLC systems based on LEDs [96]. These systems commonly use a photodiode to receive the transmitted light signals. The photodiode gets saturated or performs poorly in bright light or in fluctuating ambient light. Results from existing studies [44] show that with strong ambient light, the system will fail to deliver packets. In an indoor scenario, the VLC system needs to be robust against not only bright light but also to the level of changes in illumination throughout the day or night. Addressing the poor performance of embedded VLCs in bright and changing ambient-light conditions is necessary for embedded VLC systems to mature into systems that can provide robust and reliable communication. Most importantly, for VLC technology's potential to address the spectrum crunch, it is essential that these systems achieve a level of robustness far beyond what the state-of-the-art achieves.

Getting a low-cost embedded VLC system to work robustly in bright light and changing ambient-light levels is extremely challenging. In presence of bright ambient light, for example in a stadium or near a window during the day time, the photodiode used as the receiver will easily get saturated causing the reception to fail: the receiver will not be able to distinguish the ambient and transmitted light because it is already saturated. Similarly, different levels of sensitivity on the photodiode may be required depending on the level of ambient light in the environment. Existing prototypes of embedded VLC do not work well in these challenging environments. Previous work on low-cost VLC systems use photodetectors that perform poorly when subjected to bright or fluctuating ambient light. In typical LED to photodiode communication systems, the existing approach is to switch to a different type of receiver or transmitter when communication degrades. These workarounds, however, do not directly address the main problem by actively canceling the ambient light based on its intensity. In this work, we aim to fill that gap in the state-of-the-art of embedded VLC systems. Meanwhile, existing low-cost VLC systems do not provide a reliable high-speed communication to facilitate the research in ubiquitous light-based applications and services. Our work will also bridge this gap by providing an affordable approach with much faster data rate, better reliability, and improved robustness over state-of-the-art.

1.2 Dissertation Contribution

Heterogeneous wireless networks are being deployed in the world. The spectrum-crunch problem is rising from this deployment. We address this issue with the following contributions in this dissertation.

- We designed and implemented a large-scale testbed-based wireless performance study for both 802.11n networks and 802.15.4 networks. We provided insights on how heterogeneous wireless networks coexist and quantified the interference between these two wireless technologies.
- We proposed a novel communication primitive called crosstalk-based communication which interconnected 802.11n networks and 802.15.4 networks without using a gateway. We designed, implemented and evaluated this primitive with commercially available devices in real-world scenarios. This primitive helps to embrace the crosstalk-based interference so that a better application service can be built on top of crosstalk-based communication.
- We designed and implemented an embedded visible light communication system. The proposed low-cost low-complexity communication solution does not

bring overhead to existing heterogeneous wireless networks. Instead, it provides another complimentary wireless connectivity for today's IoT devices. We also designed an ambient light-cancellation system to offer a better reliable and robust visible light communication system.

- The proposed embedded VLC system is fully open-source and has been deployed in the university classrooms as educational kits in the US and Mexico.

1.3 Dissertation Organization

We organize the rest of the dissertation as follows. In Chapter 2, we present detailed measurement methodologies and results on wireless interferences and validate our observations from realistic wireless channels and environments. In Chapter 3, we propose a novel approach to utilize the wireless interference so that cross-technology devices will coexist in a better way while reducing system complexity and cost. Chapter 4 describes two open-source hardware and software platforms for embedded VLC exploration. These aim at the interference-free wireless network. Chapter 5 presents the detailed ambient light-cancellation mechanism to ensure an always robust embedded VLC system even in challenging ambient-light environments. Chapter 6 introduces the first embedded VLC prototype that is at least 6-7X faster compared to state-of-the-art. It describes the hardware and software design for this platform in detail and presents a rigorous evaluation in an indoor environment. Chapter 7 introduces the classroom workshops using the designed platforms as well as the lessons learned from these workshops.

Chapter 2

Understanding Radio Interference

Low-power wireless communication is the basic building block of wireless sensor networks. Applications for wireless sensor networks apply to human daily lives ranging from medical care to smart agriculture and have been developing at a fast pace. The ever more crowded wireless spectrum and the associated communication reliability and robustness problems, however, have started to have an impact on the uptake of wireless sensor network platforms by the industry. We argue that a better understanding of low-power wireless, based on detailed physical and link-layer measurements, is necessary to improve the network reliability.

Srinivasan *et al.* [85] stated that mismatch between abstraction and reality in low-power wireless has been a tremendous impediment for protocol designers. One of the main reasons is that link-layer performance of radios, conditional to different modulation schemes, transmission powers, and operating frequencies, is rarely understood at a sufficient detail. The network and link-layer decisions of when, at what power, and with which neighbor to communicate is a complex decision due to the dynamic and unstable characteristics of the low-power wireless links. Nevertheless, network protocols can benefit by utilizing high-quality wireless links in terms of energy efficiency, improved network lifetime, and robustness of the wireless sensor network.

Commoditization of low-power wireless technology has led to a decrease in the cost of hardware components. Several platforms and testbeds exist that offer wireless communication at multiple frequencies. Because multi-radio wireless sensor network platforms are relatively new, the underlying performance of radio communication on different channels and correlation of radio links across multiple bands are not well understood. This measurement study aims to provide network protocol designers and practitioners with a detailed performance evaluation of multi-band and multi-channel radio communications. We evaluated 900 MHz and 2.4 GHz wireless links operating simultaneously on two different testbeds located in office-like environments.

In addition to studying the correlation of two different radio bands, we also studied the impact of external interference on low-power wireless links. Our testbeds exhibited different external interference patterns due to their locations in different countries (USA and Australia) which may have different patterns of use for the two bands. Some of the major external interference sources in 2.4 GHz band are common around the world, because of the widespread use of technologies in offices and households, such as Wi-Fi, Bluetooth, and Zigbee. On the other hand, the 900 MHz frequency band is mostly used for proprietary low duty-cycle communications and is not as likely to record significant external interference. Less external interference allows more reliable transmissions with fewer retries, which also saves battery power.

Increasing popularity of multi-band communication has triggered studies on the benefits of multi-radio technology. Kusy *et al.* [60] found that network reliability can be significantly improved by using dual-radio communication. They compared the performance of dual-band with single-band communication on a 30-node testbed and concluded that dual-radio communication can help increase the network throughput without a significant impact on energy efficiency. However, this work is based on measurements at the network layer, which differs from our investigations of link layers. Our study is based on two different dual-radio testbeds which helps us eliminate the impact of local climate and external interference on the performance of the wireless radio links. Our experimental setup allows us to run experiments on multiple channels simultaneously, improving the scalability of our experiments.

Our main contributions for this chapter are as follows:

- We presented detailed channel measurements on a large scale dual-band testbed.
- We presented experimental results showing that the 900 MHz band provided more reliable connectivity compared to the 2.4 GHz band, especially on radio channels that experienced high levels of 802.11 interference.
- We presented a study of the temporal properties of simultaneously operating dual bands.

2.1 Related Work

Wireless sensor networks are used in various application domains in cyber-physical systems. They typically use unlicensed bands of the wireless spectrum. However, the unlicensed spectrum band is shared with other technologies such as Wi-Fi or Bluetooth which can have a severe impact on the performance of wireless links between sensor nodes. Therefore, mitigation of channel interference and coexistence of several technologies in a shared wireless spectrum have been studied intensively in recent years. In the following, we provide an overview of prior research undertaken to improve the understanding of the wireless channels in both the 900 MHz and 2.4 GHz frequency bands.

Wireless Measurements of the 2.4 GHz Band. Lee *et al.* [61] sampled the RSSI register from the TI CC2420 transceiver at 1 KHz in controlled areas. They found noise patterns with temporal variations and proposed a model to simulate packet delivery based on different noise signatures from the empirical measurements. Rusak and Levis [78] performed experimental studies to understand and model the wireless channel. They found that bursts from RSSI traces in the wireless measurements occur at longer time-scales compared to burstiness over short time-scales. Srinivasan *et al.* [87] studied packet reception in 802.15.4 channels and proposed a quantification metric for the link burstiness (β-factor). More recently, Srinivasan *et al.* [85] presented a conceptual model of wireless networks and used empirical measurements for TelosB and MicaZ nodes to validate or dispute the claims in the proposed model. They observed 802.11b interference at 45 dBm above the noise floor and suggested avoiding channels that coexist with Wi-Fi networks. Sha *et al.* [82] performed wireless measurements in the 2.4 GHz band across multiple channels in a wireless sensor network deployed in residential environments. They found significant temporal and spatial variations in the quality of channels. They also reported that 2.4 GHz channels exhibit non-periodic noise. Hermans *et al.* [42] improved packet reception ratios under heavy interference from 45% to 61% by classifying corrupted packets in unique patterns using machine learning. Boano *et al.* [25] proposed the JamLab testbed to regenerate precise interference patterns using a low-cost infrastructure to evaluate the performance of existing sensor network protocols under interference in the 2.4 GHz band. Noda *et al.* [72] presented a new channel quality metric to quantify spectrum usage based on the availability of the channel over time.

Wireless Measurements of the 900 MHz Band. Boers *et al.* [26] exploited the

Table 2.1: Comparison of transceiver configurations

Transceiver	Range	Resolution	Sensitivity	TX Power	Modulation	Trans. Rate	Channels	Frequency
RF231	81 dB	3 dB	-91 dBm	3 dBm	OQPSK	250 kbps	11-26	2.4-2.485 GHz
RF212	87 dB	3 dB	-98 dBm	3 dBm	OQPSK	250 kbps	1-10	906-924 MHz

method to classify the interference in the wireless channels of the 904 to 928 MHz ISM band, which provided us some insight to investigate the pattern of interference in 900 MHz band, especially in office-like environment. Incel *et al.* [51] studied how transmission on one channel can cause interference on neighboring channels. They used a platform with a radio transceiver operating in the 868/915 MHz ISM band.

Multi-Band Wireless Measurements. There are a few wireless sensor network testbeds on which we can perform multi-band wireless measurements. Handziski *et al.* [40] presented TWIST, a scalable and flexible testbed architecture for indoor deployment of wireless sensor networks. They deployed two platforms on TWIST: 102 TmoteSky nodes working on 2.4 GHz band and 102 eyesIFX nodes working in 868 MHz band. Each node uses one transceiver in one band. Using Fleck nodes, a predecessor to the Opal [55] platform using the same radio transceivers, Kusy *et al.* [60] performed empirical experiments on a testbed of 30 nodes to compare the CTP performance under single and dual radio settings. Opal and Fleck nodes have two transceivers operating in the 900 MHz and 2.4 GHz bands. These prior experiments mainly study the network-layer performance. Unlike those studies, our goal is to understand the physical/link-layer performance under simultaneous dual radio communication. Lim *et al.* [69] developed the Flocklab testbed for distributed, synchronized tracing, and profiling of embedded wireless systems. This testbed has four types of nodes: Tmote Sky, IRIS, Opal, and Tinynode. Thus, one could perform multi-band measurements on this testbed. In our study, we use Twonet testbed [64] at the University of Houston and a 17-node testbed of Opal nodes at the Commonwealth Scientific and Industrial Research Organisation (CSIRO).

2.2 Measurement Study

We describe the experimental setup used for this dual-band wireless-channel measurement study.

2.2.1 Platform

Opal: The Opal platform [55] was developed by CSIRO to provide the wireless channel diversity by using two radio transceivers operating in different bands. It embeds an Atmel ARM Cortex-M3 low-power MCU and two radios: the Atmel AT86RF212 transceiver (900 MHz) and the AT86RF231 (2.4 GHz) transceiver. Each radio uses a separate antenna matched to the RF frequency.

AT86RF231: The Atmel AT86RF231 [4] is a low-power 2.4 GHz radio transceiver designed for IEEE 802.15.4 applications. The AT86RF231 is suitable for applications in wireless sensor networks that operate in the 2.4 GHz band. Like the popular TI

CC2420 radio, Atmel's RF231 also supports 16 channels in the 2.4 GHz band with a channel spacing of 5 MHz. The center frequency, F_c, of these channels is defined as follows:

$$F_c = 2405 + 5 * (k - 11)[MHz], k = 11, ..., 26 \tag{2.1}$$

where k is the channel number from 11 to 26.

The AT86RF231 can update the RSSI register every 2 μs and the readings are in the range between 0 and 28. A register value of 0 indicates an RF input power of less than -91 dBm [4]. For a register value ranging from 1 to 28, we can compute the corresponding RF power as follows:

$$RSSI(dBm) = -91 + 3 * (R - 1) \tag{2.2}$$

Here R indicates the raw register value. The RF231 transceiver has a minimum RF sensitivity of -91 dBm.

AT86RF212: The Atmel AT86RF212 [3] is a low-power, low-voltage 700/800/900 MHz transceiver designed for the IEEE 802.15.4 standard and a high-data rate for ISM applications. For the sub-1 GHz bands, it supports multiple data rates (20, 40, 100, 200, 250, 500, and 1000 kbps). In our experiments, we configured both the radios to use OQPSK modulation and 250 kbps data rate. There are 10 channels in the North American ISM band from 902 to 928 MHz with a channel spacing of 2 MHz according to IEEE 802.15.4-2003/2006. The center frequency of these channels is defined as:

$$F_c = 906 + 2 * (k - 1)[MHz], k = 1, 2, ..., 10 \tag{2.3}$$

where k is the channel number from 1 to 10.

Under the OQPSK modulation, the RF212 transceiver can update the RSSI value every 8 μs. Similar to the RF231 radio, RSSI readings have a resolution of 3 dB and the register value ranges from 0 to 28. An RSSI value of 0 indicates an RF input power is equal or less than the minimum RF sensitivity, which is -98 dBm for OQPSK [3]. For an RSSI value ranging from 1 to 28, we can use the following formula to calculate the RF power:

$$RSSI(dBm) = -98 + 3.1 * (R) \tag{2.4}$$

Table 2.1 summarizes the transceiver settings used during the experiments.

2.2.2 Testbeds

We conducted our experiments on two testbeds.

Twonet: Twonet is a large-scale wireless sensor network testbed deployed at the Phillip G. Hoffman Hall at the University of Houston. The testbed consists of 100 Opal nodes which are connected to a network of 20 Raspberry Pi nodes, which are connected to a server [64]. The Opal nodes are deployed across four floors of the building.

CSIRO testbed: The indoor testbed deployed at CSIRO consists of 17 Opal nodes distributed over two adjacent wings of a large building, which includes offices, storage areas and laboratories. Each Opal node is attached to a PandaBoard embedded Linux PC, which provides an Ethernet-based backchannel to a central server for logging serial output.

2.2.3 Metrics

In our experiments, we perform channel measurements by letting each node on the testbed transmit radio packets every 10 s and receive all the packets transmitted by other nodes. There is a low probability that two nodes transmit at the same time due to randomization and jitter in the transmit logic. The RSSI was read for each received packet and the sequence numbers was embedded in the packet to calculate the packet reception ratio on a specific link. We pick the following two metrics to evaluate the experimental data gathered on the two testbeds.

Received Signal Strength Indicator (RSSI). We can read RSSI values directly from the radio transceivers. The RSSI is a combination of the received radio signal and the noise floor/interference at the receiver. Even if there is no packet currently being received, we can still capture RSSI readings to sample the background noise of the environment.

Packet Reception Ratio (PRR). The Packet Reception Ratio (PRR) represents

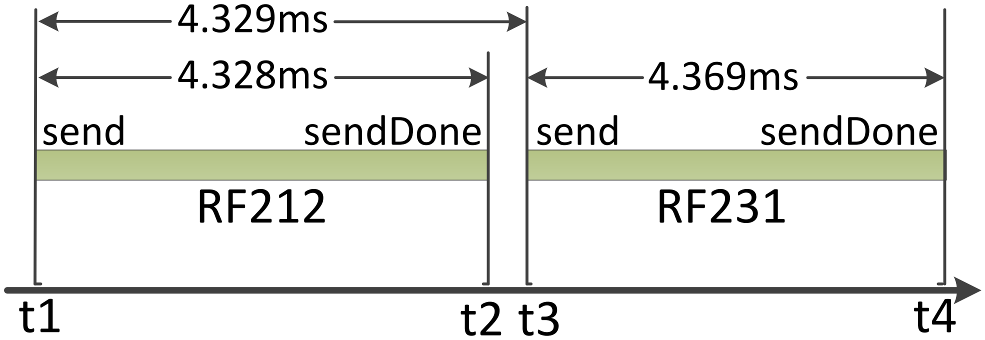


Figure 2.1: Packet transmission timing in the two bands during our experiments.

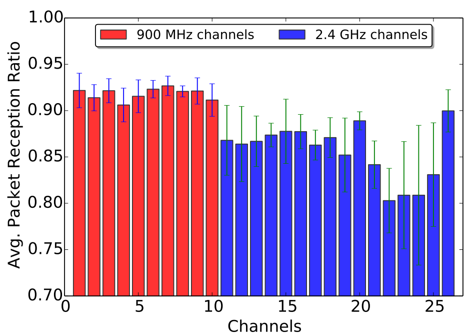


Figure 2.2: Average PRR across all 100 nodes scanning all channels.

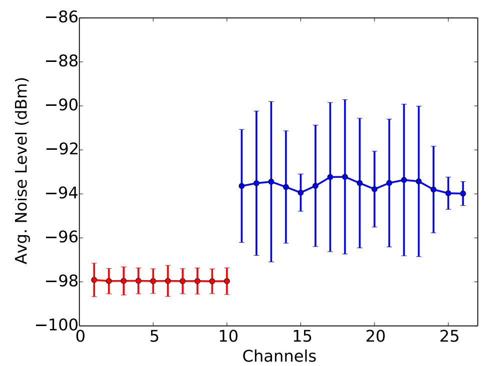


Figure 2.3: Average sampled signal strength across 16 nodes on Twonet.

the success rate for link-layer packet transmissions between a sender-receiver pair. This metric is frequently used to assess the quality of links and to predict the performance of higher-layer protocols.

Table 2.2: Node-channel Assignment Pattern Used for Concurrent Measurements on four channels

Node ID	1	2	3	4	...
Channels	(6,16)	(10,26)	(6,16)	(10,26)	...

2.2.4 Concurrent Transmissions in Two Bands

Although we use the phrase *concurrent transmission in two bands* in this study, the packets are not transmitted concurrently in the two bands. In our experiments, the mote transmits a packet using one radio. Immediately after the completion of this transmission, the node transmits a packet using the second radio. Figure 2.1 shows the average packet transmission timing on the two radios and the time between the two transmissions. RF212 takes an average of 4.328 ms to send a packet. RF231 takes an average of 4.369 ms to send a packet. There is approximately 1 μs time interval between the sendDone event from RF212 and the send call to RF231. Thus, the motes transmit the packets nearly concurrently in the two bands.

2.3 Measurement Campaigns

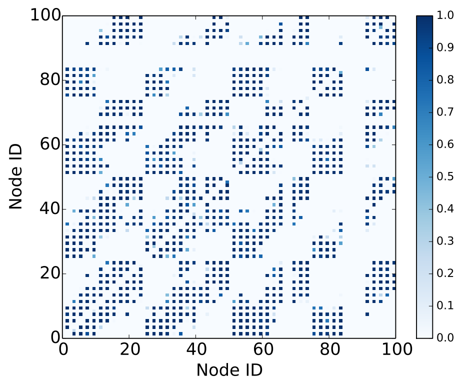
We wrote a TinyOS application that transmits a packet every 10 s with both the transceivers at the same time. The application logs all the packets received. The motes also record receive and transmit time-stamps and RSSI readings. The motes send all the captured data to the server which manages and configures the status of the testbed.

To measure PRR across all the channels on the testbed, we do the following experiment. The motes transmit/receive packets and sample noise in the two bands simultaneously, using one channel from the 2.4 GHz band (RF231) and one channel from the 900 MHz band (RF212). The motes with even node id sample two channels and the motes with odd node id sample different two channels using the pattern shown in Table 2.2. Thus, we performed packet and channel measurements on four channels, concurrently. This methodology provides channel diversity without losing the experiment scalability. We configured the radios to use the settings in Table 2.1. Each measurement campaign lasted 60 minutes, during which each mote transmitted 720 packets (360 on each channel). We repeated each experiment three times. We performed these measurements at night when the spectrum was quiet without human activity.

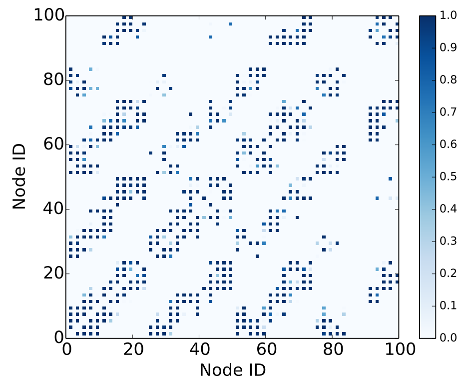
In this section, we present the results from our detailed measurements of multiple channels on multiple testbeds. Overall, we find that 900 MHz band can provide 15% more connectivity compared to the 2.4 GHz band on our testbeds.

2.3.1 Performance Across 900 MHz and 2.4 GHz Bands

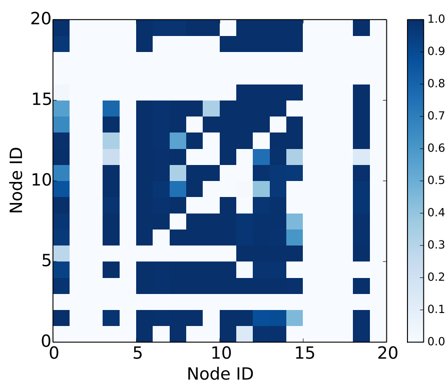
Figure 2.2 shows the average PRR across all channels in the two bands on Twonet. We can see 900 MHz channels provides better PRRs compared with 2.4 GHz channels. 2.4 GHz channels were quite busy on our testbed. This is a somewhat expected result and also reported in previous works. The main difference is that our observations were from concurrent channel sampling. Overall, the links on 900 MHz channels had



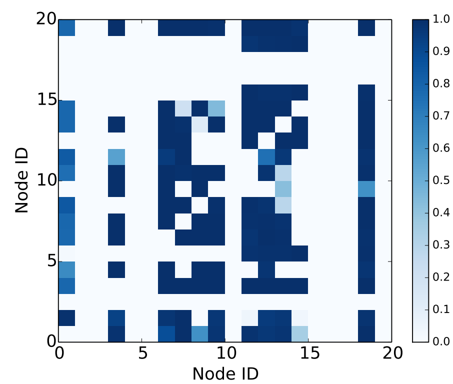
(a) Twonet testbed: PRR on channel 10



(b) Twonet testbed: PRR on channel 26



(c) CSIRO Testbed: PRR on channel 10



(d) CSIRO Testbed: PRR on Channel 26

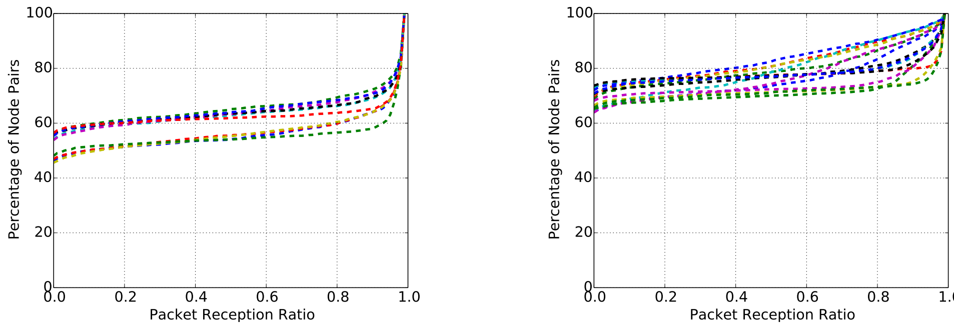
Figure 2.4: The 900 MHz band provides more connectivity compared with the 2.4 GHz band both on Twonet and CSIRO testbed. The color displayed on each small square box represents the PRR for the corresponding node pair. Darker colors imply higher PRR.

an average PRR of 91.82% (maximum was 92.67% and minimum was 90.61%) and the links on 2.4 GHz had an average PRR of 85.59% (maximum was 89.97% and minimum was 80.29%) during our measurement campaigns.

Each testbed environment may be different and hence level of noise may be different, which could be one explanation for the difference in PRRs across the testbed. To sample noise across all the channels and locations on the testbed, we ran an experiment cycling through all 900 MHz and 2.4 GHz channels using a technique called channel scanning [105]. Figure 2.3 shows the data from the measurement. We found that 900 MHz channels were quieter than the 2.4 GHz channels. Particularly, in the 2.4 GHz band, channels 15, 20, 25, and 26 were Wi-Fi free channels. Thus, these channels had less variable noise than other channels. We observed lower noise and variation on the 900 MHz channels. This is despite their overlap with GSM signals [60]. Our guess is these signals or devices that use 900 MHz are not a big factor in the building where Twonet is deployed.

2.3.2 Distribution of Packet Reception Ratio

Figure 2.4 shows how PRR is distributed across the Twonet testbed and the CSIRO testbed. The color of each (x,y) cell indicates the PRR for the corresponding node-pair x and y. The first observation was there were more colored cells and darker colors on PRR heatmap for channel 10 compared to the PRR heatmaps for channel 26. This suggested that there was better connectivity on channel 10 than with channel 26. Traditionally, in sensor network literature, channel 26 was reported as



(a) CDFs of PRR for each channel (1 to 10) in 900 MHz band

(b) CDFs of PRR for each channel (11 to 26) in 2.4 GHz band

Figure 2.5: Comparison of connectivity with 900 MHz and 2.4 GHz channels on Twonet testbed.

the best channel for network formation. Our data showed that channel 10 was even better.

The second observation was the diagonal pattern of the colored cells in Figure 2.4, especially on the heatmaps for Twonet. The pattern is the result of the deployment topology of the motes on the testbed which results in a specific connectivity pattern. On Twonet, 100 Opal nodes are deployed across four floors of an academic building, with 25 nodes deployed in the ceiling of each floor. Nodes 1-25 are on the third floor. Nodes 26-50 are on the second floor. Nodes 51-75 are on fourth floor. Nodes 76-100 are on fifth floor. On each floor, five Opal nodes, with consecutive node ids, are connected via cables to the same USB hub. Although cables of different lengths were used to separate them as far apart as possible, the motes naturally form spatial clusters around the USB hub. Hence, nodes closer in node ids are likely to be able to communicate with each other resulting in a large number of colored cells along the diagonal. The figure also has second and third diagonals, each separated by about 25 units to the left and right of the main diagonal connecting the bottom-left to top-right corners. These additional diagonals are due to inter-floor connectivity and the pattern of deployment: the mote ids were sequential on each floor. Thus, nodes 1-5, which are near one corner of the third floor, are likely to be able to communicate directly with nodes 26-30, which are deployed in the same corner but on the second floor. Similarly, nodes 6-10 are likely to have good connectivity to not only the nodes adjacent to it on the same floor but also nodes 31-35 which are on the adjacent floor direct below it. Thus, we get a second diagonal from (0,25) to (75,100). Other diagonals also capture similar geometric information about the deployment within a floor and across the floors. The PRR heatmap for the CSIRO testbed did not show similar geometric pattern because the nodes were deployed over a large area within the same floor.

Figure 2.5 shows the distribution of link PRR across node-pairs for the channels in 900 MHz and 2.4 GHz bands. We find that a large number of node-pairs do not have viable links. With 900 MHz channels, approximately 45-60% of node-pairs are not connected with links. On 2.4 GHz channels, 65-75% of node-pairs are not connected with links. Thus, 900 MHz band provides more connectivity than 2.4 GHz band. We also find that almost 20% of the links are close to 100% PRR in the 900 MHz band. Regardless of the channel, a much smaller fraction of nodes have links with close to 100% PRR in the 2.4 GHz band. Another observation is the clustering of PRR distribution for 900 MHz into two distinct bands. We do not understand the

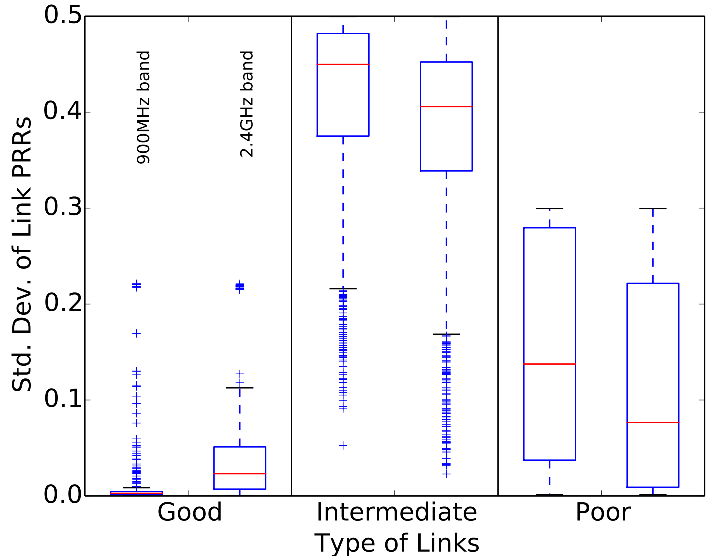


Figure 2.6: Boxplot of standard deviation of link PRRs across all the links in 900 MHz band and 2.4 GHz band.

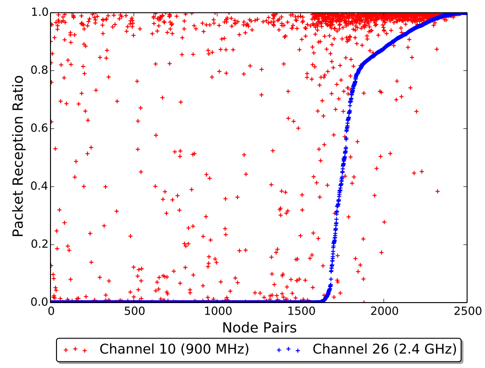
reason for this clustering given 900 MHz channels are fairly similar to each other in terms of their PRRs as shown in Figure 2.2.

Next, we examine the variation in the channel PRRs in the two bands. We sampled PRR of channel 10 and channel 21 concurrently for 60 hours on Twonet using the technique described in Section III. We calculated the PRR for each link over each one-hour interval, sliding the window by 1 h. Thus, we have 60 PRR values for each link covering 60 h. Then, we compute the standard deviation of these 60 PRR values as a measure of temporal variation of link PRR. We aggregated the standard deviation of PRR from all the links and divide the dataset into three segments by the average PRR: links with less than 10% PRR (as poor links), links with 10-90% PRR (as intermediate links), and links with greater than 90% PRR (as good links).

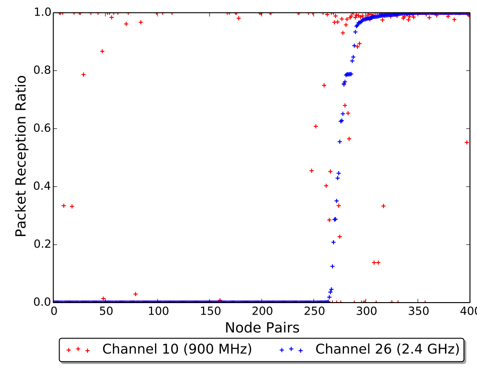
We plot these standard deviations for the two channels in Figure 2.6 to study how dynamic the channels in the two bands are. We observed that intermediate quality links were the most dynamic, uncorrelated with distance and quite unstable both in the 2.4 GHz band and 900 MHz band. Poor-quality links show small dynamics than the intermediate links. The good links were the most stable. Our observation regarding the relative stability of good links compared to intermediate links confirms previously reported results [22]. Finally, we also observed that good links in the 900 MHz channels were more stable than the good links in 2.4 GHz channels. Overall, we find that 900 MHz channels provide more connectivity that is more stable over time compared to the channels in 2.4 GHz band.

2.4 Measurement Results

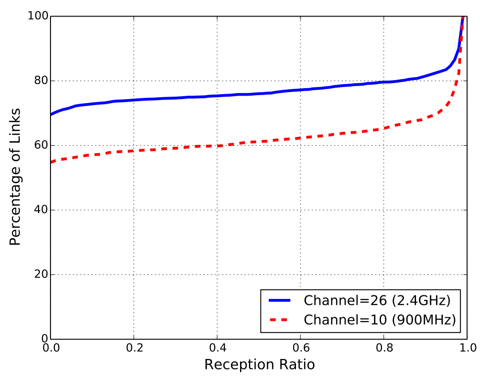
Different testbeds have different PRR distribution due to their deployment topology and physical environment. Figure 2.7 shows the PRR across two bands across two testbeds. We collected link PRR on Twonet and CSIRO testbeds for 24 h. We show the PRR for all node pairs sorted by the PRR on channel 26 in Figure 2.7(a) and 2.7(b). The data allow us to compare the PRR on channel 10 and channel 26 for a given node pair. On Twonet, the blue dots (PRR on channel 26) were mostly on the bottom-left side of the red dots (PRR on channel 10) suggesting links formed between a given node-pair on channel 10 is better than the link on channel 26 between the same node-pair. On CSIRO, the PRR on channel 10 is still better than the PRR on



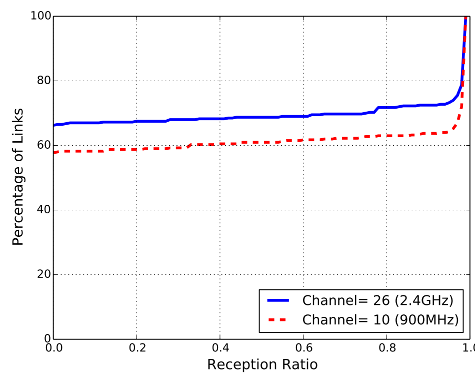
(a) PRR in two bands sorted by PRR on channel 26 on Twonet testbed.



(b) PRR in two bands sorted by PRR on channel 26 on CSIRO testbed.



(c) CDFs of PRR for Twonet testbed.



(d) CDFs of PRR for CSIRO testbed.

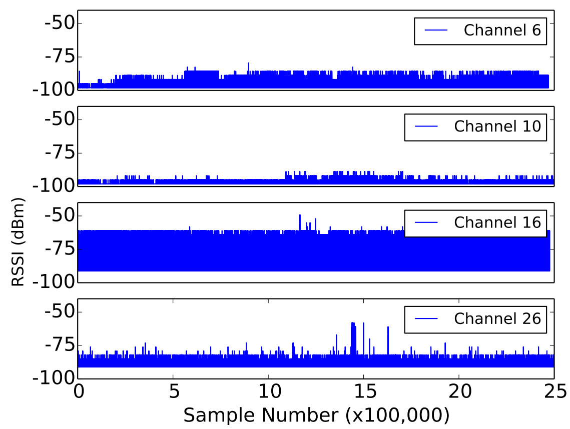
Figure 2.7: Comparison of PRR across the two testbeds in two bands. Higher PRR of 900 MHz channels on both the testbeds.

channel 26 but the difference was not as large as it was on Twonet. We summarize this dataset using CDF in Figure 2.7(c) and 2.7(d). Channel 10 in the 900 MHz band had 15% more links than Channel 26 on Twonet and around 10% more links on CSIRO Testbeds. Both channels had less than 20% intermediate links on both the testbeds. There are fewer good links (as a fraction of all the links) on Twonet compared to CSIRO in both the bands.

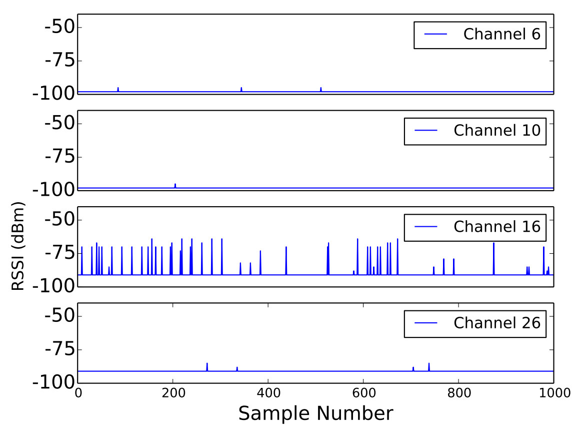
2.4.1 Channel Noise and Burstiness

In earlier sections, we studied the long-term properties of the channel using measurement probes sent at several second intervals. Now, we study the short-term properties of the wireless link. The burstiness of the wireless channel in the 2.4 GHz band is dominant especially in the channels that overlap with the Wi-Fi channels, such as channel 16. We extended our observation to not only the burstiness of 2.4 GHz but also to the burstiness of 900 MHz at the same time.

To sample noise on the channel, we programmed the nodes to sample the RSSI continuously at 100 Hz. We configured both the radios to run with OQPSK modulation scheme and 250 kbps data rate. We sampled RSSI on channels 6, 10, 16, and 26 on Twonet testbed for 24 h collecting almost 2.5 million samples. The channel assignment for the measurement followed the pattern described in Table 2.2. Figure 2.8 shows the RSSI on the four channels over 24 h. We observed that channel 6 and 10 had less ambient noise than channels 16 and 26 and confirmed the short-term observations as shown in Figure 2.3. The noise patterns were largely consistent over



(a) RSSI traces over 24 hours



(b) RSSI traces over 10 seconds

Figure 2.8: RSSI samples on four channels.

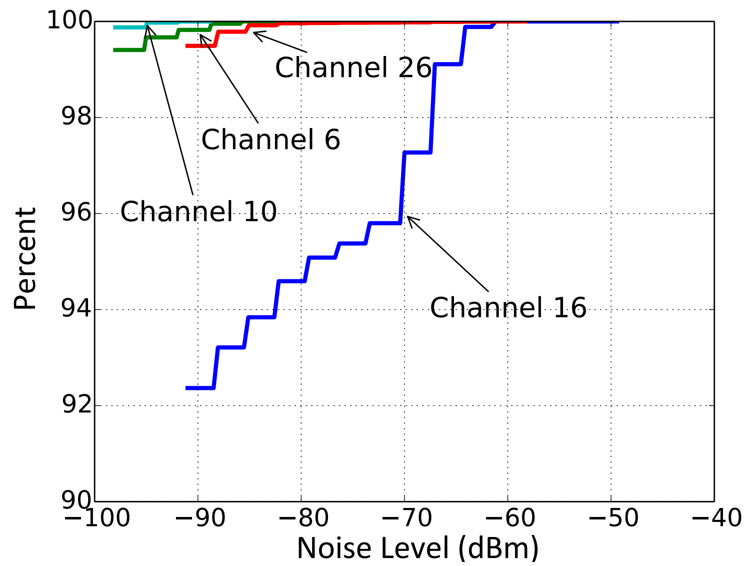


Figure 2.9: CDFs of noise level for four channels in two bands.

the long term. Figure 2.8(b) shows 1000 noise samples on the four channels. We found that channel 16 was bursty as expected. Rest of the channels were less bursty.

Figure 2.9 shows the CDFs of RSSI sample values in the four channels sampled concurrently. We find that a large fraction of RSSI samples on channel 16 range from -91 dBm to -70 dBm, which is 25% of the full dynamic range supported by the RF231 radio. Most RSSI samples on channel 6 fall on a narrower range of -98 dBm to -95 dBm, which is 3% of the full dynamic range supported by the RF212 radio. For channel 26, the RSSI range is -91 dBm to -79 dBm, which is 15% of the full dynamic range of the radio. For channel 10, the RSSI range is -98 dBm to -92 dBm, which is about 7% of the full dynamic range of the radio. We summarize these results in Table 2.3.

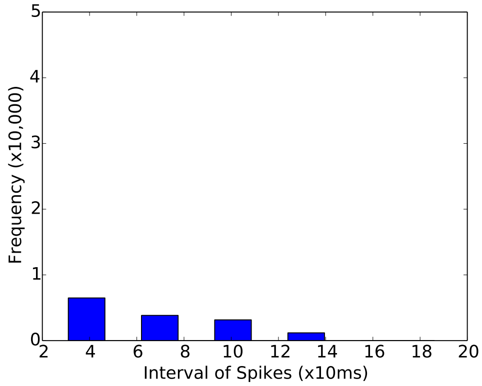


Figure 2.10: Spike interval distribution on channel 6 on Twonet testbed.

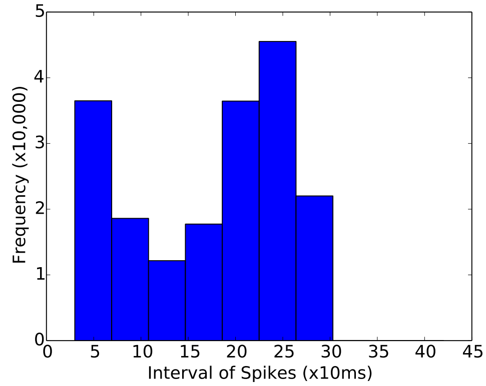


Figure 2.11: Spike interval distribution on channel 16 on Twonet testbed.

Table 2.3: Summary of RSSI distributions in two bands

Channel	6	10	16	26
Band	900 MHz	900 MHz	2.4 GHz	2.4 GHz
Avg. of RSSI (dBm)	-97.997	-97.994	-90.604	-90.984
Std. of RSSI (dBm)	0.0997	0.142	2.237	0.260

Next, we study how often these bursts occur in the channels. We define a spike as a sample which is larger than the minimum sensitivity of the transceivers: -91 dBm for RF231 and -98 dBm for RF212. We then compute the interval between the spikes and plot the histogram in Figure 2.10 and Figure 2.11 after filtering the intervals larger than 450 ms. The figure shows that the spikes on channel 6 (avg. 60 ms) occur more frequently than in channel 16 (avg. 170 ms) when observed through our 100 Hz probes.

2.5 Summary

In this work, we presented results from our study of the 900 MHz and 2.4 GHz bands typically used in low-power wireless networks. We found that the 900 MHz band provides 15% more connectivity compared with the 2.4 GHz band. Compared to prior work that has also done some studies of these two bands, we made concurrent measurements on multiple channels. Based on the insights from our work, one could design protocols to increase network reliability by taking advantage of the two bands. We performed our measurements on two different testbeds, which gives us confidence about the validity of the results.

Chapter 3

Crosstalk-based Communication: Utilizing RF interference for Useful Communication

Wireless sensing and control applications are increasingly being deployed in our homes and environments to enhance comfort for the occupants [33], understand activity and energy use in a home [47, 77], increase energy efficiency [34], and allow better automation and control [35]. Many of these applications require users to interact with the sensor or control devices. For example, the user may want to control the light or thermostat in the house. The user may use a smartphone to perform such control actions. The control actions are conveyed to the wireless sensors or controls through the Internet. Some smart-home automation applications are non-interactive. Yet, they require Internet access either to upload the data or download configuration information. Thus, in many scenarios, the wireless sensor and control devices in a smart home, office, or environment require communication to or from the Internet.

The existing solution to enable such a communication is through a gateway or a bridging device. At a high level, the device is a router. It has one 802.15.4 interface to communicate with the 802.15.4 network. The other interface may be WiFi or wired. The device shuttles traffic between the two interfaces. A control message (e.g., to turn a light on) coming from a smartphone app, travels to the gateway (possibly through the Internet), is translated appropriately for 15.4 network, and is transmitted by the 15.4 radio. On the software side, bridging may happen at the application layer (with custom application-specific messages) or at the network layer (with standardized network layer protocols). Recent IETF standards such as 6LoWPAN [83] support development of sensor networks with this architecture. This architecture, which we call gateway-oriented architecture, has served us well as evidenced by vibrant ecosystem of smart-home devices and companies that sell those products. Despite some application deployments using WiFi-based sensors and controllers, 802.15.4 or low-power low-rate radios occupy a unique point in the price and design space that they are likely to be a radio of choice for many years to come.

In this work, we challenge the premise behind the gateway-oriented architecture: that to enable WiFi devices to send messages to 15.4 devices, we need to build a gateway with the two interfaces. While modern gateway devices provide additional functionalities such as local storage service, the gateways that ship with the 15.4 devices are primarily used to bridge between the Internet and the 15.4 network. We propose to eliminate the gateway from the network and enable WiFi devices to directly communicate with the 15.4 devices. If this is possible, we would significantly simplify the deployments and reduce the device and maintenance cost of the networks.

The core idea in our approach is to have a WiFi device transmit packets with special patterns representing the information to be conveyed to the 15.4 network. The transmission is done on a WiFi channel that overlaps with the 15.4 channel on which 15.4 devices are listening. The 15.4 devices sample the signal on the channel due to WiFi transmissions (which we would typically call crosstalk or interference and try hard to avoid) and interpret the information in the pattern. We call this technique *crosstalk-based communication (CTC)*.

Building such a modulation and demodulation scheme to enable communication from WiFi devices to IEEE 802.15.4 devices using crosstalk has two main challenges. First, WiFi channels and IEEE 802.15.4 channels are allocated for different frequencies, though the frequency bands partially overlap. Transmissions on the overlapping channels result in crosstalk and interference rather than the communication of data. Second, direct communication requires both the devices to perform modulation and demodulation, compared to the gateway-oriented solution, in which the gateway does the modulation or demodulation using the radios designed for the specific frequency band. The demodulation, especially on the 15.4 devices has to be efficient in both power and computation. Any system we design must not only overcome these challenges but also offer at least a modest but usable data rate, for example, sufficient for device configuration or commands.

We have designed and implemented the proposed system on multiple WiFi devices (laptop with WiFi interface, and an OpenWRT-compatible wireless AP) and on two mote platforms (TelosB and Opal). We find that the proposed technique can be used to successfully send messages from WiFi devices to 15.4 devices. Even in uncontrolled environments with other APs and Bluetooth in a residential environment, we were able to achieve a data rate of up to 2 bytes per second with less than 10% bit error rate.

We make these contributions in this work:

- We presented the first crosstalk-based primitive to enable communication between WiFi devices and IEEE 802.15.4 sensor nodes without a physical gateway. The primitive is a novel modulation scheme that runs on WiFi devices and a demodulation scheme that runs on the 15.4 devices.

- We implemented the proposed technique on real WiFi and 15.4 devices and performed experimental validation of the techniques in both controlled anechoic chamber and in uncontrolled environments. The system achieved a data rate of 2 bytes per second with less than 10% bit error rate in uncontrolled environments.

3.1 Related Work

We briefly review work related to Internet connectivity to sensor networks and study of cross-technology issues in wireless networks.

Connecting to the sensor and control devices from the Internet. Most interesting and useful sensor network and control applications require them to be connected to the Internet for configuration or data access. Some sensor networks use WiFi radios. These networks directly connect to the Internet. Many sensor and control networks use low-power radios, such as the 802.15.4-compliant radios. Gateway devices are typically used to bridge those networks. There have been two major efforts on this front. The first and slightly outdated method uses a various application or other types of gateway devices built over serial, USB, or Ethernet hardware interface to a gateway device. Classic TinyOS serial forwarder protocol is an example of this approach. A more modern approach is to use a standardized protocol, such as 6LoWPAN [21, 50, 83, 93], over the serial or other interface, so the gateway essentially becomes a network-layer routing device. Regardless of the layer at which the message switching occurs, the gateway device needs a 15.4 radio and a wired or WiFi interface where Internet devices may connect. On research projects, it is common to connect a TelosB [74] or other mote to the computer and use the computer as a gateway between the Internet and the sensor network. In commercial products, the gateway often is a standalone device that connects either to the home router by Ethernet cable or by WiFi. Chebrolu *et al.* [28] investigated the feasibility of the unidirectional communication from 802.11 devices to 802.15.4 devices. But no system implementation or evaluation has been conducted based on their experience.

In this work, we design and implement a technique to connect to the sensor and control devices from the Internet without using a separate physical gateway.

Wireless Interference. There has been a large body of work in understanding wireless interference, either within sensor networks or cross-technology interference. Gollakota *et al.* [37] presented a decoding methodology to make 802.11n network robust under the presence of high-power cross-technology interference. The system can decode messages even when receiving interfering signals from other technologies, allowing devices from different technologies to coexist. Hithnawi *et al.* [46] presented a real-time approach to detect and mitigate cross-technology interference. Hauer *et al.* [52] introduced an interference detector which was capable to distinguish different types of interference as well as WiFi beacons. Hermans *et al.* [42] also presented a system which can detect different interferers by observing the disrupted 802.15.4 packet. Hauer *et al.* [41] investigated how to estimate bit error positions in a corrupted packet based on RSSI temporal variations. All the listed papers here assume WiFi activity can corrupt bits in an 802.15.4 packet and design techniques to survive from such interference [68, 115]. There is another body of work that tries to understand the performance of links on different channels [86, 106]. Many such studies empirically studied the performance on channels that also overlap with WiFi thus quantifying the negative impact of WiFi traffic on packet transmission performance on the 15.4 links. In our work, rather than looking at interference and crosstalk as

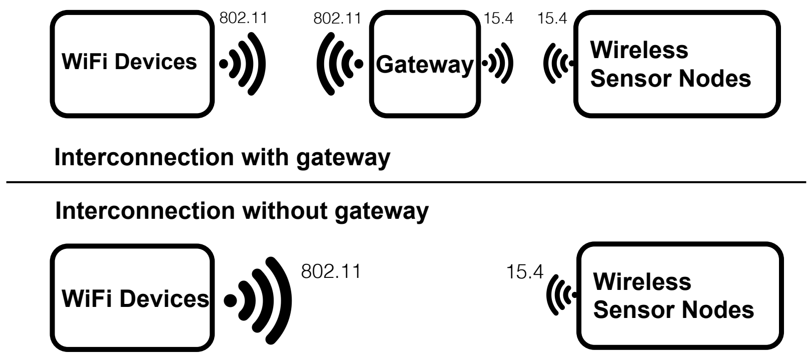


Figure 3.1: Difference between the prevalent approach that uses the gateway device and the proposed approach that does not use the gateway devices for WiFi devices to communication with the wireless sensor nodes.

a nuisance, we use it to enable communication between WiFi and 15.4 radios.

New Wireless Communication Channels. Recently, new types of wireless channels have been developed for use in sensor networks. For example, Liu *et al.* [70, 112] presented a design for communication using only ambient RF by backscattering the ambient RF. There are also interesting work on developing Visual Light Communication channels for communication in wireless sensor networks. For example, Giustiniano *et al.* [36] and Wang *et al.* [97] created a visual light communication system with a fully functional Linux-based PHY and MAC layer implementation. Rajagopal *et al.* [75] enabled light communication for low-power embedded devices by utilizing cameras on consumer devices. They achieved a data rate of 1.25 bytes per second. These are examples of research developing a new medium for wireless communication. In a similar spirit, in this work, we design and implement CTC between 802.11 and 802.15.4 by utilizing crosstalk between the two technologies.

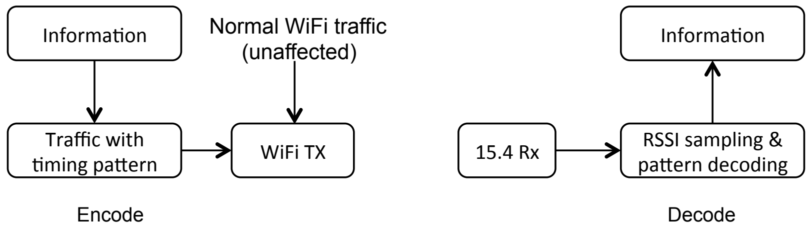


Figure 3.2: Components of the proposed communication system that utilizes crosstalk between 802.11 and 802.15.4 channels.

3.2 Design

In this section, we present the design of our system that allows direct communication from a WiFi device to 802.15.4 networks without using a physical gateway device.

3.2.0.1 System Architecture

Our goal is to allow WiFi devices to send messages to the devices that use the 802.15.4 radios. Thus, the users of our system consist of devices in these two networks. First, the devices that operate in 802.11 networks. For example, iOS and Android-based phones, the wireless adapters used in the laptop, or wireless-access points operating in 2.4 GHz frequency band. Second, we also have the platforms deployed in 802.15.4 networks. These are typically low-power devices with transceivers operating in 2.4 GHz frequency domain. Examples of such devices include TelosB as research platforms or smart gadgets in smart homes. As shown in Figure 3.1, the main difference between the prevalent approach and our approach is we enable the communication between these two sets of devices without the gateway device.

The basic idea of our approach is to make use of the cross-technology interference to encode and decode information. Figure 3.2 shows the main components that make this type of communication possible. Information is encoded as special timing patterns of UDP packet frames. The idea is inspired by Lee *et al.* [62] on covert timing channel in which they control and access every bit transmitted in the physical layer. Such precise timing pattern was implemented on a highly customized NICs with a wired network. They created the covert channel by controlling inter-packet delays to guarantee the network security. In our work, the packets were sent over commodity WiFi interface of an AP or other wireless devices with no such precise timing control on the inter-packet delay nor any change in device drivers. The 802.15.4 receiver samples RSSI on the overlapping channel and decodes the timing pattern. The timing pattern represents the information, which was passed to the application. In the following sections, we describe each step in more detail with the design nuances and tradeoffs.

3.2.0.2 Utilizing Crosstalk Between 802.11 and 802.15.4

Our approach takes advantage of cross-technology interference that exists between the 2.4 GHz channels and the 802.15.4 channels. The WiFi transmitter does nothing special at the physical layer to encode information using the crosstalk. At the physical layer, the transmission looks like the transmission of any other packets. The 802.15.4 receiver, however, is not designed to receive packets from 802.11. So, a regular packet-reception mechanism does not work. Instead, the receiver samples the RSSI on the

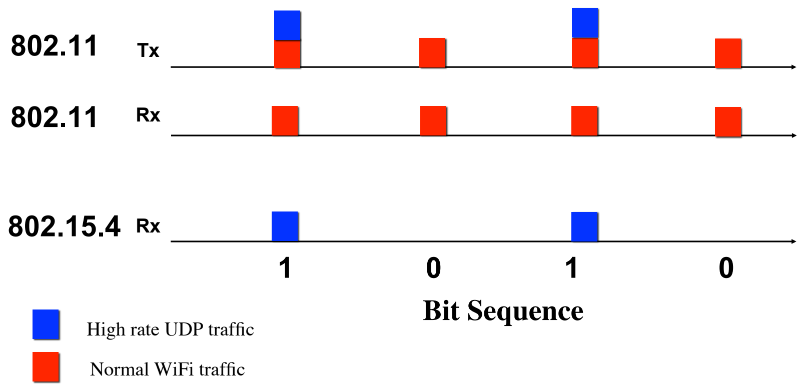


Figure 3.3: Transmission from WiFi devices to IEEE 802.15.4 sensor nodes on crosstalk channels. The sensor nodes can detect the presence or absence of high-rate UDP traffic on the channel even though they cannot receive the normal WiFi packets. These signals can be used to encode information. In this example, presence or absence of high-rate UDP traffic on the channel is used to decode the bit string "1010".

channel at a few KHz. The signals transmitted in 802.11 channels can be received (even though the packets cannot be decoded) in the nearby 15.4 channels (Figure 3.4). Such transmissions cause the 15.4 channels to be many times saturated with the signal. This leaked signal can be detected through background RSSI sampling on the 802.15.4 transceiver. We can modulate and demodulate these crosstalk signals based on the leaked signal characteristics. For example, in our system, we modulate the leaked signal to enable the communication from WiFi devices to IEEE 802.15.4-based sensor node (Figure 3.3).

3.2.1 Modulation by WiFi Devices

WiFi devices modulate the crosstalk signal to send information to the sensor nodes. The information is encoded as timing patterns (on-off). The code itself is represented by controlling the presence and absence of high-rate UDP packets on the WiFi channel. The presence of high-rate UDP packets is defined as *One*. The absence of high-rate UDP packets is defined as *Zero*. For accurate modulation, the timing of the traffic patterns needs to be accurate. In a general-purpose operating system, maintaining accurate timing on the outgoing WiFi interface requires accurate time-stamping. For our experimentation, we build a packet generation tool. Our packet generation tool is similar to $iperf$, but it can generate high-rate UDP packets with microsecond-level accuracy.

Using the packet generation tool, we can send back to back packets to achieve a maximum packet rate of nearly 3000 packets per second. Each packet has 1500 bytes. In the best case, if we send one packet to saturate the channel (indicating a '1') and wait for one packet to indicate a '0', we can theoretically achieve a data rate of 3 kbps with level triggering technique. However, sending one packet will only take 300 μs. This symbol rate will typically be too fast for sensor nodes to decode successfully without errors. The decoder would need to be synchronized and perform high-speed channel sampling. Thus, in our system, we use much lower symbol rate so even a modest sensor platform such as a TelosB or an Opal mote can decode the information correctly.

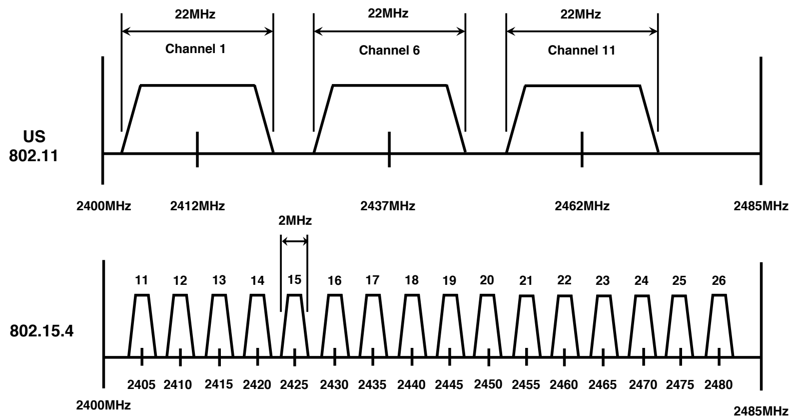


Figure 3.4: Map of 802.11 and 802.15.4 channels. These two sets of channels overlap with each other and cause crosstalk.

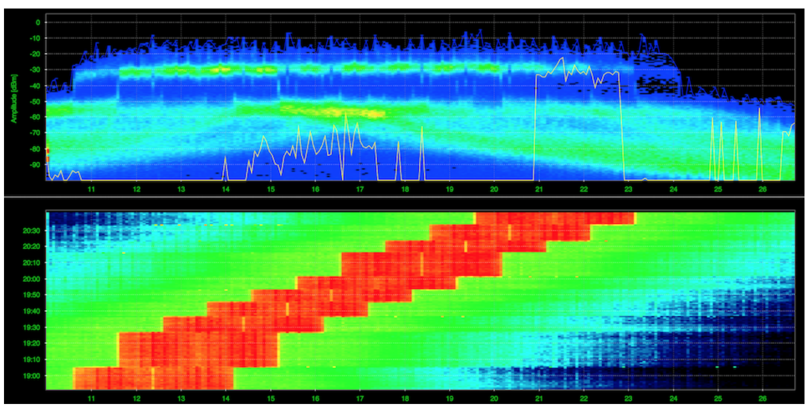


Figure 3.5: Screen shot taken from Chanalyzer (a tool for visualizing wireless landscape) using Wi-Spy 2.4x tool during the measurement study in the anechoic chamber. The figure shows quiet channels other than the ones used for WiFi transmissions (red).

3.2.2 Demodulation by Sensor Nodes

We now describe how the sensor node detects the channel and decodes the information on that channel.

3.2.2.1 Channel Detection

There are two models for how the sensor node decides on the channel to use for reception. The first model is manual configuration. This approach is similar to how we configure many WiFi or sensor devices. For example, when we program sensor devices, we set the radio channel. Similarly, in our system, we can manually configure the sensor device to listen for messages from the WiFi network on the 15.4 channel with the largest overlap with the WiFi channel.

The second model uses automatic detection of a channel. We performed several experiments to collect data and provide heuristics to detect the channel used for communication. At a high level, WiFi transmitter sends a known pattern of signals on the channel. The sensor node receiver cycles through all the channels to receive the stated pattern. To test the feasibility of this technique, we performed RF experiments in an anechoic chamber. In this experiment, we had one laptop transmitting packets back to back on WiFi channels 1-11. We used Wi-Spy [18], a portable USB spectrum analyzer, to collect the wireless signal in 2.4 GHz frequency band and visualize them with Channalyzer (Figure 3.5). We also had 16 TelosB motes tuned to 15.4 channels 11-26 sampling their respective channels at 4 KHz. Figure 3.6 shows the results of measurement study. It shows that whenever a device transmits on a WiFi channel, the few motes with their radio operating on the channels overlapping with the WiFi channel can successfully sample the channel and detect the signal. The other channels were relatively quiet. Thus, if we cycle through the channels when there are known signal patterns, we may be able to detect channels to be used for reception at least in

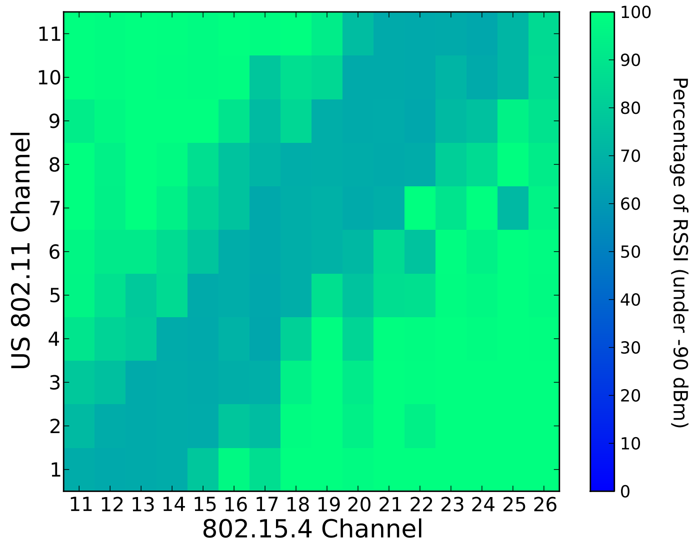


Figure 3.6: Signal on the channel sampled by the sensor nodes with WiFi transmitters transmitting on all the channels in an **anechoic chamber.** Each cell represents an average from 11 rounds of 65,536 measurements.

a controlled or a quiet environment. In an uncontrolled environment, this heuristics will not work reliably as demonstrated by the second round of measurement studies, which we describe next.

In the second study, we repeated the same measurements but in an apartment building. There were other WiFi and Bluetooth devices and hence may bleed into the channels used for crosstalk-based communication (CTC). Figure 3.7 shows the results of these measurements. Although, the pattern has some similarity to the pattern from the controlled environment, there is one important difference: the blue vertical bands indicate certain channels are saturated (from the perspective of the 802.15.4 devices) regardless of the channel used by our WiFi transmitter. This is due

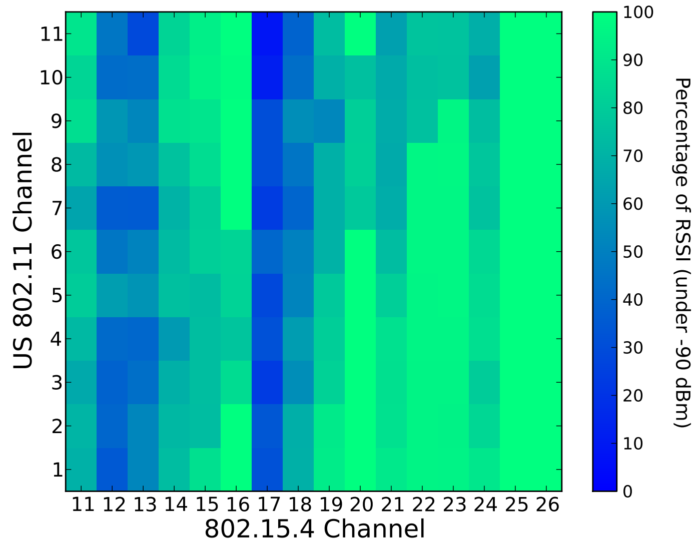


Figure 3.7: Signal on the channel sampled by the sensor nodes with WiFi transmitters transmitting on all the channels in a **residential building.** Each cell represents an average from 11 rounds of 65,536 measurements.

to WiFi routers using channels 1 and 6 in the building. In the residential apartment, most wireless APs were operating on channel 1 and channel 6. Thus, simple stateless channel scanning alone will not be able to detect the channel used for CTC in this uncontrolled setting. A robust preamble or channel detection code will be required so the sensor receiver and the WiFi transmitter converge on a channel. An alternative is to perform aggregate analysis (e.g, CDF) of the signals sampled on the channel. Figure 3.8 presents the CDF of sampled RSSI from channel 15 to channel 20 from the study in the uncontrolled environment with WiFi transmitter on channel 6 in 802.11 networks. We found that more than 90% of RSSI samples on channel 15 and 20 are close to -100 dBm or lower. The number is 70% for channels 15, 16, 17, and 18. Thus, these four channels may be good candidates for

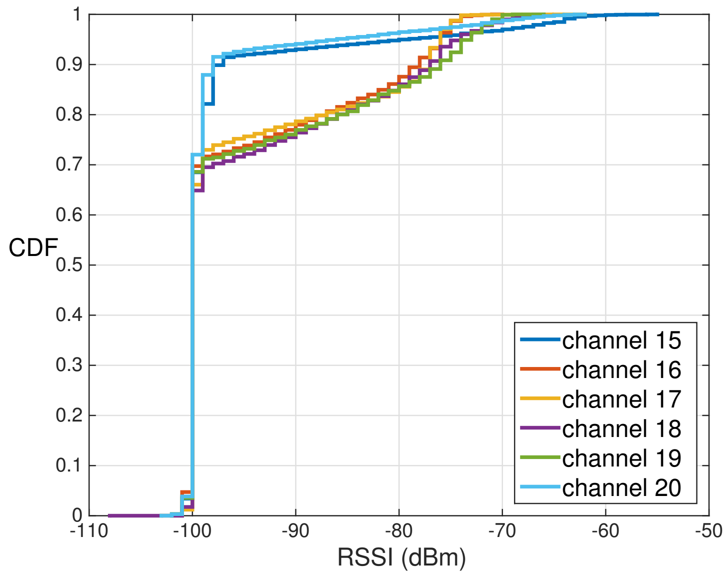


Figure 3.8: CDF of RSSI on 802.15.4 channels 15-20 with WiFi transmitter on 802.11 channel 6.

crosstalk-based communication (CTC) with WiFi channel 6. Further measurements could narrow down the set of good channels, in this case, channel 17 which has more interference. Thus, in our approach, we tried to find the channels that offer the most interference (in contrast to interference avoidance work that tries to find the channels with the least interference). We have also empirically established that for manual configuration approach, we should use 802.11 channel N together with 802.15.4 channel N+11. This rule also matches the inferences shown on standard channel maps such as the one in Figure 3.4.

Thus, with measurements in an anechoic chamber and an uncontrolled environment, we test the feasibility of channel scanning to find the channel for communication. We also note that manual configuration of channels is the most reliable way to

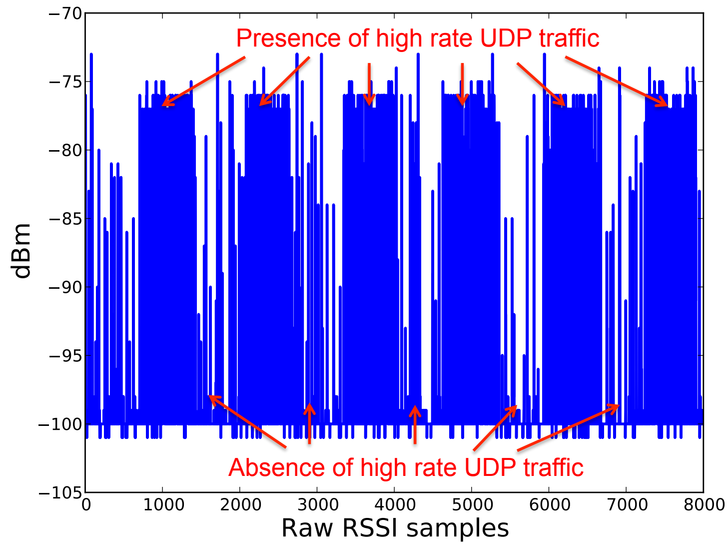


Figure 3.9: Raw RSSI values sampled by a TelosB mote on channel 17 with WiFi transmission on channel 6. The spikes during the absence of the high-rate UDP traffic are caused by normal WiFi usage, e.g, web browsing, video streaming.

synchronize the channels similar to how we configure many sensing systems today.

3.2.2.2 Signal Decoding

During a reasonably strong WiFi transmission, the 15.4 transmitter typically gets saturated. Thus, the RSSI samples showed a pattern consisting of small values (when there are no WiFi transmissions) and high value (when there are WiFi transmissions). Figure 3.9 shows a sample RSSI trace captured at 4 KHz by a TelosB mote during a WiFi transmission with our encoding scheme. We observe a lot of raw RSSI spikes during the absence of the high-rate UDP packet. It is due to normal WiFi traffic indicated in Figure 3.2. The raw RSSI is a reflection of both the high-rate UDP

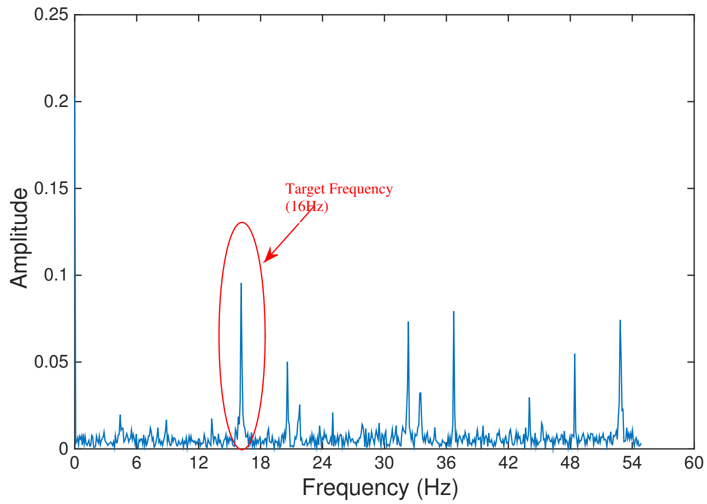


Figure 3.10: FFT for RSSI traces sampled by a TelosB mote on channel 17 with WiFi transmissions on channel 6.

streams and normal WiFi streams when the wireless AP is transmitting a wireless signal. We can identify the periodic on and off patterns in the raw RSSI values. Figure 3.10 plots the FFT of the time series signal. The largest peak corresponds to the periodicity our WiFi-based modulation for crosstalk-based communication. This result provides evidence about the feasibility of detecting WiFi signals modulated by UDP packets with a 15.4 radio. Given the feasibility, we now design two strategies to demodulate the crosstalk signals without incurring high-memory overhead.

Strategy 1: Minimum RSSI Fraction. 802.15.4 wireless transceivers report minimum RSSI values if the received signal is below or equal to the sensitivity. Strategy 1 basically applies the minimum RSSI Fraction as an indicator to distinguish between presence and absence of high-rate UDP packets.

Assuming that the CTC-data rate and the RSSI sampling rate is known, the

Algorithm 1 Decoding Algorithm

Input: **RssiSamples, WindowSize, Strategy** in

Output: **RssiList** out

```
if (Radio = CC2420) then
   MINRSSI = -101
else if (Radio = AT86RF230) then
   MINRSSI = -91
end if
   Initialization :
create queue with size equal to the WindowSize
   LOOP Process
if (Strategy = Min.RSSIFraction) then
   for item in RssiSamples do
      if queue is not full then
         enqueue item
      else
         minRssiFrac = queue.count(MINRSSI)/WindowSize
         RssiList.append(minRssiFrac)
         dequeue queue
         enqueue item
      end if
   end for
else if (Strategy = AverageRSSI) then
   for item in RssiSamples do
      if queue is not full then
         enqueue item
      else
         avgRSSI = avg(queue)
         RssiList.append(avgRSSI)
         dequeue queue
         enqueue item
      end if
   end for
end if
return RssiList
```

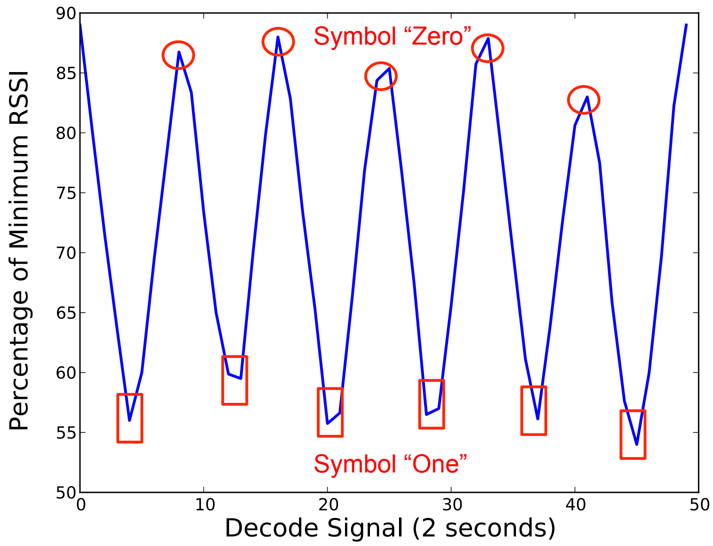


Figure 3.11: Sensor node decoding the WiFi signal using the *Minimum RSSI fraction* strategy.

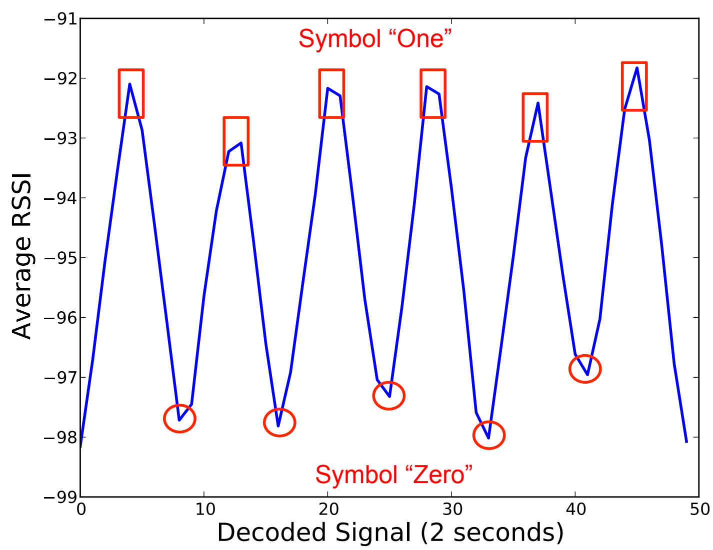


Figure 3.12: Sensor node decoding the WiFi signal using the *Average RSSI* strategy.

window size is configured to be:

$$\text{window size} = \frac{\text{sampling rate}}{\text{data rate}\times\text{sliding steps within the window size}}$$

Within each window, we first find the smallest RSSI value, which is similar to the CCA algorithm proposed in B-MAC [73]. Then, we calculate the minimum RSSI fraction over the window size. Intuitively, on a quiet channel, this fraction will be large, i.e. a symbol 0. On a busy channel, this will be small, i.e. a symbol 1. The minimum RSSI value is the constant which represents the smallest value the wireless transceiver can report. Figure 3.11 shows the result decoded by this strategy. Algorithm 1 shows the details of this technique.

Strategy 2: Average RSSI. Similar to minimum RSSI fraction, the average RSSI method also uses the same window size and computes the average RSSI for each window. Based on the average RSSI, we find the peak, i.e. "1", and the valley, i.e. "0", to decode the information. Figure 3.12 shows the result decoded by this strategy. Algorithm 1 shows the details of this technique.

3.3 Evaluation

In this section, we evaluate the proposed communication technique.

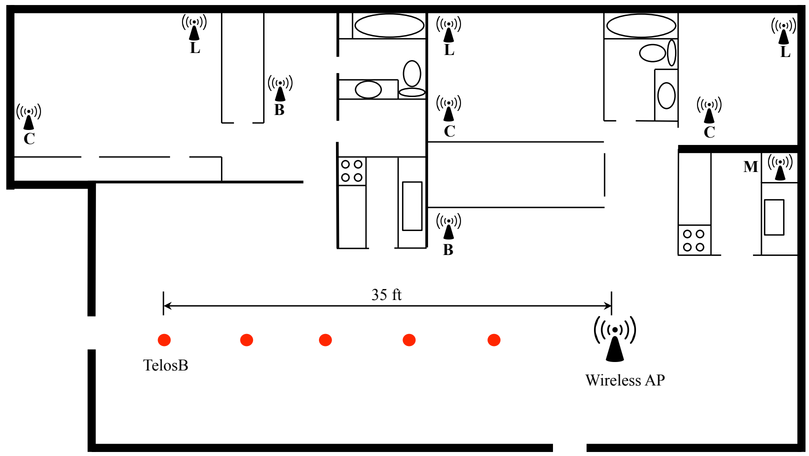


Figure 3.13: Settings for experimental evaluation in a residential apartment with a few devices operating in 2.4 GHz frequency band. (M for microwave, B for bluetooth speakers, C for cellphone, L for laptop.)

3.3.1 Metrics and Settings

We used BER (Bit Error Rate) as the primary metric to evaluate the system reliability. We perform experiments in both residential and office-like environments since these areas are equipped with a lot of WiFi devices creating a challenging environment for our communication system. Figure 3.13 shows the residential setting used in our experiment. This is an apartment with a microwave oven, a wireless AP as well as portable devices such as cellphones, tablets, laptops, and several Bluetooth speakers. All these WiFi devices are connected to the wireless AP for Internet access. By experimenting in this uncontrolled environment, we can test the robustness, and reliability of the system.

3.3.2 Rate of generated UDP Packets (Packet Rate)

Our system uses generated UDP packets to modulate signals, but the artificial traffic could negatively affect the normal use of WiFi network. So our goal is to generate the traffic with the optimal 802.11 packet rate while maintaining high reliability. In order to achieve this goal, we evaluated our system in a real WiFi network scenario. We generated traffic when video streaming, web browsing, and online gaming sessions were taking place by the residents. Figure 3.14 shows the BER achieved at different packet rates, which is correlated with the symbol rate. During the experiment, we enabled the wireless AP for Internet access, then started generating bit sequence from the devices connected to this wireless AP. To control the environment settings, only one associated device was allowed. We ran the Wireshark packet capture tool on this associated device, which was a MacBook Pro with an Intel i5 CPU. We captured all the incoming 802.11 packets on the wireless interface. Figure 3.14 shows that the packet rate can directly affect the system reliability. With 980 packets per second, it is possible to achieve a BER of less than 10%.

3.3.3 Packet Rate

We configured the CTC-data rate as 16 bits per second. As we can see in Figure 3.14, as the 802.11 network traffic went up, the Bit Error Rate significantly decreased to less than 10% and tended to be near 0%. However, the 802.11 network traffic can have has a major influence on the network performance of WiFi network users. We

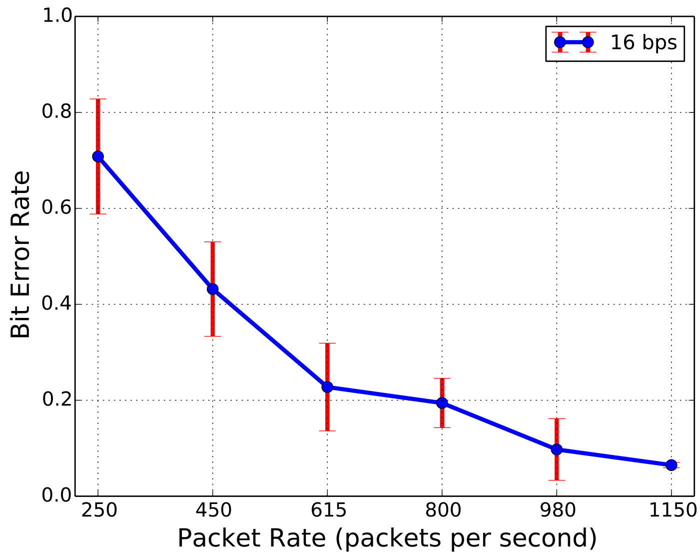


Figure 3.14: BER vs UDP packet rate with a CTC date rate of 2 bytes/s.

even tested the network performance when setting the packet rate up to 1600 packets per second. We recommend 1000 packets per second as the optimal rate for the UDP packet generated by our system. Compared to the BER in lower packet rate, this setting provided stability and high-decoding accuracy.

3.3.4 RSSI Sampling Rate

The sensor node samples the channel to interpret the symbols. A lower sampling rate is less costly in hardware resources and energy. A higher sampling rate makes the system more robust and potentially allows a higher data rate but at a hardware or energy cost. We performed experiments with different RSSI sampling rates on the motes under two settings to determine the best sampling rate. The first setting is

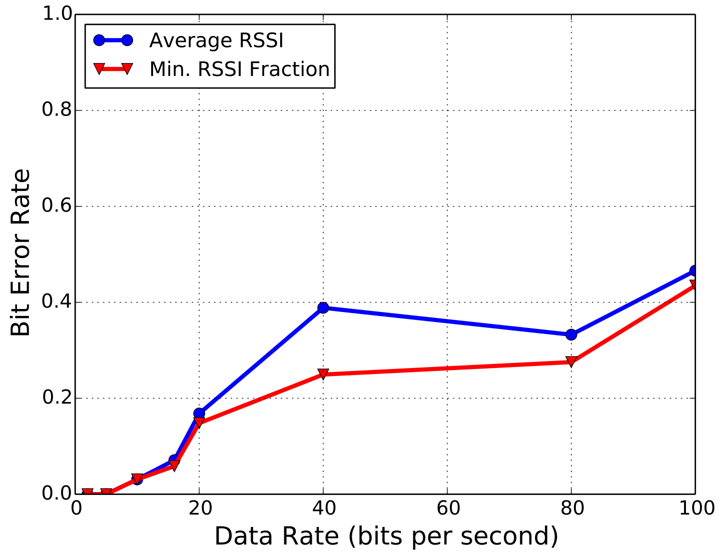


Figure 3.15: BER vs CTC-data rate using different decoding strategies.

called *WiFi with Internet Connection*, in which case, the WiFi AP had normal WiFi users performing browsing and other activities. The second setting is called *WiFi without Internet Connection*, in which case, we unplugged the uplink cable from the WiFi AP and thus AP provided only local connectivity with no Internet access. Table 1 shows the results from our experiments in these two settings. We found that under *WiFi without Internet Connection*, low-sampling rate, e.g. 2 KHz, was sufficient to achieve a low BER. With normal WiFi traffic (*WiFi with Internet Connection*), we needed a higher-sampling rate to achieve a low BER. Overall, higher-sampling rates were better when the WiFi AP was serving other normal WiFi users and also modulating the information for the CTC.

Table 3.1: BER in two settings: WiFi with and without Internet Connection

Internet Connection	2 KHz (avg.)	2 KHz(std.)	4 KHz(avg.)	4 KHz(std.)
No	2.71%	0.71%	2.18%	1.74%
YES	17.15%	1.59%	10.51%	1.69%

3.3.5 Decoding Strategies

We evaluated the reliability of the crosstalk-based communication (CTC) with different decoding strategies. We modulated the high-rate UDP traffic into five continuously increasing the data rate. For each modulated data rate, we decode them with different strategies to evaluate their performance. Figure 3.15 presents the BER of the two decoding approaches under different rates at which the WiFi device sends information to the mote using CTC. For CTC-data rate less than 16 bps, the two approaches have almost the same bit error rate, which is near 0. However, for CTC-data rate that is larger than 20 bps, using the *minimum RSSI Fraction* decoding method results in a lower BER.

3.3.6 Platform Independence

Next, we evaluate if our system works on multiple platforms both on the 802.11 and 802.15.4 networks. For the 802.11 network, we tested the CTC system on wireless AP and a laptop. For sensor nodes, we tested the CTC system on TelosB and Opal motes. We connected both TelosB and Opal motes to a 10 port USB hub which was connected to a laptop. We programmed both the platforms with an application that

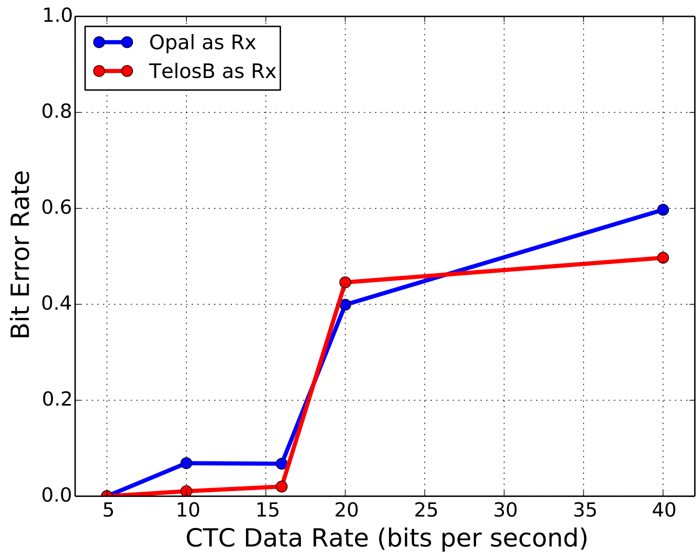


Figure 3.16: BER vs data rate using different sensor nodes.

sampled the RSSI at 4 KHz. The motes sampled the RSSI and saved them to the local storage. We later sent this data to the laptop for data analysis. Figure 3.16 compares the decoding performance for different platforms. For CTC-date rate up to 16 bps, Opal provided communication with a BER less than 7% while TelosB provided more reliable communication with a BER less than 2%. However, increasing the CTC data rate to more than 16 bps caused the BER to become unacceptably high.

In the next experiment, we used a laptop as our WiFi transmission device. We run our packet generation tool on this laptop generating UDP packet patterns that encoded the information we want to transmit using CTC. The destination IP of these unicast UDP packets was set to be another laptop associated with the same wireless AP. While the destination IP of these packets was another laptop, the motes were

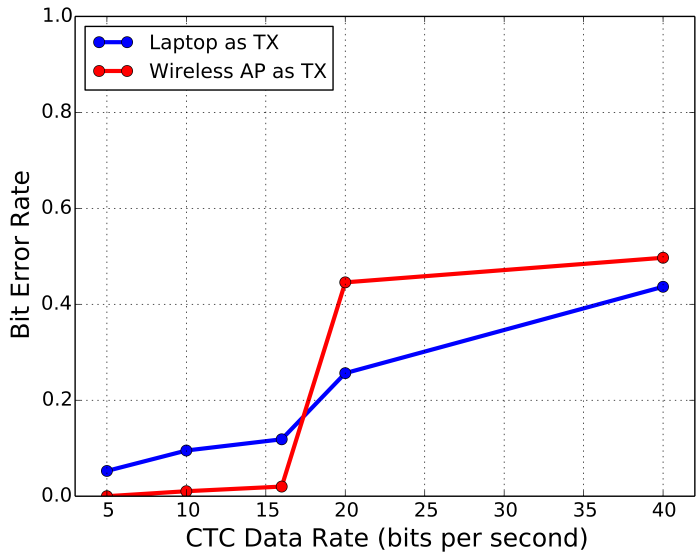


Figure 3.17: BER vs data rate using different WiFi devices.

able to decode information embedded in the patterns of these UDP packets using CTC. We compared the CTC-data rate achieved by the system when there were other active normal WiFi users in the network as seen in Figure 3.17. We found that the CTC was more reliable if the modulation was conducted by a wireless AP. Our guess was due to APs being specialized hardware for WiFi packet reception and transmission, they provided better control in timing and signal strength in packet transmissions.

3.3.7 Multiple Interferers

We now evaluate the system by exposing the system to different interferers such as Bluetooth and WiFi traffic from different applications commonly used by other users of the AP. We experimented with three groups of interferers. The first was Bluetooth audio streaming. In this case, we used bluetooth to connect a keyboard, magic mouse, and and JBL Bluetooth speaker indicated in Figure 3.13 as B (Bluetooth Devices) to MacBook Pro as L (Laptop). We used CTC-data rate of 16 bps and changed the UDP packet rate to evaluate BER as a function of rate at which our system generated the UDP packets. In the second experiment, we used Bluetooth and YouTube streaming simultaneously on the laptop. In the third case, we used Bluetooth, YouTube streaming, and a 3GB file downloading on the laptop to understand the robustness of the system under strong interference. In all these experiments, we used the wireless AP as the WiFi transmission device. We used three TelosB motes as the receiver in the sensor network. Figure 3.18 shows the result of our experiment. It is worth noting that during file downloading, the WiFi nominal bit rate was always automatically adjusted by the AP. During the experiment, when we changed the UDP-packet rate, we noticed that the Bluetooth speaker experienced serious time lags, which disappeared after some time. Under all circumstances, the communication achieved 10-15% BER with the highest UDP-packet rate of 1800/s. Since this was a very challenging environment, in which even a WiFi-WiFi or 15.4-15.4 communication would experience losses, it is not surprising that, with smaller UDP-packet rate the CTC BER goes up to 35%. Thus, we find that with appropriate modulation rate, even under heavy interference, the crosstalk-based communication system can achieve less than 10% BER.

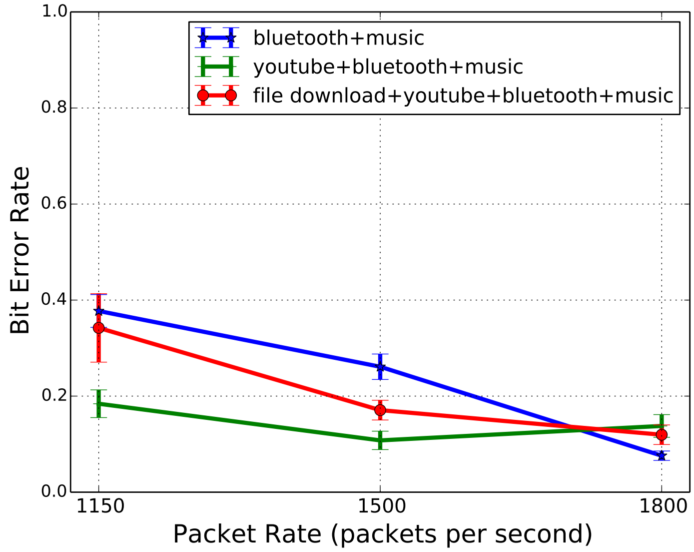


Figure 3.18: BER vs UDP data rate with different WiFi traffic scenarios.

3.3.8 Communication Range

Next, we evaluated the performance of the crosstalk-based communication (CTC) at different distances. For CTC to be useful in practice, it must work at moderate distances. For example, a tablet may need to send a command to a smart device in the home. When the user carries the tablet to a different location in the home, we still need to be able to send the commands to the smart device. We setup an experiment to evaluate the performance of CTC at different distances in a residential apartment as shown in Figure 3.13. We setup five TelosB motes as receivers at 7 ft increment in distance from the AP being used as the CTC transmitter. The AP is a commercial Buffalo router running OpenWRT. The transmission power for the WiFi access point was configured as 17dBm. Figure 3.19 shows the result. In

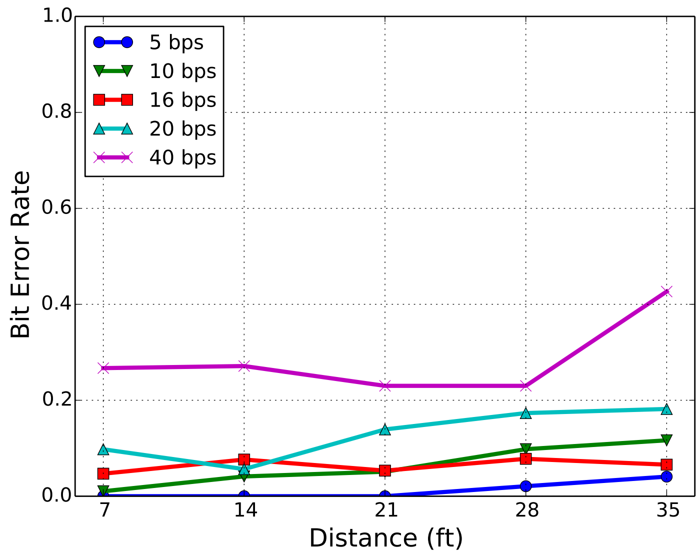


Figure 3.19: BER vs. distance under different CTC-data rates.

this residential apartment, there were multiple WiFi devices including cell phones, laptops, wireless printer, wireless access points, and other reachable access points nearby. The motes sampled RSSI at 4 KHz. We ran the experiment with UDP packets generated at 1000/s. We plotted the BER vs distance for different CTC data rate in Figure 3.19.

We found that the CTC-data rate of up to 16 bps achieved a BER less than 10%. Further, the BER was stable within the 35 ft x 35 ft physical range, which was sufficient for typical CTC-usage scenario within an apartment. This CTC-data rate of 16 bps was sufficient to send commands to smart devices at homes. With lower CTC-data rates, the BER could be close to 0 at moderate distances.

3.4 Discussion

In this section, we discuss different aspects of CTC design and performance issues.

1. *Unidirectionality*. Current implementation of CTC is unidirectional. Only the WiFi devices can send information to the 15.4 devices. Implementation of CTC in this direction is easier than CTC from 15.4 devices to WiFi. We can easily get WiFi transmission to saturate the 15.4 receiver and hence distinguish the times of transmission from times with no transmission. The main challenge in getting CTC to work from 15.4 to WiFi is getting the WiFi radio to detect 15.4 transmissions, which do not have a lot of power, from other transmissions in the crowded 2.4 GHz range. The implementation may be feasible in a commercial WiFi NICs but may require changes to the firmware for low-level access to the device for spectral scans.

2. *Energy Consumption*. The implementation required the 15.4 devices to turn on their radios to listen to the ambient wireless signals, thus greatly weakening the design goal for low-power wireless sensor networks. However, we can reduce the power consumption by coordinating the radio on and off times with the WiFi devices. Many smart gadgets in smart homes, however, may be always powered on. If the 802.15.4 devices are always powered on, leaving the 15.4 radio on all the time may be acceptable.

3. *Data Rate*. With the proposed techniques, we have achieved a data rate of 16 bps. Theoretically, we can achieve a data rate of 3 kbps with the maximum packet rate transmission on the WiFi devices, however, that will require RSSI sampling and decoding at much faster rate on the motes. Operating at such high-rates may also cause the BER to increase. Besides the challenge in high-speed RSSI sampling, symbol alignment also becomes challenging in a WiFi network with other traffic. Furthermore, the traffic generation must be real-time to ensure that the symbol duration is accurate. Otherwise, the decoding signal will not be synchronized with the encoded signal. Fortunately, even a low-data rate CTC is useful for device configuration and commands and we expect CTC to be useful for those applications.

3.5 Conclusion

We designed and implemented a WiFi to 15.4 communication system that utilizes crosstalk or interference between the channels to deliver useful information between the devices. The proposed technique allows WiFi devices to directly communicate with 802.15.4 devices without a physical gateway. We provided a detailed description of the modulation and demodulation schemes and their evaluations in controlled and uncontrolled environments. The results show our proposed system can provide a reliable wireless communication to interconnect WiFi devices with IEEE 802.15.4 sensor nodes with an achieved data rate of 2 bytes per second with less than 10% bit error rate.

Chapter 4

Bypassing RF Interference by using Visible Light Communication

Visible Light Communication (VLC) is emerging as a complementary technology to traditional Radio Frequency (RF) technologies. It provides a noise-free RF environment for transceivers as light and RF are independent frequencies of electromagnetic spectrum as shown in Figure 4.1 [7]. This benefits light-based communication since all existing RF-related interference disappears by default.

VLC is believed to be a candidate for the next-generation cellular networks [39, 91], accurate indoor localization [59, 63, 103, 111] and the Internet of Things. Recent attempts for "softwarization" of VLC networks [23, 98] show the need to speed up the research progress in this new field. Among these former works, only the low-cost OpenVLC project is open-source and it is designed to provide real-time

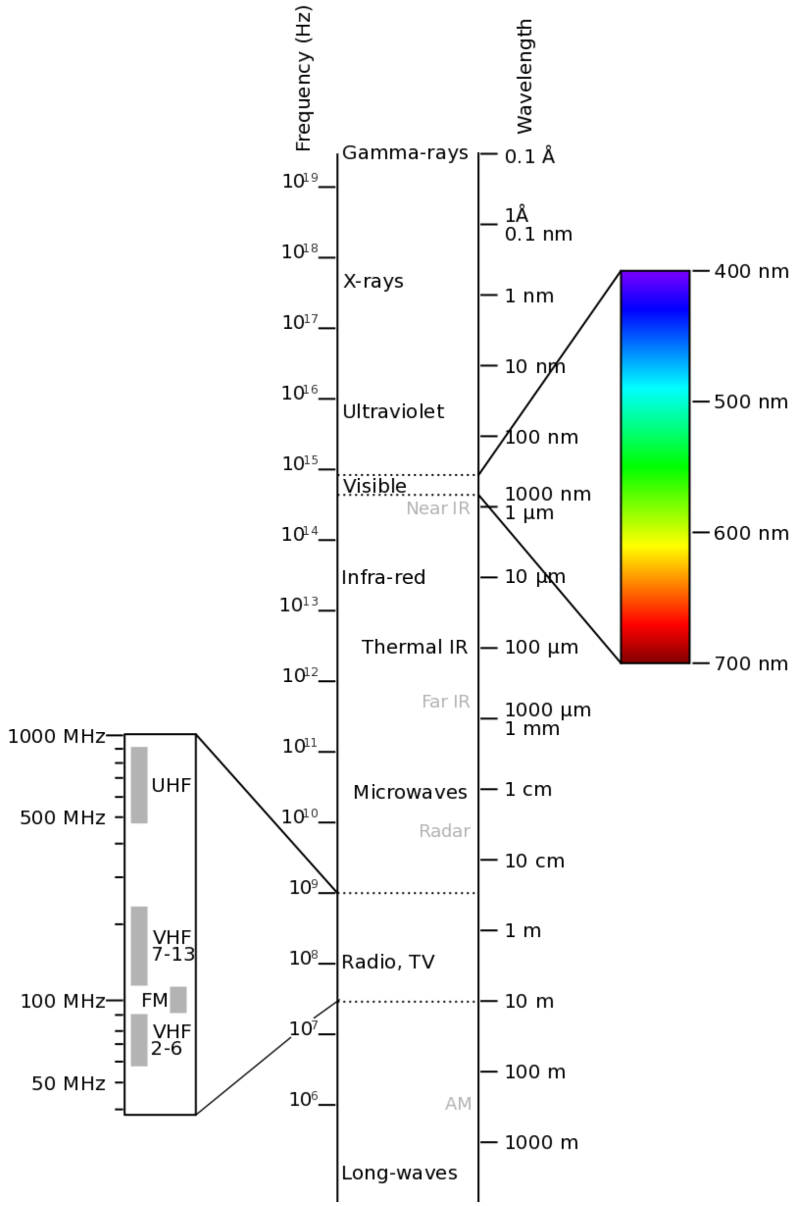


Figure 4.1: Electromagnetic Spectrum from [7].

functionalities for rapid prototyping of networked VLC systems. As part of this dissertation, we introduce the OpenVLC1.0 platform [99], that interfaces an optical front-end consisting of a high-power LED, a low-power LED and a Photodiode (PD) to a cost-effective and powerful embedded board. In order to ease the exploration of the optical front-end, we have designed and built a printed circuit board (OpenVLC1.0 cape) that can be easily attached to the main embedded board. This plug-and-play approach allows researchers to focus on the software design of communication network protocols, without the hassle of wiring the optical components and the electronics on a breadboard. The cape is controlled using the OpenVLC1.0 driver, that implements key primitives at MAC and PHY layer such as signal sampling, symbol detection, coding/decoding, channel contention, carrier sensing, and Internet protocol interoperability. Among its many benefits, OpenVLC1.0 provides the basic tools to implement various protocols and prototype them in real world VLC network setups.

4.1 Design of OpenVLC1.0

OpenVLC1.0 (in Figure 4.2) is a software-defined platform built on a BeagleBone Black (BBB) board [5] and a front-end transceiver that adopts a high-power LED, a low-power LED to transmit and a photodiode (PD) to receive light signals. Several communication links are possible: OpenVLC1.0 can choose between high- and low-power LEDs as the optical transmitter, and between low-power LED and PD as the optical receiver. Each configuration comes with its own unique features in terms of

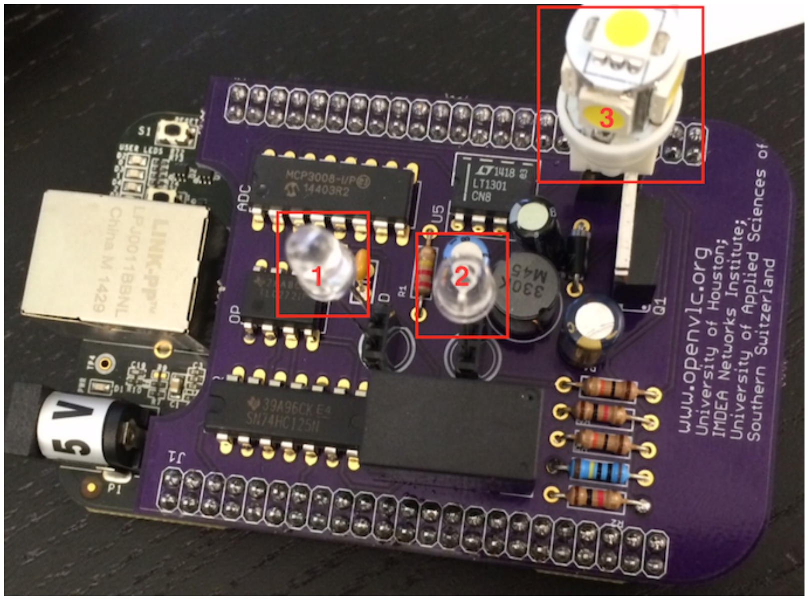


Figure 4.2: The OpenVLC1.0 Cape. The optical components are: (1) low-power LED; (2) Photodiode (PD); (3) High-power LED.

channel propagation, receiver sensitivity, Field-of-View (FoV), etc. Flexible protocols can be designed that can dynamically choose the most desired configuration based on the current circumstance. For instance, the high-power LED can be used to emulate the scenarios of communication under typical indoor illumination from the ceiling, while low-power LED can be enabled for those applications where the primary goal is communication, and the illumination is used as visual feedback. The LED as a receiver can be used to increase the resilience to ambient noise (e.g., sunlight and indoor illumination with no need for additional optical filters. However, it comes with a smaller FoV (Field of View) than PD (Photodiode). In our design, a software-defined optical selector allows the choice of either the low-power or the high-power LED as the transmitter. Similarly, we can choose the low-power LED or the PD as

Table 4.1: Benefit of different communication links provided by OpenVLC1.0

Link	Transmitter	Receiver	Benefit
1	high-power LED	PD	wider field of view
2	low-power LED	PD	longer range
3	low-power LED	low-power LED	time-division duplex
4	high-power LED	low-power LED	link flexibility

the receiver as shown in Table 4.1.

- **Low-/high-power LED-to-PD communication:** OpenVLC1.0 supports the communications between a low-/high-power LED and a PD. Low-power LED-to-PD can be used in scenarios where a more directional communication (e.g., secure communication) is perceived of interest, while a high-power LED can be used as an access point that serves a number of users.

- **Low-power LED-to-LED communication:** while photodiodes are normally used as receivers, a reverse-biased LED (rather than a photodiode) may be used as a receiver to implement bidirectional LED-to-LED communication [66]. This principle has been exploited to introduce the concept of LED-to-LED communication networks and to design an open-source platform for VLC research. In our design, a software-defined transmitter/receiver switch allows us to control the operation mode of the low-power LED. This approach enables the implementation of time-division duplex protocols in the visible light spectrum.

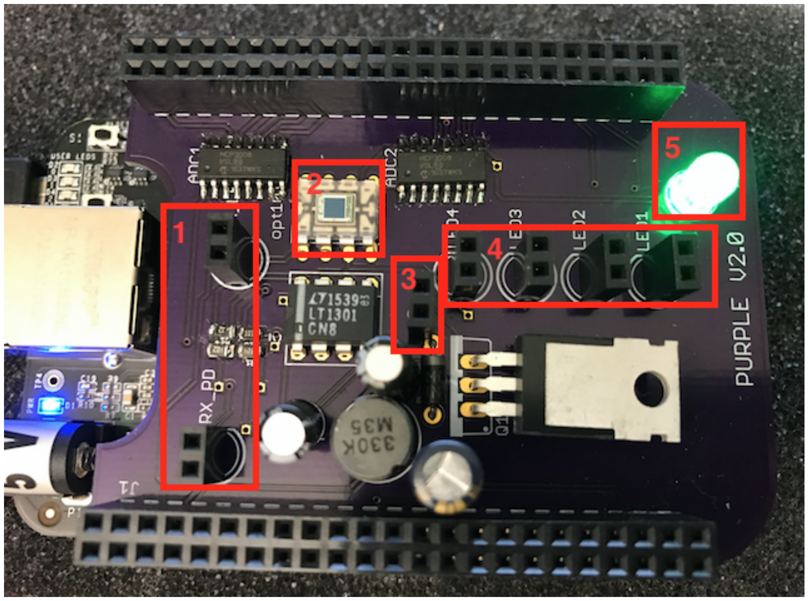


Figure 4.3: The PurpleVLC Board. The optical components are: (2) Dual link LED/PD as RX; (2) Photodiode (PD); (3) High-power LED. (4) LED Array, (5) RGB LED

- **High-power LED to low-power LED communication:** the communications between high- and low-power LEDs are also supported by OpenVLC1.0. Under this case, a pair of high-power LED and a PD can form a transceiver that acts as an access point with wide FoV; while a single LED can be a transceiver residing into embedded size-limited devices.

A study on VLC robustness has been conducted based on this platform in (Chapter 6).

4.2 Design of PurpleVLC

The PurpleVLC board introduces significant improvements to OpenVLC1.0. It also works with BeagleBone Black embedded computer. However, PurpleVLC has taken one step further over state-of-the-art embedded VLC prototype design. It can provide higher throughput with better reliability as well as longer communication range. We can connect PurpleVLC board to BeagleBone Black to serve as the full transceiver system. We implemented the firmware running on PRU to send and receive the packets. The improved performance are due to the following design features: (1) adaptive transmission; (2) parallel communication; (3) I/O offloading; and (4) Full Duplex. Detailed description on this new design is presented in (Chapter 7).

Chapter 5

Making Embedded Visible Light Communication Robust

Visible Light Communication (VLC) has been proposed not only as an alternative wireless channel for communication for IoT but also as one of the ways to address spectrum crunch. Many research projects have tried to advance different flavors of VLC systems in the last few years. Most discussed VLC systems are perhaps the ones that use light bulbs at homes (e.g., LiFi [39]), however there is a body of work on low-cost low-power embedded VLC systems based on LEDs [96]. These systems commonly use a photodiode to receive the transmitted light signals. The photodiode becomes saturated or performs poorly in bright light or in fluctuating ambient light. Results from existing studies [44] show that with strong ambient light, the system will fail to deliver packets. In an indoor scenario, the VLC system needs to be robust against bright light but also to changes in illumination throughout the day and night.

Addressing the poor performance of embedded VLC in bright and changing ambient-light conditions is necessary for embedded VLC systems to mature into systems that can provide robust and reliable communication. Most importantly, for VLC technology's potential to address the spectrum crunch, it is essential that these systems achieve a level of robustness far beyond what the state-of-the-art achieves.

Getting a low-cost embedded VLC system to work robustly in bright light and changing ambient-light levels is extremely challenging. In presence of bright ambient light, for example in a stadium or near a window during the day time, the photodiode used as the receiver will easily become saturated causing the reception to fail: the receiver will not be able to distinguish the ambient and transmitted light because it is already saturated. Similarly, different levels of sensitivity of the photodiode may be required depending on the level of ambient light. Existing prototypes of embedded VLC do not work well in these challenging environments.

Previous work on low-cost VLC systems use photodetectors that perform poorly when subjected to bright or fluctuating ambient light. In a typical LED-to-photodiode communication system, the existing approach is to switch to a different type of receiver or transmitter when the communication degrades. These workarounds, however, do not directly address the main problem by actively canceling the ambient light based on its intensity. In this work, we aim to fill that gap in the state-of-the-art of embedded VLC systems.

Our approach consists of two main parts. First, we design a DC-restoration circuit that can eliminate the effect of ambient light adaptively depending on the level of ambient light. Second, we use the frequency-shift keying modulation with a small number of frequencies rather than the on and off keying modulation to provide better SNR (Signal Noise Ratio) with the proposed circuit. Compared to the state-of-the-art embedded VLC system, our system demonstrates the reliability by offering an extremely low symbol error rate (nearly 0) and an acceptable data rate of up to 3 kbps with a distance of 50 cm in a controlled indoor environment.

We make the following contributions in this work:

- We present the circuit and accompanying physical layer design, which together form the ambient-light cancellation primitive to enable communication with high reliability and robustness in bright and changing ambient-light conditions.
- We prototype the system and perform extensive evaluations to understand the effectiveness of the system in challenging scenarios.

5.1 Related Work

We briefly review work related to visible light communication.

Low Cost Visible Light Communication Prototypes:. Klaver and Zuniga in [58] introduced a low-cost VLC prototype named Shine. The prototype uses a LED transmitter and photodiode receiver. They observed that the SNR varies with the ambient-light levels and the distance between the transmitter and receiver. While in this work, we argue that the SNR can be maximized by nullifying the background noise. Schmid *et al.* [80] presented a LED-to-LED communication system which demonstrated a data rate of 800 bps with an operating distance of 2 m. The Linux Light Bulb idea was proposed by Schmid *et al.* [79]; they embedded a wireless System-on-a-Chip (SoC) running OpenWRT into a normal light bulb to connect to the Internet. The performance of this bulb, especially the reliability of communication, is not yet reported. More recently, OpenVLC1.0 was proposed as a low-cost embedded VLC platform [99]. It has a full IP stack and can use a LED or photodiode as the receiver. One key problem with this prototype is its limited reliability and robustness. Heydariaan *et al.* [44] has investigated its performance under various experimental settings. Unfortunately the platform cannot assure reliable communication with bright ambient light. Zhang *et al.* [113] has proposed a new circuit to cancel the minimum offset voltage from the input optical signals. They demonstrate the proposed prototype can be immune towards sunlight and indoor fluorescent lights. However, they did not evaluate the prototype in a dynamic environment. We differ in both the design and evaluation in a dynamic and challenging environment.

Ambient Light Effects:. Li *et al.* [67] presented a system that can reconstruct human movement using off-the-shelf LEDs and photodiodes. They claimed a higher level of ambient-light intensity will cause the saturation problem when the photodiode is operating outside the linearity area. Meanwhile, Yang *et al.* [103] also claimed

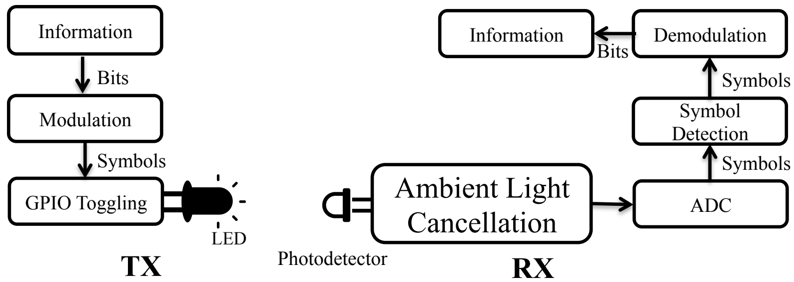


Figure 5.1: System Overview: The ambient-light cancellation block will filter the ambient light, leading to a ZERO output corresponding to the reception of ambient light, thereby enhancing the reception of VLC packets.

this effect from the ambient light. Li *et al.* [90] later proposed a method to fade the effect from ambient noise by recognizing the rising edge of the encoded light pulse from the fluctuated ambient light. However, this does not cancel the entire interference from the ambient light leading to increased effort to improve edge detection accuracy.

5.2 Design

5.2.1 Modulation

Our goal is to design a modulation scheme, that is robust to light interference, and with simple logic it will be possible to implement, mostly in software, in a low-cost embedded platform (in contrast to OFDM in high-resource systems such as LiFi [39]). Current low-cost LED-to-LED or LED-to-Photodiode VLC systems use variations of an On-Off Keying (OOK) modulation scheme due to its simplicity [99]. However,

Table 5.1: OOK with Manchester Encoding used in OpenVLC1.0 and BFSK Encoding used in our design

Bit	**OOK(OpenVLC1.0)**	**BFSK(Our Encoding)**
0	01	1100
1	10	10

OOK is susceptible to ambient-light interference [44]. Although there are other modulation schemes in the broader VLC space (e.g., LED-to-Camera [48], Binary Frequency Shift Keying or BFSK [76]), they do not directly address the ambient-light interference in embedded VLC.

We use BFSK as the modulation scheme rather than OOK in our design since it is less susceptible to ambient noise. Table 5.1 shows the modulation schemes used in OpenVLC1.0 platform and our current design. Different from OpenVLC1.0, we represent bit ZERO with 4 symbols which are ’1100’(HIGH, HIGH, LOW, LOW), we represent bit ONE with 2 symbols which are ’10’(HIGH, LOW). The transmitter represents bit ’1’ with a frequency of 2 KHz to blink the LED. It represents bit ’0’ with a frequency of 1 KHz to blink the LED. Assuming we are continuously sending one byte. We represent the modulated signal in Figure 5.2 with the encoding scheme. We use more symbols to represent one bit compared to OpenVLC1.0. Our approach improves robustness, but reduces the communication rate.

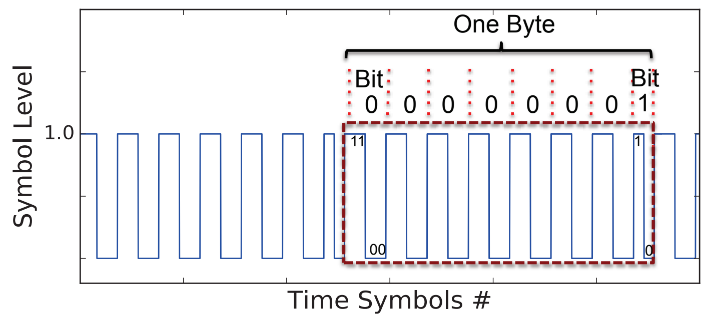


Figure 5.2: Encoding method to represent bit ONE and bit ZERO.

5.2.1.1 Ambient Light Cancellation

While DC restoration circuit [10] has been used to remove the effect of constant ambient light, those designs do not directly address the dynamics in ambient light in typical indoor deployments. A static DC restoration circuit would not work in bright and fluctuating light, which can all occur in a single deployment. The compensated current generated from the circuit needs to be adaptively adjusted in the presence of these ambient light changes with a feedback and control mechanism. This mechanism has three main parts as show in Figure 5.3.

Photodetector Amplifier: This amplifier amplifies the current from the photodiode upon light reception and outputs V_o.

DC restoration: The DC restoration will generate the compensating current to the summing point between the photodetector amplifier and the DC restoration. We add the adaptive components (Figure 5.3) to the basic DC restoration circuit [10].

Adaptive cancellation: We use the digital potentiometer P1 to control the

Algorithm 2 Adaptive Cancellation Algorithm

Input: WS (WindowSize) in

```
numZero = 0; numOne = 0
count = 0; min = 0; max = 255
mid = read from potentiometer
minSps = 20% * WS; maxSps = 75% * WS
while TRUE do
    value = read one symbol from ADC
    count++
    if value == 0 then
        numZero++
    else if value != 0 then
        numOne++
    end if
    if count == WS then
        count = 0
        gap = numZero-numOne
        if (gap>minSps) and (numZero>maxSps) then
            min = mid ; mid = (max+min) / 2
            write (mid) to potentiometer
        else if (-gap>minSps) and (numOne>maxSps) then
            max = mid ; mid = (max+min) / 2
            write (mid) to potentiometer
        else if (|gap <= minSps|) then
            mid = 0; max = 255
            mid = read from potentiometer
        end if
        numZero = 0; numOne = 0
    end if
end while
```

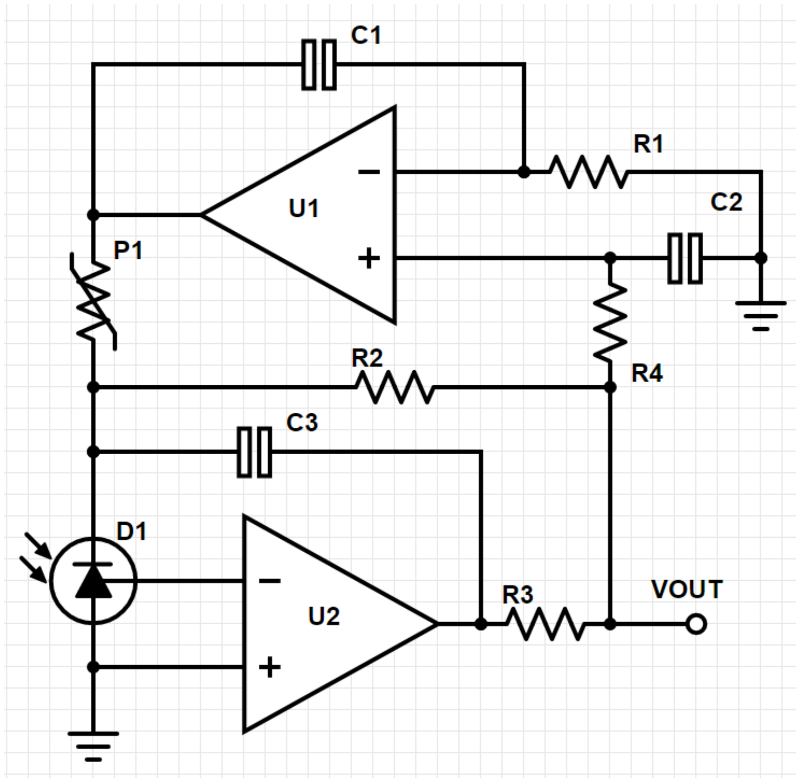


Figure 5.3: Ambient Light Cancellation Circuit Diagram consisting of a block for photodetector amplifier, a block for DC restoration, and one for adaptive cancellation, which uses feedback from the output voltage V_o using a potentiometer (P1) and a LED indicator (D1). The input and output interface to the embedded VLC board.

amount of current to provide to compensate for the current generated by the ambient light. If V_o is in saturation range, the resistance of P1 is decreased. After each adjustment, the onboard LED is turned ON and OFF using the symbol rate to calibrate the resulting ADC range. If the V_o range for ON and OFF is below a certain threshold necessary for robust disambiguation between an ON and an OFF, the resistance is decremented, otherwise the resistance is incremented until a suitable resistance value is found. Algorithm 1 uses the binary search to find the resistance

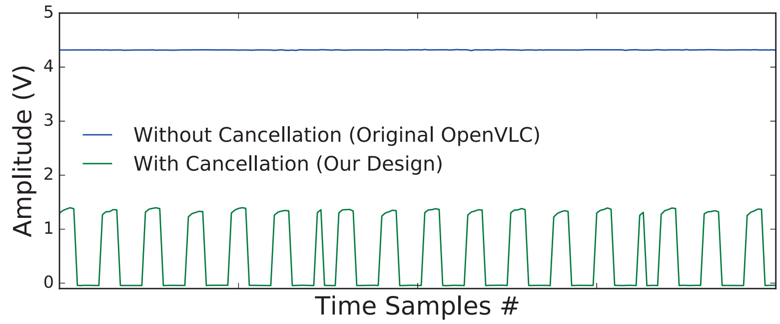


Figure 5.4: Comparison of received light signal both with and without ambient-light cancellation.

value for P1 that allows output voltage to stay below saturation and maintain a sufficient voltage range between an ON and OFF.

Figure 5.4 shows the output from the receiver with and without the cancellation mechanism. An oscilloscope is used to plot the received signal from the modified VLC node. We use a lamp that was equipped with the GE 100-Watt A21 3-way reveal light bulb as the ambient-light interferer. The transmitter, which is placed 50 cm from the receiver, transmits a sequence of ones and zeros with a symbol rate of 1 KHz. With the bulb turned on, the light signal was fully recovered with the cancellation process, while the photodetector on the unmodified OpenVLC1.0 platform gets saturated and outputs a flat voltage.

5.2.1.2 Symbol detection

We separate this task into two steps:

Synchronization: The receiver and transmitter clocks need to be synchronized for correct decoding of the symbols. In low-cost platforms such as BeagleBone, that uses CPU clocks to maintain timers, the clock drifts, hence it requires a transmitter-receiver resynchronization after a modest number of symbols are transmitted. We borrow this technique that uses Xenomai [19], a real time software framework from OpenVLC1.0 to keep the receiver and transmitter synchronized. Through experimentation, we found that the synchronization is consistently accurate for several thousand symbols, hence we use a symbol buffer of 1000 bytes.

Packet Framing The ADC output from the cancellation circuit will be LOW when no packets are transmitted. Hence we use a sequence of 50 LOW symbols to occupy the space between the packets. Reception of either ONE or ZERO starts with a HIGH symbol, which indicates the beginning of a packet after a sequence of LOWs.

5.3 Implementation

We prototyped the transmitter using OpenVLC1.0 platform [19] with a Linux kernel module implementation for our proposed BFSK modulation scheme. We prototyped the receiver using OpenVLC1.0 platform to include the adaptive ambient-light cancellation circuit between the photodetector and the ADC without modifying the platform. Instead of doing symbol detection in kernel level, we dumped the symbols received from the kernel space to user space for symbol detection and symbol error rate calculation. This arrangement allows the kernel to continuously receive

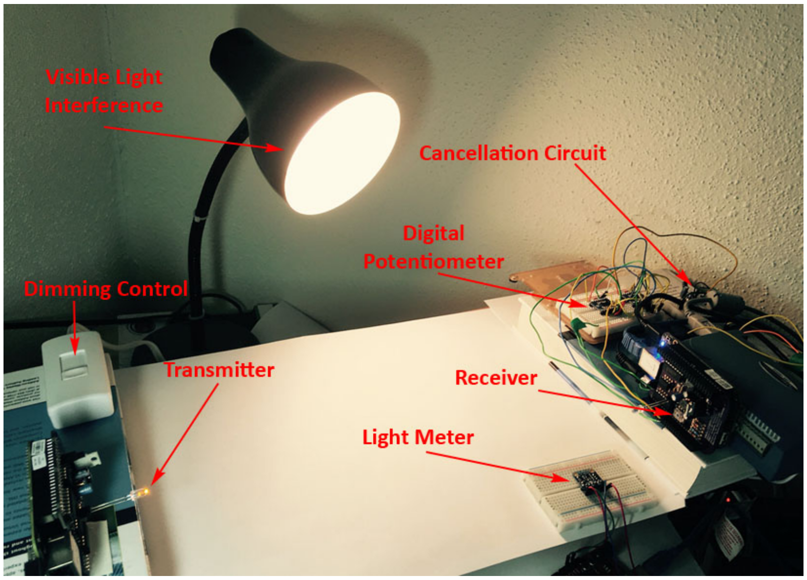


Figure 5.5: The testbed setup for experiments to evaluate the adaptive cancellation mechanism.

the signals without incurring in-line signal processing delays. Figure 5.5 shows the testbed setup. We use a 5V, 2A power adapter to power each node. Each node costs around $45 excluding the BeagleBone Board. The current draw for the receiver with cancellation is about 315 mA while the BeagleBone is running.

5.4 Evaluation

5.4.0.1 Adaptation Latency

Now, we evaluate how fast the system can adapt to the changing ambient light level. As we need a potentiometer to provide feedback for the cancellation circuit as shown in Figure 5.3, we select AD5242 from Analog Device with the range of 1 MΩ

Table 5.2: The system took the adaptation time to converge to the shown potentiometer value to fully cancel the different levels of ambient light

Ambient Light Intensity (lux)	**(120)**	**(160)**	**(200)**	**(400)**
Potentiometer Value (Ω)	2 M	250.9 K	188.2 K	158.7 K
Adaptation Time (ms)	0	34	85	102

in the prototyping. AD5242 can generate 256 different resistance values in the 0-1 MΩ. Binary search (Algorithm 1) allows us to determine the right resistance in a maximum of 7 steps. Assume the sampling rate is S, window size for the resistance adaptation is W, binary search steps is N. Then the time, T, required for one adaptation is $T = N * W/S$. With sampling at 6 KHz, 100 symbol window and maximum of 7 steps, the system will take a maximum of 117 ms to adapt to a new ambient-light level. We consider this latency as acceptable because we do not expect ambient-light dynamics to be of a higher frequency. Data communication resumes after the optimal resistance is found.

Experiment: We use the sliding dimmer on a GE 100-Watt A21 light bulb to generate different ambient-light intensities. We place a photodiode 50 cm from the light bulb. We then connect two AD5242 (digital potentiometer) between the photodiode and the cancellation circuit. We then utilize a TSL2561 light meter to measure the ambient light to help perform the experiments at different lux values.

Results: Figure 5.6 shows that the resistance from the digital potentiometer will decrease once the ambient light goes up and vice versa. It also demonstrates that resistance change is very fast (up to 117 ms) towards the change of the ambient light.

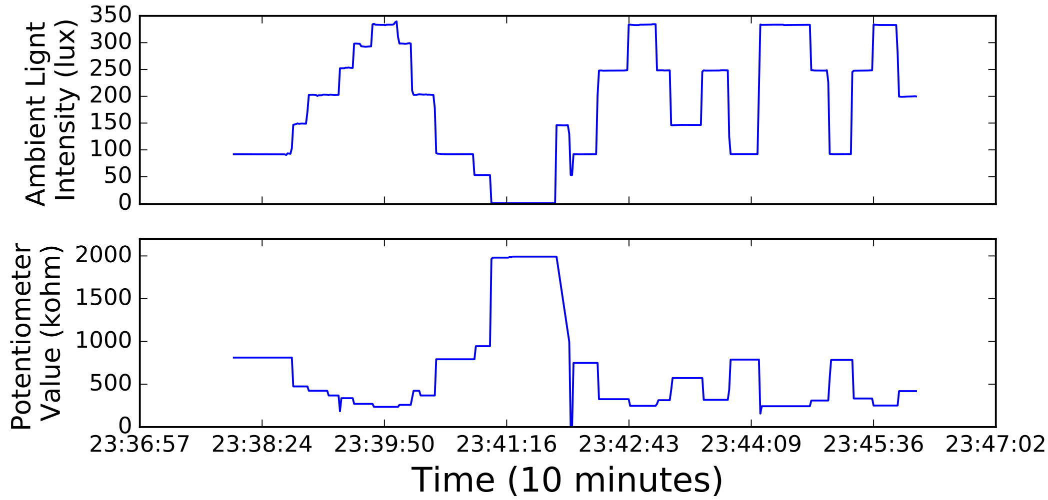


Figure 5.6: The top figure shows the fluctuation of the ambient light during the experiment. The bottom figure shows the value of the digital potentiometer due to the adaptation algorithm.

Note that each resistance value showing on the plot was the optimal resistance that provided the noise-free environment for highly accurate symbol detection. Table 5.2 shows four levels of ambient light and the corresponding four suitable resistances determined by the algorithm. As seen on Table 5.2, adaptation time increased with an increase of ambient-light intensity. A smaller resistance value was used to cancel higher ambient-light intensities, leading to more steps from the initial resistance value (2 MΩ) to the optimal value, and this required longer adaptation times.

In another experiment, we let the system run for 15 h in an office space, allowing the system to experience different levels of ambient light and hence different resistance values selected by the cancellation circuit in response as shown in Figure 5.7.

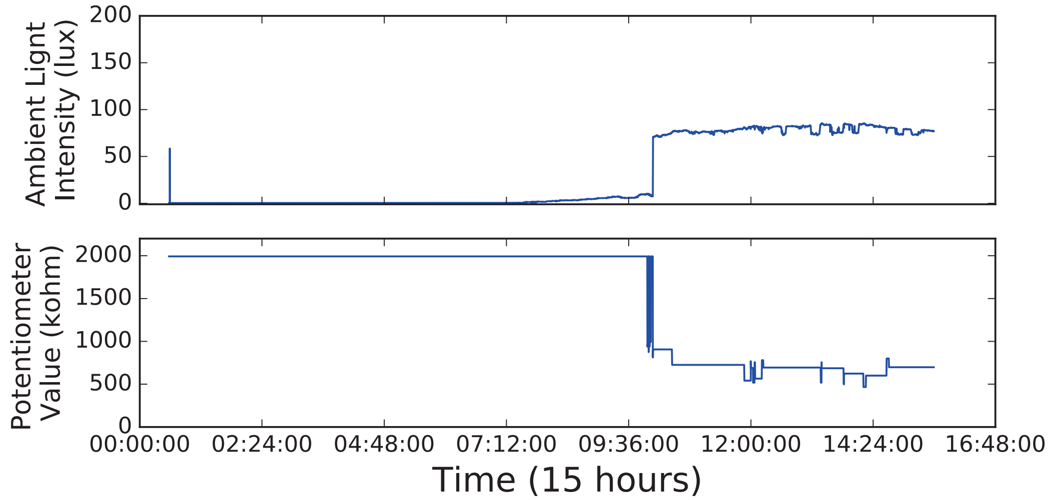


Figure 5.7: Ambient-light intensity and the resistance value selected by the cancellation circuit over 15 h in an indoor office environment. The ambient light was dynamic as people performed different activities in the office space.

Table 5.3: Experiment settings to study the impact of various ambient-light intensity on symbol error rate

Ambient Light Intensity	**Symbol Rate**	**Distance**	**Buffer Length**	**No. of Symbols**
120, 160, 200, 400 lux	1 KHz	50 cm	1000 B	30000

5.4.1 Impact of Various Levels of Ambient Light on Symbol Error Rate

Experiment: We configured the TX and RX to operate with the symbol rate 1 KHz. We placed the receiver and the transmitter 50 cm apart. We performed the experiments at three different light levels as shown in Table 5.3). In each experiment, the transmitter sent 1000 bytes. We also configured the unmodified OpenVLC1.0

Table 5.4: Symbol error rate under different ambient-light intensities

Ambient Light Intensity	**SER with Cancellation**	**OpenVLC1.0 No Cancellation**
(160lux)	0.00%	Fail
(200lux)	0.00%	Fail
(400lux)	34.67%	Fail

platform to be another pair of TX and RX for comparison purposes. We listed the configuration on Table 5.3.

Result: Table 5.4 shows the symbol error rate (SER) achieved at different light levels. The bulb was turned on, causing the heavy ambient light with 160 lux. We used dimming control to make the bulb brighter generating an intensity with 200 lux. We can also set the bulb to be brightest leading to an ambient-light intensity with over 400 lux. The result demonstrates a highly adaptive robustness and reliability which can be achieved even in the presence of strong ambient light. The proposed system fails at 400 lux while the OpenVLC platform fails at the light levels on Table 5.4. Thus, the proposed system is a significant improvement over the state-of-the-art.

5.4.2 Impact of Symbol Rate on Symbol Error Rate

We use experiment settings listed on Table 5.5 to study the impact of symbol rate on system performance. All experiments were conducted in a very bright ambient-light

Table 5.5: Experiment settings to observe the impact of various symbol rate on symbol error rate

Ambient Light Intensity	**Symbol Rate (KHz)**	**Distance**	**Buffer Length**	**No. of Symbols**
200 lux	1,2,4,6,8,10	50 cm	1000 B	30000

Table 5.6: Experiment settings to study the impact of distance on symbol error rate

Ambient Light Intensity	**Symbol Rate**	**Distance** (cm)	**Buffer Length**	**No. of Symbols**
160 lux	6 KHz	20, 30, 50, 60	1000 B	30000

background by turning on the light bulb to brighter levels. We observed that the OpenVLC1.0 platform without cancellation does not work in this lighting condition.

Result: Figure 5.8 plots the relation between the symbol error rate and symbol rate. Here the symbol rate is equal to $1/symbol$ duration. It shows that the symbol error rate increases significantly once the symbol rate increased from 6 KHz to 8 KHz to 10 KHz.

This is because higher symbol rates can lead to lower SNR, causing higher symbol error rates.

5.4.3 Impact of Distance on Symbol Error Rate

Next, we study the performance of the system over distance using the settings listed on Table 5.6.

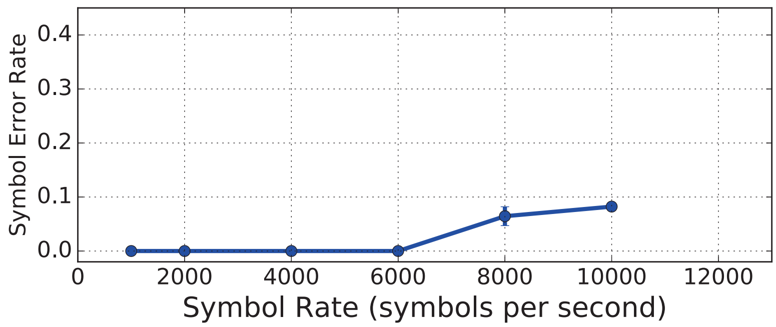


Figure 5.8: Symbol error rate vs. symbol rate.

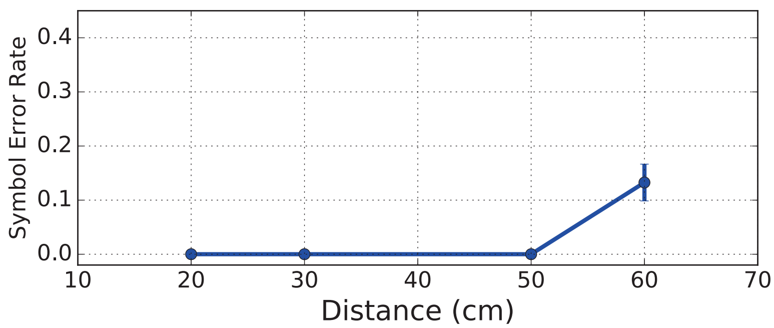


Figure 5.9: Symbol error rate vs. distance between TX and RX.

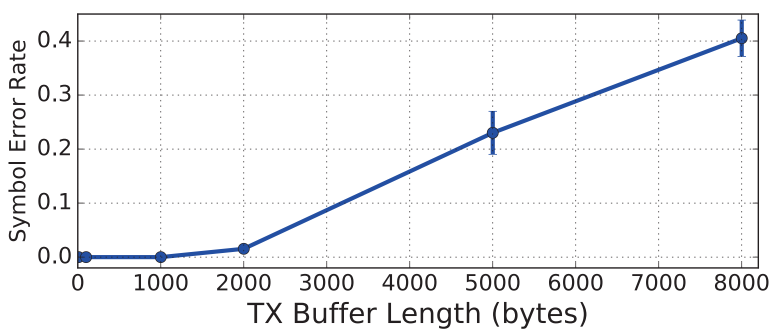


Figure 5.10: Symbol error rate vs. TX buffer length.

Result: Figure 5.9 plots the symbol error rate with different distance settings when the light bulb was turned on. We are concerned about the operating distance

Table 5.7: Experiment settings to study the impact of buffer length on symbol error rate

Ambient Light Intensity	**Symbol Rate**	**Distance**	**Buffer Size (x10^3B)**	**No. of Symbols (x10^4)**
160 lux	6 KHz	50 cm	1, 2, 5, 8	6, 12, 18, 24

for the system. It shows that as the distance goes up to 60 cm, the symbol error rate increases to 17%. We consider this error rate to be unacceptable given the reliability requirement. We then limit the experiment to be with a distance of 50 cm. It also provides insights on how to make the system robust at longer distances by adding another amplifier after V_o as shown in Figure 5.3.

5.4.4 Impact of TX Buffer Size on Symbol Error Rate

The system synchronizes the transmitter and the receiver after transmitting a buffer worth of symbols. The optical clock can have a clock drift if the TX buffer is too large. We use the settings on Table 5.7 to study this issue.

Result: Figure 5.10 shows the plots of the symbol error rate with different TX buffer size. It is interesting to notice that with a 1000 B TX buffer, no symbol error found. While with a TX buffer of 8000 B, the symbol error rate went up to 41%. This suggests smaller TX buffer size is preferred if we want to build a reliable low-cost visible communication system.

5.5 Conclusion

We made the following contributions in this work:

- We presented the circuit and accompanying physical layer design, together form an ambient-light cancellation primitive to enable communication with high-reliability and robustness in bright and changing ambient-light conditions.
- We prototyped the system and performed extensive evaluations to understand the effectiveness of the system in challenging scenarios.

Chapter 6

Making Embedded Visible Light Communication Faster

Wireless Communication has expanded from the radio spectrum to light spectrum. The majority of current high-end visible light communication are limited within 1 m communication range [94], [56]. Meanwhile, high-end VLC systems tend to use avalanche photodiode as the receiver, this is not cost-effective and restricts the system from being practical. Also, the system complexity for the high-end VLC is not affordable for the general public, which significantly limits its usage compared to WiFi devices. On the other hand, researchers have proposed low-end VLC platforms that are built using off-the-shelf components, such as OpenVLC [99], led-to-led communication [80]. However, these platforms provided limited data rates for low-power applications and also lack robustness and reliability.

It is important to have an easy-to-use platform for low-end VLC because embedded computers have the capability to process certain computations for domain-specific applications. Smart phones are able to process more computation-intensive tasks. Similarly, a great shift in the market is an increase of edge computing devices for industries to provide real-time monitoring and actuation. In this work, we argue that these off-the-shelf edge devices are able to provide high-speed data rate with limited-power usage. A major benefit of enabling high-speed low-power VLC to avoid the interference on unlicensed ISM band, especially on the 2.4 GHz band. Numerous low-power Bluetooth devices with 1 Mbps throughput are reported to experience strong interference from WiFi devices. From this perspective, it is interesting and important to develop complimentary technologies to make data communication reliable and robust for a practical noisy environment.

It is challenging to achieve high-speed VLC because of hardware selections and software approaches. For the hardware, it is not possible to use highly sensitive photodetectors due to cost and system complexities. It will not be practical to use USRP devices to aid the design and implementation for the modulation and demodulation process. For software approaches, it will be impractical to apply complex modulation and demodulation techniques on edge devices because of limited resources.

There are two main reasons why previous approaches have limited performance in low-end VLC in terms of throughput and communication range. First, the edge devices used in the system have GPIO toggling limitations. For example, if we use Arduino Uno, the GPIO toggling speed is up to 16 MHz. If we use BeagleBone Board, the GPIO toggling speed can be up to 200 MHz. This is the fastest GPIO toggling device we can find in edge devices to the best of our knowledge. We did not see related research based on this observation. Previous approaches ignored this aspect and focused on the software technique in Linux Kernel Driver. However, Linux Kernel is not deterministic in terms of real-time communication. Writing drivers in Linux Kernel with software-defined SPI communication will limit the speed to the sampling rate of ADC (Analog-to-digital converter), which further limits the throughput for the system. From this perspective, we propose a new design that requires a multi-processor approach to speed up low-end VLC.

In this work, we proposed a new architecture for low-end visible light communication system running on edge devices with a multi-processor architecture to enable a 100 kbps data rate with reliability for room networking technologies.

6.1 Related Work

We briefly review work related to reliable and robust visible light communication.

Low-cost embedded VLC platforms:. Shine [58] is a low-cost VLC platform that can provide a data rate of 1 kbps with a communication range of around 1 m. LED-to-LED communication system [80], [81] was reported with a data rate of 800 bps and an operating distance of 2 m. More recently, OpenVLC1.0 was proposed as a low-cost embedded VLC platform [99], [95]. Heydariaan *et al.* [44] investigated its performance under various experimental settings and found that the system achieves a maximum data rate of 12 kbps. These platforms do not provide a data rate for IoT devices comparable to Bluetooth or low-power wireless devices, making visible light communication far less appealing as a medium for room-area networking. ModBulb [43] took the VLC performance one step further using a FPGA-based control. Their system can generate 1 mbps base band signal, however their work focused on the transmitter side and, hence, it is unclear what a low-cost receiver design would look like for this system. Although FPGA can be low-cost, it does require modest complexity hardware design compared to providing a mostly software-based solution by fully utilizing resources on low-cost and popular IoT platforms such as Raspberry Pi and Beaglebone. In our work, we address both transmitter and receiver design to build a complete end-to-end system. Our transmitter routinely achieved more than 1 mbps but the receiver became the performance bottleneck. Most recently, Philips Lighting Research proposed a two way communication using one single RGBW LED [65,66]. Their prototype can transmit and receive light signals in parallel using only one LED rather then photodetectors and achieve several kbps data rate at a distance of tens of centimeters. Yin *et al.* [104] proposed an adaptive ambient-light cancellation VLC platform robust to dynamic ambient light. The operating distance for the platform is also limited to tens of centimeters. In this work, PurpleVLC can achieve in the order of hundred of kbps data rate at a distance up to 6 m.

Multi-processor architecture. Researchers have utilized I/O offloading in multi-processor architecture. Zhai *et al.* [110] identified the problem that various I/O devices can slow down the system performance and they reduced the overhead significantly by offloading I/O performance to a slower processor while keeping the main processor running normal OS scheduling tasks. Islam *et al.* [88] presented a system with multi-processor architecture that can offload the audio-data sampling from main processor to a weak processor. Inspired by this work, we offloaded VLC related TX/RX IO tasks to the PRU on the Beaglebone platform and achieved the same speedup benefits that were realized in other domains. Thus, our work showed that the offloading architecture is useful for VLC design and should be considered in future platform designs.

Light-based sensing and communication system:. Light-based sensing applications are active research areas in the past few years, especially using light for indoor localization. Yang *et al.* [103] proposed an indoor positioning system using polarized visible light. Their system had a data rate of 50 bps which is sufficient to send location beacons to a camera. LiTell *et al.* [111] introduced a visible light-based localization system that can sense the unique oscillating frequency for fluorescent light by using a portable light-sensing dongle with TIAs (Transimpedance Amplifier) and Picoscope. Varshney *et al.* [92] introduced a new visible light sensing system that utilized solar panels as the light receiver to achieve ultra low-power sensing. They leveraged radio backscattering for the communication part. This added system complexity since it involved both light sensing and backscatter platforms. However, the proposed design in our system focused on simple and dedicated light-communication performance.

6.2 System Architecture

In this section, we describe the system design and implementation. We highlight the benefits that each component provides to the communication system.

6.3 Limitations of Existing VLC Platforms

We first give an overview of state-of-the-art embedded VLC platforms and the trade-offs they make in their design.

Table 6.1 summarizes current embedded VLC platforms and compares them based on their system performance, networking architecture, and system flexibility. Among all these platforms, OpenVLC1.0 demonstrated the highest performance with a data rate of 16 kbps and a communication distance up to 5 m using LED-to-PD antenna pair. To achieve that performance, it used a 1 W LED-array as the transmitter and a relatively expensive photodiode with on-chip amplifier. The PurpleVLC platform has a 6-7x performance gain with similar electronic parts, but the transmitting LEDs consumes much less power. Another important aspect for embedded VLC networking is support for multi-hop communication. Shine is the first platform that supports this feature and demonstrated potential for future embedded VLC networking but it is limited to a half-duplex multi-hop. Our design of PurpleVLC achieves multi-channel and full duplex networking and consequently high data rates. We next discuss the main performance and flexibility bottlenecks

Table 6.1: Performance for state-of-the-art embedded VLC

System	**Dietz** ***et al.*** **[31]**	**Schmid** ***et al.*** **[80]**	**Klaver** ***et al.*** **[58]**	**Wang** ***et al.*** **[96]**	**Hewage** ***et al.*** **[43]**	**Li** ***et al.*** **[66]**	**Our Work** ***et al.*** **[107]**
Data Rate	250 bps	800 bps	1 kbps	16 kbps	1 mbps	1-10 kbps	100 kbps
Distance	∼10 cm	∼2 m	∼1 m	∼ 5 m	NA	∼20 cm	6 m
Multi-hop	No	No	Yes	No	No	No	Yes
Full-Duplex	No	No	No	No	No	No	Yes
Parallel Channels	No	No	No	No	No	No	Yes
Implementation	MCU	MCU	MCU	ARM	FPGA+MCU	MCU	ARM + PRU
Antenna	LED-to-LED	LED-to-LED	LED-to-PD	LED-to-LED/PD	LED-to-PD	LED-to-LED	LED-to-LED/PD

of state-of-the-art VLC platforms.

6.3.0.1 Limited Clock Rate on Microcontroller-based VLC

Microcontroller-based VLC such as Arduino-based VLC system has up to 1kbps data rate [80]. The same group also designed a SoC-based Linux bulb that can achieve less than 1kbps data rate in [79]. The bottleneck for their design is due to the limited GPIO toggling frequency since the Microcontroller clock runs at 8-64 MHz. Supporting higher data rates will require faster GPIO toggling. The other issue for Microcontroller-based VLC is due to the limited GPIO pins and ability to control them concurrently to support concurrent VLC channels. Limited performance for Shine platform [58] was mainly due to the controller selection. As a result, on those platforms with a single MCU, the system cannot both transmit and receive at the same time and be limited to half-duplex communication. A full-duplex feature would not only increase point-to-point data rate but also minimize forwarding latency in a multi-hop setting.

6.3.0.2 Kernel Overhead with I/O scheduling

Platforms such as OpenVLC [99] are built on top of more resourceful single-board computer platforms such as Beaglebone which can run Linux. OpenVLC implements a full VLC PHY/MAC layer as a kernel module. Implementing VLC RX/TX IO as part of kernel module can potentially make the software largely portable to other platforms that run Linux but the flexibility comes at a significant performance penalty because the kernel handled busy GPIO toggling tasks. In addition, because of that approach, to maintain low-CPU utilization, we need to also limit the toggling rate. As a result, their platform caps the ADC-sampling rate used in reception: 75 ksps with 3.0 V power supply and 200 ksps with 5.0 V power supply. Offloading the RX/TX related IO tasks to another processing unit (PRU in case of Beaglebone) would free the processor from the overhead while achieving a much faster IO performance leading to higher VLC-data rates.

6.3.0.3 FPGA-based transmission and USRP-based reception

FPGA has been used to achieve faster IO, for example in modBulb [43]. USRP could also achieve high performance [89]. These approaches can easily generate a toggling rate in the MHz range but introduces system complexity and cost. Our work is motivated by the observation that the low-cost single-board IoT platform has resources (e.g., the additional processors called PRUs in Beaglebone) which can be utilized to achieve similar performance without additional hardware or cost. Beaglebone-based VLC system can generate toggles at more than MHz without adding USRP or FPGA hardware to the platform and the associated complexities.

6.4 Transceiver Design

The design of PurpleVLC, a new embedded VLC platform, can achieve high data rates without using FPGA or USRP while allowing low-power platforms to keep CPU utilization low. The platform integrates RX/TX I/O offloading with the capability to change the number of LEDs for communication performance objectives, multiple concurrent communication channels, and full-duplexing to achieve high data rates. Next we describe the design of PurpleVLC.

Figure 6.1 shows the transceiver architecture. The transceiver architecture consists of a TX (transmitter) and an RX (receiver) module. The transceiver supports LED-to-LED communication or LED-to-photodiode communication. It can use single-color LED or RGB LED as the transmitter. Figure 6.2 shows the implementation of our design. We have populated four sockets (each with a two-pin header) for the transmitter on the printed circuit board (PCB) to use multiple or different types of LEDs and enable different combination depending on their communication objectives. Thus, the platform can support multi-transmitter VLC with up to 4 LEDs. We also populated two sockets for the receiver. The users can insert either photodiode or LED as the reception antenna. The rest of the components and the software are designed so that the board can use any combination of LED or Photodiode from any socket. Concurrent channels can be established with this board since the TIA and the ADC can support 2-channel data acquisition simultaneously.

6.4.0.1 I/O offloading

Fundamentally, a VLC transmitter requires toggling or controlling of the LEDs to encode bits using different coding schemes. Previous approaches used a kernel-based driver to control the IO [99] but could not achieve high-data rates due to the lower

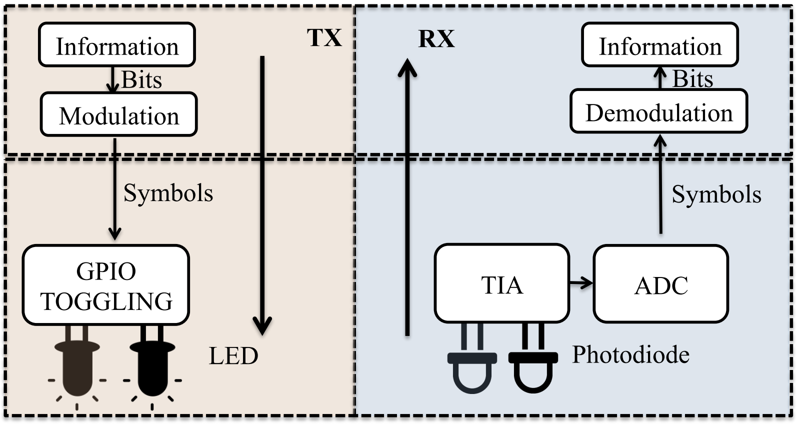


Figure 6.1: Bidirectional transceiver

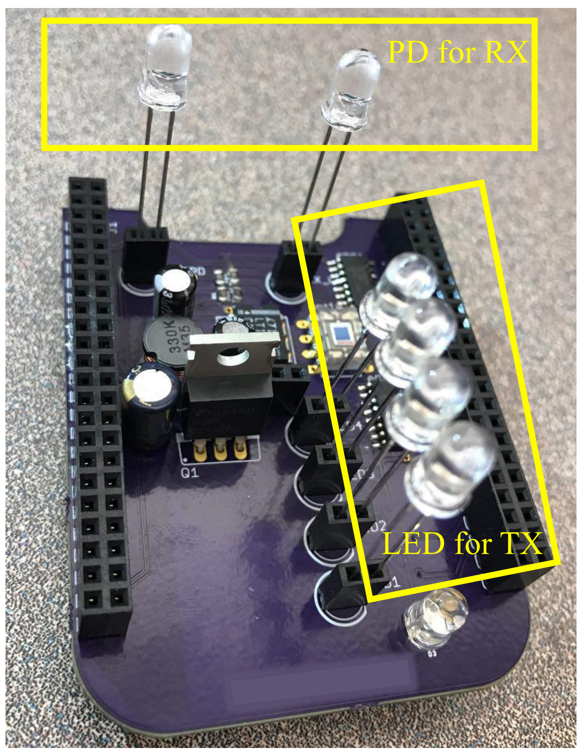


Figure 6.2: The PurpleVLC board showing four TX LEDs plugged in and two elements plugged in for RX chain.

GPIO toggling rates: first high rates are difficult to achieve with that approach and second the rates had to be kept lower than what is possible to keep the CPU-utilization low. In our design of PurpleVLC, we offloaded the control of IO operations from the main processor to the PRU, which is an auxiliary processor on the Beaglebone platform. Our use of the Programmable Real-Time Unit (PRU) to toggle GPIOs to encode the bits frees up the main CPU from this responsibility while achieving a high-frequency and flexible-GPIO control [32,53]. Reports indicate PRU-based GPIO toggling is `~40` times faster than CPU-based GPIO toggling [2,24]. The CPU will have a longer processing delay over GPIO toggling compared to PRU since it needs to run OS and manage all the available onboard resources. Thus, I/O offloading from CPU to PRU can provide the flexibility of resource management while offloading timing-critical tasks related to the VLC RX/TX to a specific coprocessor to handle the transmission or reception in a fast and reliable manner. Further, the software architecture is portable and runs on the BeagleBone board [11] with a standard Debian image.

The PRU is a coprocessor with a 200 MHz clock frequency. It can be utilized for real-time tasks and time-critical responses. Considering this feature, PRU is an ideal processor to run RX/TX software for embedded VLC and achieve high performance. Figure 6.4(a) shows PurpleVLC generating approximately 1 MHz signal to blink the LEDs in the transceiver without using additional hardware such as FPGA or USRPs. Speeds higher than 1 MHz can also be achieved but may degrade in performance due to high-speed signal issues, not due to processing bottlenecks. One MHz toggling has been reported in VLC literature [43] and requires FPGA but PurpleVLC simply utilizes the resources readily available in Beaglebone platform. Thus, one of our contributions is to educate, the community about this under-utilized resource that dramatically improves the performance of the IoT platforms.

6.4.0.2 Using Different Number of LEDs for Transmission Depending on Performance Objectives

Depending on the goals, the system has to control LEDs *concurrently* or *independently*, and both modes are supported by PurpleVLC. We next describe design issues and tradeoffs to support the two modes of multi-LED control.

The capability to change the number of LEDs in the VLC transmitter, enables PurpleVLC to support different scenarios and performance objectives. If the VLC transmitter uses more LEDs, generally more light is generated allowing the photodetectors to detect more light, making the system perform better compared to a single LED. This is especially good for longer distances. More LEDs can achieve a larger field of view (FoV) increasing deployment flexibility: the photodetector does not need to be aligned precisely with the transmitter. Using more LEDs, consumes more energy compared to using a single LED. On the other hand, using fewer LEDs will decrease the amount of light and also decrease the FoV. Thus, PurpleVLC's ability to support different number of LEDs is essential in a flexible VLC platform because different scenarios will have different constraints and objectives.

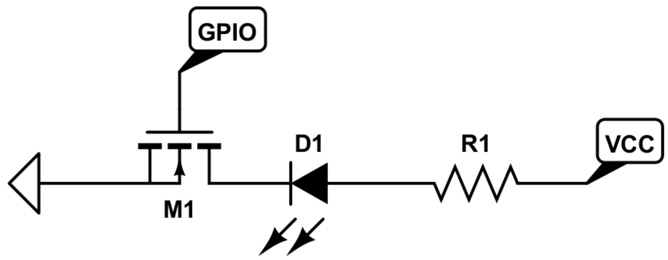


Figure 6.3: GPIO controlled LED. We duplicate the driver circuit to drive 4 LEDs with the GPIOs on board.

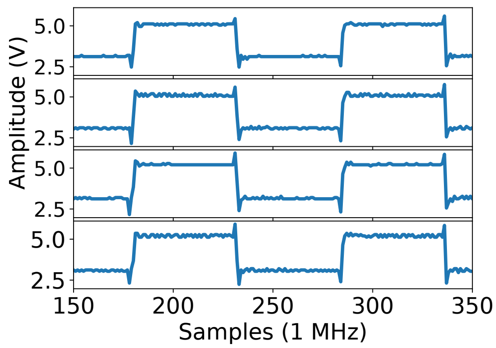
Figure 6.3 shows the circuit for controlling one LED. For off-the-shelf 5 mm LED, the driving current is approximately 20 mA. PurpleVLC has four copies of such circuit. Further, each LED driver circuit can be controlled independently. Controlling such a large number of GPIOs and LEDs independently at high speed using a single-core architecture was not possible for the previous systems from the literature and hence required the IO offloading technique we introduced in this work.

LED Synchronization Issues: If we use multiple LEDs to extend the range and improve reliability, all the LEDs should be controlled synchronously, i.e. all the LEDs are turned on and off at exactly the same time to transmit the encoded information. If extending the range and improving the reliability was the only goal for using multiple LEDs, we would simply hard wire multiple LEDs on the board or use enable/disable switch on the circuit for each LED and still control them from a single GPIO. That way, all the LEDs are precisely synchronously controlled. In our case, we also want to control the LEDs independently (to support concurrent channels) thus, hard wiring or just enable/disable approach is not adequate since each LED requires independent GPIO control. Thus there is a need to synchronize GPIO toggling for data transmission using multiple LEDs. In our design, the PRU can control multiple GPIOs in a concurrent manner. In order to evaluate the performance of the GPIO toggling, the PRU allows us to control the GPIOs independently using an IO register (32-bits) within one clock cycle. Rather than control one pin at a time, we can control a larger number of pins within a clock cycle. We can generate various toggling speeds for the LEDs by modifying the waiting period. We use an oscilloscope to measure the generated clock signal while toggling the LEDs with the driving circuit shown in Figure 6.3. The oscilloscope can sample four channels at 250 MHz concurrently. Figure 6.4 shows the waveform at various toggling frequencies. We find that the signals across the GPIO channels are synchronized due to the precise control from the PRU. At 1 MHz, the four channels has a minimum offset of 0.5 ns and a maximum offset of 1.5 ns; and at 16.7 MHz, we observed the same minimum offset and maximum offset. Although the resolution of this instrumentation is 5 ns, we used a manual cursor on the oscilloscope to measure these sub-5 ns offsets. These offsets are mainly due to the imprecise response time of the LEDs. Typically, these LEDs can respond to toggling within 2 ns. However, at 16.7 MHz, the waveform becomes distorted, making it more challenging to represent symbols using simple encoding schemes. PurpleVLC uses a symbol rate of 100 KHz, thus the 0.5-1.5 ns offset is negligible at this rate. With a resolution of less than 5 ns synchronization across the pins, PurpleVLC is able to generate a stable symbol rate at 1 MHz for multiple LEDs. This level of synchronization would be difficult to achieve from an on-CPU approach, with low utilization, using a Linux driver seen in previous platforms. In addition, users can decide the number of GPIOs to be controlled by the coprocessor and how to control them. For example, the user can configure three LEDs to provide illumination (The LED is always turned on) while one LED will be used to provide communication. Thus, using a co-processor such as a PRU not only enable the control of multiple LEDs when needed to transmit synchronously but also can enable a flexible design without the use of hardware for such precise and synchronous control (e.g., FPGAs or USRPs).

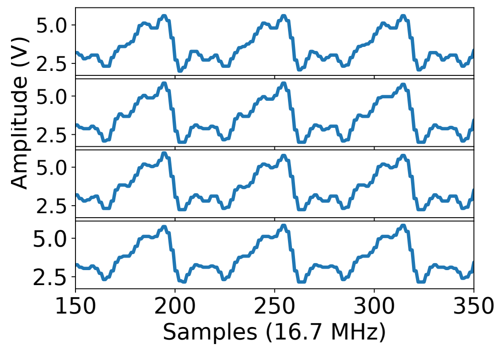
Transmission Power Control: Increasing the transmit power generally extends the range or improves reliability. Using the PurpleVLC's capability to synchronously toggle the LEDs and select the number of LEDs, the user can change the effective VLC transmit power. The other way to provide transmit-power control capability would be the use of adaptive resistors that limit the current flowing through the LEDs. Hence we opt to change the number of LEDs which provide the capability to adjust transmit-power control.

6.4.0.3 Isolating Concurrent Channels

Now we describe how we achieve concurrent channels in embedded VLC. An overview of the concurrent design is seen in Figure 6.10, this Figure shows concurrent data flow from TX to RX in two channels. For software design, we configured one node to run the transmission program and the other node to run the reception program. We used two off-the-shelf photodiodes to detect the light signals with a double channel transimpedance amplifier (TIA). The amplified signal will be sampled using an ADC



(a) 1 MHz signals on four pins



(b) 16.7 MHz signals on four pins

Figure 6.4: 1 MHz and 16.7 MHz signals captured by the oscilloscope on the four pins with current driving circuit. 16.7 MHz signals are significantly distorted making them unsuitable for VLC transmission. 1 MHz signals are clean and tightly synchronized demonstrating the level of control needed for independent and synchronous operation.

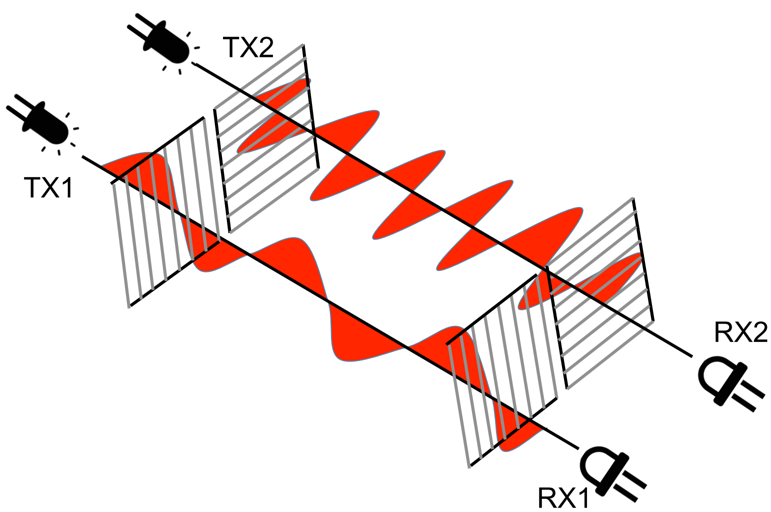


Figure 6.5: The links are isolated because of polarizers which allow data to be sent concurrently without interference.

with simultaneous channel sampling.

In a RF-based communication system, mutual interference across the two channels is the main problem in concurrent communication at the same frequency. However, we use polarizers to cancel the mutual interference so each link can be independent. Polarizers have been explained in [103]. The idea is to isolate the two concurrent transmissions so the receivers can receive the corresponding data on each link. Figure 6.5 shows the concept behind this approach. When the polarization angle between the receiver and the transmitter matches, there is effective VLC TX/RX communication (see Figure 6.6) otherwise not enough light passes through to the receiver resulting in poor communication. (see Figure 6.7). Marus' law tells us how much light passes through the polarizers at different angles:

$$I = I_0 cos^2\theta \tag{6.1}$$

I represents the light intensity after the two polarizers. I_0 is the light intensity between the two polarizers. Marus's law applies if these two polarizers form an angle θ. For example, $\theta = 0$ in Figure 6.6, $\theta = 90°$ in Figure 6.7.

Figure 6.8 shows how we implement the design in practical systems. We created a cylindrical LED housing/case with polarizers on the front and insert the LED inside this housing. Light emitted by the LED goes through the polarizer in the front. The housing can be physically rotated to achieve the desired angle of polarization. Light from the two LEDs will be light beams with different polarization. The receivers LED or PD also have cylindrical polarizer. We turned on two LEDs in one node. Figure 6.9 shows the PD's view after light passing through the polarizers. We observed only one LED illuminating from each photodiode demonstrating isolation between the

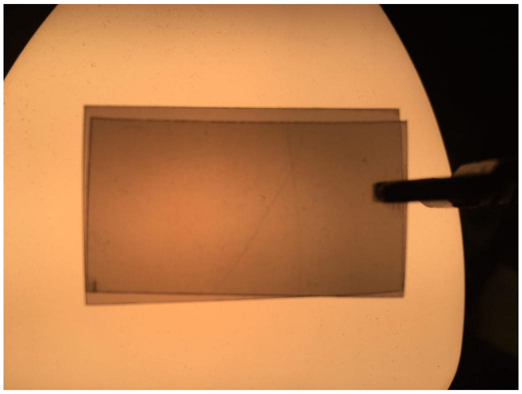


Figure 6.6: Polarizers in parallel orientation.

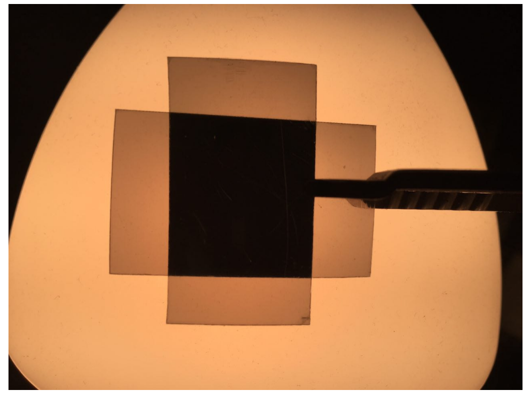


Figure 6.7: Polarizers in orthogonal orientation.

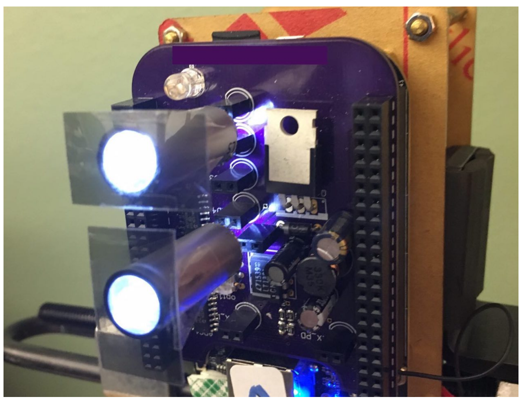


Figure 6.8: TX: vertical and horizontal polarizers.

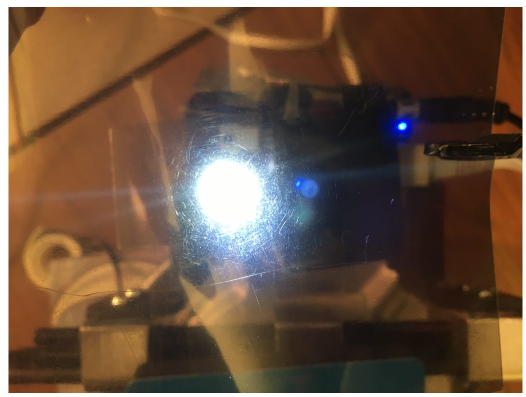


Figure 6.9: RX's view of the transmitter.

two concurrent channels.

6.4.1 Full-Duplex VLC

Existing embedded VLC platforms are either simplex or half-duplex, i.e. they can only transmit or receive at one time even though they support bi-directional communication. In a single MCU-based design, half-duplex communication is naturally

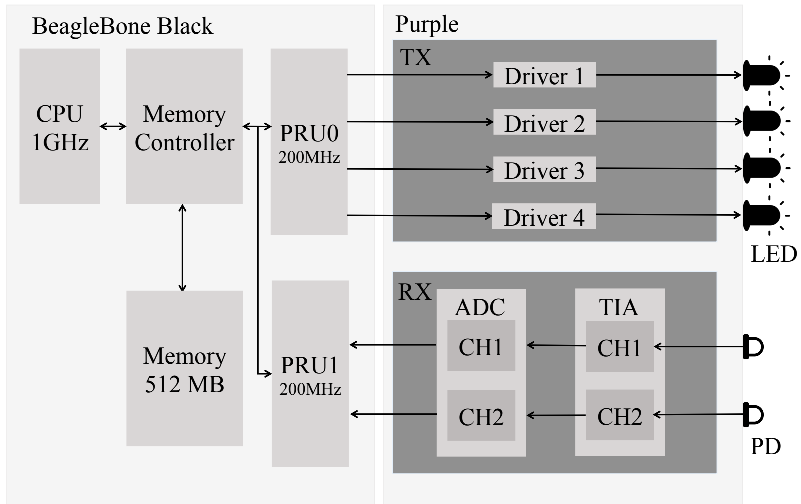


Figure 6.10: System architecture to enable concurrent communication.

supported: the MCU can run the transmit logic or receive logic at one time. However, in single-board computer-based design, such as the OpenVLC1.0, have sufficient on-board resources to provide full-duplex communication. This can potentially increase communication throughput and reduce latency. Full duplex communication has been explored in RF communications for decades. The problem for RF full duplex is self-interference, which means, the TX will interfere with RX on the same node. In the RF domain, a complex cancellation circuit is needed to cancel the self-interference, particularly in the omnidirectional RF.

Light transmitted by LEDs is directional, selection of LEDs, careful placement, and orientation of the transmitting elements are sufficient to prevent self-interference. However, one also needs appropriate hardware and software architecture to allow concurrent execution of transmission and reception logic and IO operation. Thus, two

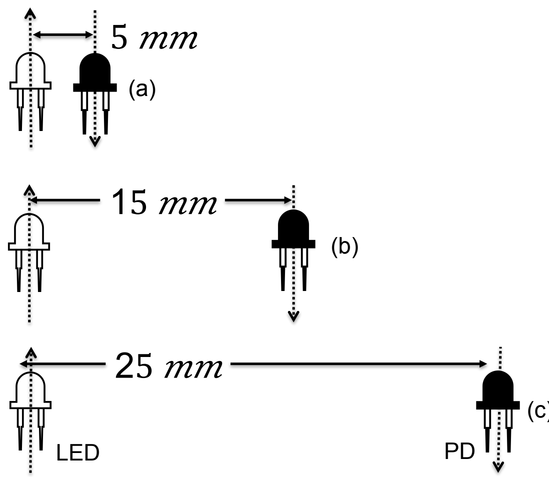


Figure 6.11: Three configurations for antenna distance.

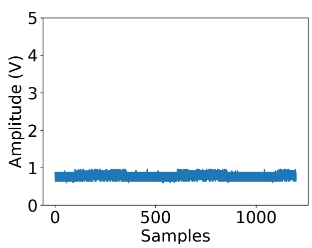


Figure 6.12: Received signal with antenna distance of 25 mm.

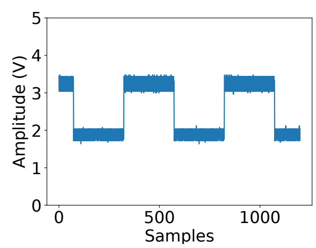


Figure 6.13: Received signal with antenna distance of 15 mm.

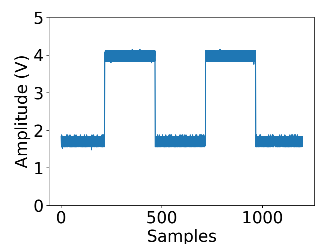


Figure 6.14: Received signal with antenna distance of 5 mm.

capabilities are needed in full-duplex VLC platform for the prevention of self interference and concurrent TX and RX processing. PurpleVLC provides both capabilities by utilizing resources available on Beaglebone.

In PurpleVLC architecture (Figure 6.10), the two PRUs (coprocessor) can operate independently supporting both transmission and reception at the same time. We configured the platform to connect one LED to PRU as the TX and one PD as the RX. The onboard TX and RX operate in parallel to achieve full-duplex point-to-point

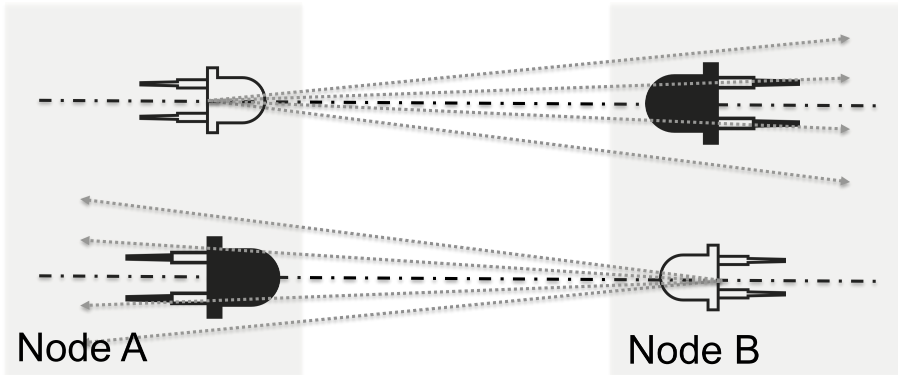


Figure 6.15: Configurations to enable full duplex communication.

communication.

Self-interference in full-duplex VLC. The level of self-interference that can occur depending on the placement of the LEDs on the VLC board. More specifically, we ran experiments to determine the level of self-interference as a function of antenna distance on the board. Figure 6.11 shows the configurations used for the antenna distance. The distances between LED and PD were at 5 mm, 15 mm, and 25 mm. The LED is highly directional, the LED and PD with transmitting and receiving beam was placed at an angle of 180° at each distance. We configured the LED to send continuous periodic signals and observed the received signal from the PD using an oscilloscope with passive sampling. We ran the experiment in a lab environment where the ambient light was always on. The expected received signal was constant as we assumed the PD was only detecting the ambient-light intensity.

We observed the received signal had minimum interference from LED when the antenna distance was 25 mm as shown in Figure 6.12. We found that this level of interference was negligible on full-duplex VLC communication. With a 5 mm distance between the LED and PD, there was strong interference from the LED to the PD as shown in Figure 6.14. When the LED and the PD were side by side at a distance of 5 mm, theoretically, the PD should not receive signals from LED since it is out of the LED's FoV. However, the directional LED used also emitted photons outside of its FoV. The imperfect manufacturing process might cause unpredictable FoV. For the antenna distance of 15 mm, the received signal had weaker interference from the LED (Figure 6.13) compared to 5 mm. For each distance, we also slightly rotated the PD to form a different beam angle between LED and PD ranging from 150° to 180°. The rotation for each antenna distance was similar, but the pattern was the same. The interference from LED became weaker at larger antenna distances. Detailed result will be presented in the evaluation section.

6.4.2 Encoding and Decoding

6.4.2.1 Transmission

Our transmission mechanism consists of two main parts: (1) framing and (2) modulation. Similar to any asynchronous communication system, a frame format for data transmission is required to ensure that the receiver is able to detect and receive the packet. We use the frame format shown in Table 6.2. By including a preamble, the receiver was able to adapt its threshold to distinguish between 0 and 1 symbols. To further improve the reception, the preamble can be sent continuously when no data

Table 6.2: Data Transmission Frame Format

Preamble	**SYNC**	**SFD**	**Data**	**EFD**
0xAAAAAA	0xD5	0x02	N bytes	0x03

is available. Start of frame delimiter (SFD) indicates where the receiver should start reading the packet. The end of frame delimiter (EFD) indicates that the packet has ended and is a constant and weak form of error checking compared to the Cyclic Redundancy Check (CRC).

We modulate every byte with on-off keying (OOK) with Manchester coding. Symbol sequence of 10 represents bit 0 and a symbol sequence of 01 represents bit 1. To keep the LED in a light emitting state, we added a 0 start bit (10 symbol) and a 1 end bit (01 symbol) to each byte. A delay was added between the transmission of each symbol which defined the symbol duration and consequently the data rate.

For dual-link communication, the transmitter sent packets in two channels at the same time, with a delay of 1 instruction cycle (5 ns) for the second packet. This implementation runs on the PRU and is based on a polling-base approach instead of interrupt-based approach.

6.4.2.2 Reception

The decoding process was specific to the encoding schedule used in our design. The raw samples were extracted from the ADC, and the decoding algorithm was run to recover the packets.

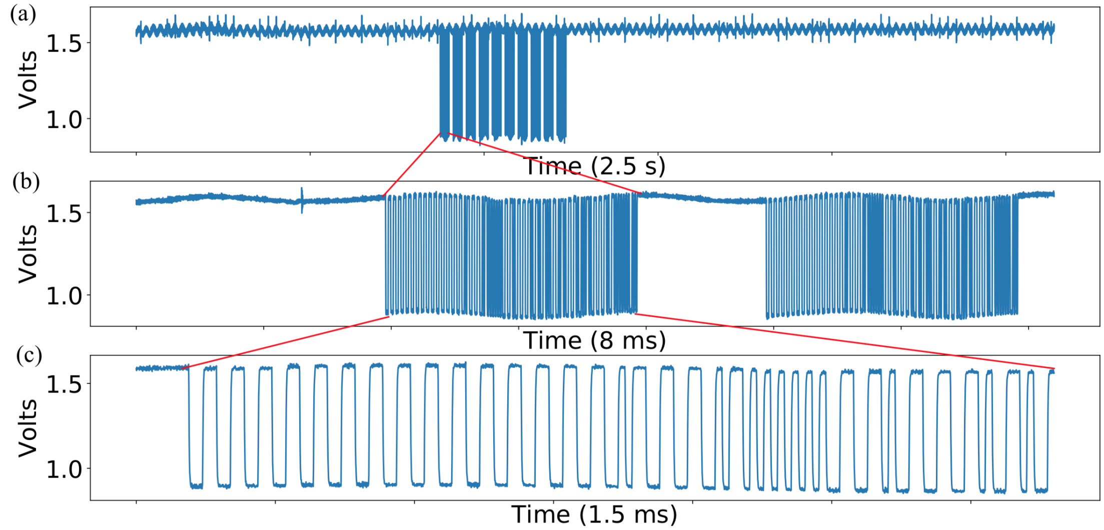


Figure 6.16: Raw data collected from onboard ADC.

Bit Detection. The decoding algorithm was based on a sampling-based Manchester decoding [9]. We identified the rising and falling edges as bit 1 and bit 0 using an adaptive thresholding from preamble symbols. We first decoded the raw samples into a bit sequence, and the predefined preamble was used to synchronize the packet to find the starting bit. Figure 6.16 shows the general reception process. The Figure 6.16(a) shows the reception of 10 packets. The Figure 6.16(b) shows the zoomed in version for one packet. The Figure 6.16(c) presents details of the symbols for one packet.

6.4.2.3 Communication Latency Analysis

We characterize the communication latency in our system from TX to RX by taking into account delays in different modules in the design. Figure. 6.1 shows the data flow

Table 6.3: Propagation Delay Characterization

System Module	**PRU Modulation**	**LED**	**PD**	**TIA**	**ADC**	**PRU Demodulation**
Propagation Delay	20 μs	5 ns	5 ns	10 μs	2 μs	20 μs

from TX to RX based on half-duplex's setting . We sent information from the Linux Userspace to PRU through a character device. The PRU received the information and ran the encoding algorithm by toggling LEDs in a deterministic manner. For a symbol rate of 100 KHz, it takes one clock cycle (5 ns) to change the LED status, and 10 μs to transmit one symbol. Two symbols are needed to represent one bit. Thus 20 μs is required to transmit one bit. Meanwhile, the LED takes 2 ns as a response for changing its status. The processing delay on PRU modulation for one bit takes about 20 μs. On the RX side, the symbols were detected by the photodiode first, the response time for the photodiode is 5 ns [14]. The TIA will then amplify the detected symbols. The bandwidth of the transimpedance amplifier determines the upper bound of the transmission speed. In our design, we selected a 2 MHz bandwidth operational amplifier [16] which was adequate for the symbol rate used.

Once the signal was amplified by the TIA, it goes into the ADC for conversion into a digital form. Based on observations, we selected the ADS7783 as the ADC for our circuit since it has a sampling rate of 3 msps and a previous approach used to sample signals with 1 msps using the PRU [71]. The symbol/clock rate should not be larger than 100 KHz. The delay at each module is summarized in Table 6.3.

6.5 Evaluation

Now, we describe various aspects of the system performance.

6.5.1 System Implementation

We connected the PurpleVLC board to BeagleBone Black (BBB), which is a single-board computer, in order to assemble the full transceiver system. We implemented the firmware running on the PRU to send and receive packets. We loaded the firmware into the PRU core using the Remoteproc framework on top of Debian Linux running on BBB. We used PRUDAQ, a high-speed ADC-sampling board [12] to passively collect data from the PurpleVLC board for performance analysis. Figure 6.17 shows our experiment setup for the system evaluation. We used an assistant tool to set up the platform for experimentation. We implemented the firmware running on PRU and made the data communication controllable in the embedded Linux environment running on the BBB. We then dumped the data into a file for offline analysis. The system was evaluated in both an office room and a corridor with constant ambient light.

6.5.2 Metrics

The following three metrics were used to evaluate the system performance with the experimental data gathered from our testbeds.

Packet Loss Ratio: This metric (PLR) represented the failure rate for the link layer packets transmitted to the receiver. This metric was used to assess the quality of established visible light links.

Bit Error Rate: This metric (BER) represented the failure rate for the bits transmitted through the physical layer. It was used to evaluate the channel quality in the physical layer.

Effective Throughput: This metric represented the amount of the successfully received bits per second and we use it to quantify the PHY layer performance since this is a widely used experiment metric in previous studies [27, 102]. It is usually utilized as a metric for the transport layer of a TCP/IP network model. However, we used effective throughput to quantify the correctly received bits in the detected packets.

6.5.3 Single Link Throughput

We evaluated the performance using different numbers of LEDs on our platform for a single-link scenario.

Experimental Setup: We configured TX and RX to operate with a symbol rate of 100 KHz. We first placed the TX and RX nodes at a distance of 2.5 m and ran the experiment with 100 packets. Each packet consists of 1000 bytes. Then, we increased the distance at 0.5 m steps and repeated the experiment. At each distance, we repeated the experiment with different numbers of LEDs.

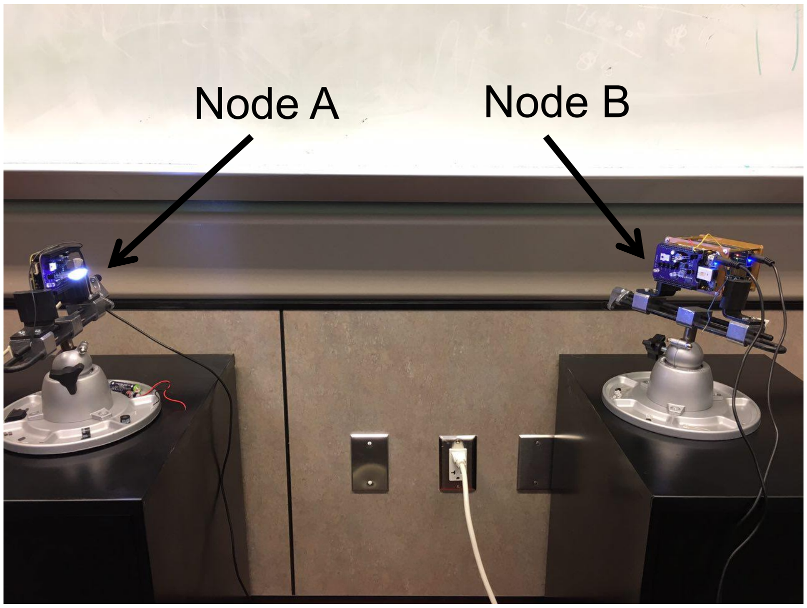


Figure 6.17: Experimental Setup

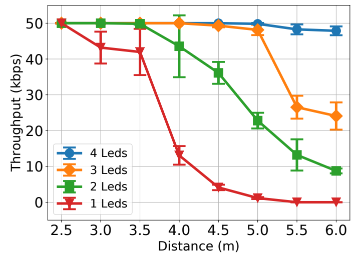


Figure 6.18: Throughput under various distances with different numbers of LEDs.

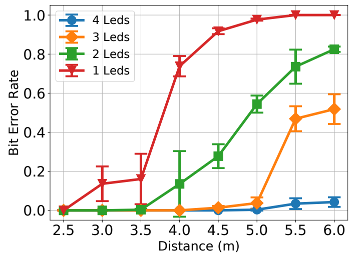


Figure 6.19: Bit error rate vs communication range using a symbol rate of 100 KHz.

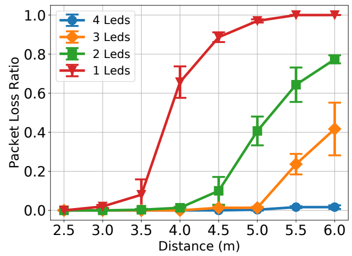


Figure 6.20: Packet loss rate vs communication Range using a symbol rate of 100 KHz.

Result: Figure 6.18 shows the communication performance using different numbers of LEDs. It is clear from the figure that more LEDs resulted in a higher throughput at the same distance. With a larger number of LEDs, the generated illuminance was higher, and more light signals were detected by the photodetector, leading to a higher SNR. This observation indicated that adaptive transmission power was feasible using our system. Meanwhile, as seen in Figure 6.20 that we can achieve 99% reliability (PRR) when using 4 LEDs with all distances up to six m. For three LEDs, the reliability starts to drop at five m. These can be verified through bit error rate seen in Figure 6.19. The BER for our system with 4 LEDs was close to 0, while the BER with 1 LED was 100% with a communication range of 6 m.

Figure 6.19 shows the system has a working distance up to 6 m with less than 1% BER. This is sufficient to validate the effective throughput for indoor scenarios, especially for room-area wireless communication. Meanwhile, the design can be scalable for the transmitter since more LEDs can be added to current design to increase the communication distance. It also suggests that multiple LEDs controlled by a cheap single-board computer are rapidly deployable to replace current home light bulbs for both communication and illumination purpose. Our design can be tuned to contain both PDs and LEDs in a single package as the front-end, like the flashlight, making it feasible to design smart bulbs that can send and receive data at the same time.

Figure 6.20 shows the link quality for multiple transmitter evaluation. The packet loss ratio is less than 4% with a distance up to 6 m. This is far better than state-of-the-art OpenVLC link quality demonstrated in [44], where the packet loss rate goes beyond 20% within 2 m. With this performance, it is feasible to achieve room-area networking. Here we demonstrate a highly reliable visible light link for data communication with a symbol rate of 100 KHz. The symbol rate has not been reported with this level of reliability in embedded VLC literature. The link quality for embedded visible light communication can be as reliable as possible in a static scenario where both TX and RX does not require mobility.

6.5.4 Concurrent Communication Performance

Now we study the communication performance when two concurrent channels were used.

Experimental Setup: We configured the transmitter to transmit packets on two links, concurrently. The packets transmitted on each link are different. We used the polarizing setup described earlier (LED housing with polarizer at one end of the cylinder) to isolate the two channels. We then rotated the polarizing case so the two channels have opposite polarization (90 degrees). We placed the two transmitters side-by-side as shown in Figure 6.8. The TX and RX were placed one meter apart and the experiment was run with 100 packets for each link at a symbol rate of 100 KHz for each link. The system supported simultaneous ADC-sampling from different channels which enables a dump of the data from these two links simultaneously. We repeated the experiment with the same setup with a single link to perform the comparison. Mutual interference between these two links was canceled using the polarizers with crossed orientation. The effective throughput for concurrent channels was expected to be twice the throughput for a single link. BER for dual link and single link was expected to be similar.

Table 6.4: Throughput and BER with different number of links

Metrics	Single Channel	Concurrent Channel
Throughput (kbps)	49	95
BER	0.001	0.025
Packet Loss Ratio	0.0135	0.025

Result: Table 6.4 shows the system performance with different numbers of links. Concurrent communication achieved nearly twice the throughput of the single link indicating the effectiveness of the polarizer setup to isolate the channels. The BER for the dual link was comparatively higher than with a single link likely due to an imperfect polarizer and some cross-talk light that escaped the LED case. The link layer for concurrent transmission had a higher packet loss ratio compared to a single link. However, the overall link-reliability for concurrent communication was 97.5%, which still validated our design was efficient to cancel the mutual interference for concurrent communication. We also tried other approaches to cancel mutual interference and construct concurrent channels. We used the red and blue LEDs as a dual-link concurrent transmitter and the red and blue LEDs as a dual-link concurrent LED receiver. Since the red and blue LEDs have different operating light frequencies, we expected them to be interference-free. But the transmitted signals could not be recognized by the receiver design. The low-cost indicator-type LEDs do not have perfect matching responses to light frequencies. For example, we can receive blue light signals using the red LED as the receiver, but we can't receive red-light signals using blue-LED receiver.

6.5.5 Full-duplex Throughput

Experimental Setup: In this section, we evaluated the full-duplex communication capability of PurpleVLC. We focused on the aggregate throughput gain compared to half-duplex system. We configured two nodes to transmit and receive packets simultaneously. We also ensured the packet transmission for the two nodes was overlapping. There should be two active flows at one time compared to half duplex. We ran our experiment in a lab environment with normal lighting. We configured the distance between the two node as 2.5 m. We used aggregate effective throughput introduced in [30] as the metric here to evaluate the performance. For running half-duplex mode, we calculated the aggregate effective throughput by averaging the two single-directional links. For full-duplex mode, we calculated the aggregate effective throughput by adding the throughput for each direction when both links were active. For evaluating full-duplex mode in RF scenario [101], we doubted whether the full duplex in visible light communication could double the throughput.

Result: We configured various distances between LED and PD to observe the impact on the visible-light link reliability for both full-duplex and half-duplex mode. The result is shown in Figure 6.21.

For the half-duplex mode, the photodiode received the light signals transmitted from the other node, at roughly 50 kbps. This suggested the aggregate throughput with half-duplex design would not be affected by the distance between the LED and

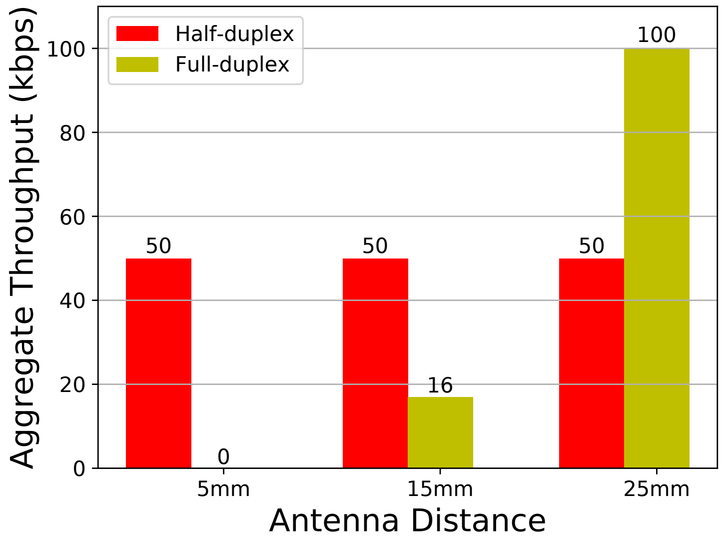


Figure 6.21: Aggregate throughput over various antenna distance.

PD on the same board. However, it is interesting to note the aggregate throughput for full duplex design. In Figure 6.21, the aggregate throughput is 0 kbps when the LED and PD were placed 5 mm apart. The onboard ADC would only accept the voltage between 0-2 V. At this distance, the PD received too much light causing the amplified signal larger than the threshold of the ADC . The offline processing would not detect signals, causing the aggregate throughput to be 0. At 25 mm, there was minimal interference on the PD from the LED. At 15 mm, the received signal for the photodiode included both the signal from the other transmitting LED and the onboard transmitting LED. From Figure 6.21, the aggregate throughput was 16kbps. The link was highly unreliable since most packet were dropped during the communication. We found that in average 92% and 62% packet loss ratio was seen for the two active links. We suggested board designers to place the LED and PD

Table 6.5: Power consumption for one laser diode and one LED

	Current (mA)	Voltage (V)	Power (mW)	Cost ($)
Laser Diode	25	2.5	6.25	5.96
LED	21	2.9	6.19	0.44

at least 25 mm apart to achieve the best full-duplex performance in an embedded visible light communication. We can further demonstrate a quadruple performance boost with the current design since we can configure the platform to enable full-duplex mode for dual-link communication. The potential gain can be 4 times over half-duplex mode.

6.5.6 Comparison between Low-cost LEDs and Laser Diodes

Next, the VLC communication performance was studied with low-power laser diodes vs one or multiple LEDs. Our platform also support off-the-shelf low-power laser diode, such as 5 mW or 1 mW laser diode, and the laser diode is energy efficient and more reliable with focused light beams. It also demonstrates a longer communication range compared to a single LED. Table 6.5 summarized the power consumption between LED and laser diodes. We configured the PurpleVLC to use four LEDs and ran the experiment in a long corridor in an academic building. We repeated the experiment with a low-power laser diode. Figure 6.22 shows the throughput between 4 m and 30 m for a laser diode and four LEDs. The laser diode maintained the same reliability and effective throughput in all distances between 4 m and 30 m. However,

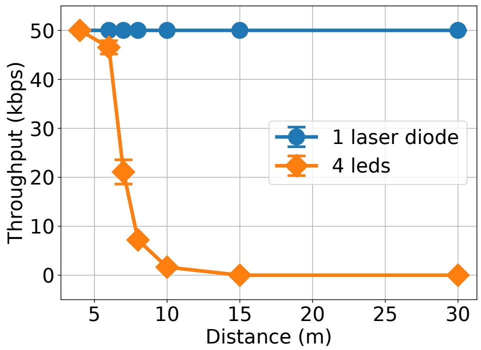


Figure 6.22: Throughput calculated at various distances.

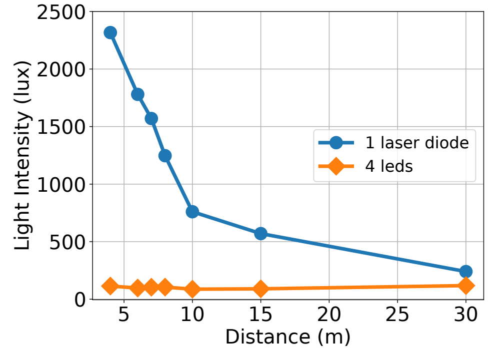


Figure 6.23: Light-intensity measured at various distances.

the throughput dropped significantly from 6 m to 8 m using the four LEDs. In order to understand the impact from the light intensity received by the photodiode, an experiment was run to collect the LUX value at the different distances using the light sensor TSL2561. We plotted the LUX value as seen in Figure 6.23. It shows light intensity emitted from laser diode is higher compared to LEDs at shorter distance, such as 4 m. It then goes down as distance increases. Light emitted from LEDs was less focused, and the receiver registered constant light intensity due to the ambient light in the corridor. The laser diode has a safety classification of IIIa, which is the same classification used in laser pointers. It is safe, but it still requires careful handling [8]. We demonstrate the possibility of further increasing the performance of low-cost embedded VLC using low-power laser diodes while still remaining within the energy budget. These diodes require careful handling than standard LEDs but are feasible to use in indoor spaces and useful as options to allow performance, power, and deployment constraint tradeoffs for different applications.

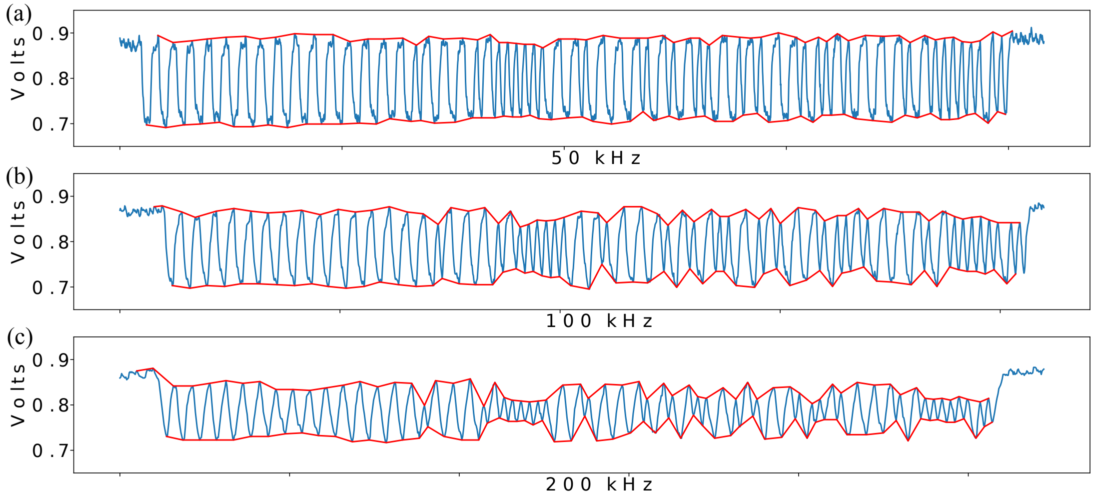


Figure 6.24: Impacts of symbol rate on reception.

6.5.7 Maximum Symbol Rate

PurpleVLC's performance bottleneck was studied, particularly the stability of reception at higher symbol rates. The symbol rate represents the clock rate used in our communication system. Since we are using OOK with Manchester coding, the bit rate for our system is half the symbol rate.

Experimental Setup: We configured the TX and RX to be within a distance of 5 m, We configured the TX to transmit packets at a symbol rate of 50 KHz and then repeated the experiment for a symbol rate of 100 KHz and 200 KHz.

Result: Figure 6.24 shows the received signal (one packet) sampled by the ADC when the system was configured using different symbol rates. Figure 6.24(a) shows that the detected high and low symbols were stable over a symbol rate of 50 KHz. The

Table 6.6: Standard Deviation for Local Peaks and Local Valleys

	50 KHz	**100 KHz**	**200 KHz**
Std. for Peaks	0.758%	1.265%	2.104 %
Std. for Valleys	0.887%	1.561%	2.213 %

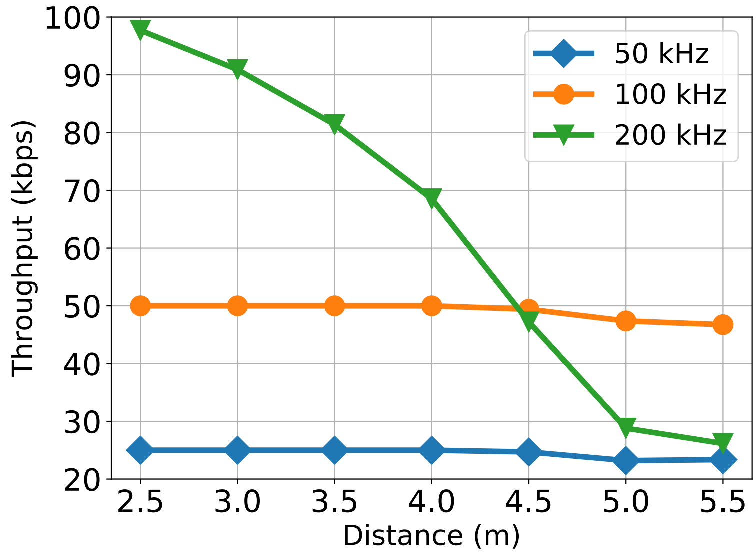


Figure 6.25: Throughput under various distances with different symbol rates for 4 LEDs.

detected high and low symbols become more noisy when the symbol rate becomes higher due to the limitation of the TIA design as shown in Figure 6.24(b). We selected an operational amplifier (two dollars each) with a bandwidth of 2.2 MHz to serve as the TIA. The amplified signal was distorted when the symbol rate went from 100 KHz to 200 KHz. For 200 KHz, the distortion was the worst as shown in Figure 6.24(c). It was extremely difficult to detect the fluctuating signal. For 100 KHz, the system was operated as expected, which makes it an optimal symbol rate for long-range experiments regarding to the performance. This also explains why the designed system performed well in terms of various symbol rates as seen in Figure 6.25. Effective throughput was expected to be at the maximum symbol rate of 50 KHz and 100 KHz. However, the rate dropped significantly for longer distance at a symbol rate of 200 KHz. We quantified the standard deviation of the local peaks and local valleys to show the fluctuation in Table6.6. The amplified signal at 200 KHz symbol rate was unstable. We can further improve the current TIA design with a high-speed operational amplifier. However, that also brings a trade-off in terms of high-speed circuit design and system complexity.

6.6 Conclusion

In this work, we designed and implemented PurpleVLC: a novel embedded VLC platform that integrated I/O offloading, concurrent channels with polarized light and full-duplexing to significantly improve the performance for low-cost embedded VLC platform. The results indicate that we can achieve an aggregate throughput up to 100 kbps given an operating distance up to 6 m. Our design combines I/O offloading, concurrent channels, and full-duplexing to offer more than 99%-link reliability. We expect more reliable and useful applications to be developed based on our platform. We believe this can be the complimentary technology other than RF for wireless connectivity in a room area.

Chapter 7

VLC in Education

OpenVLC1.0 and PurpleVLC, developed in this dissertation, have been used in teaching systems & networking related courses. These platforms are closely related to hardware and software integration and prototyping. It is useful to engage students from computer science and electrical engineering departments with hands-on skills so the students can better understand the concepts. We describe classroom workshops using these platforms and lessons learned in relevant courses.

7.1 Platform Availability

We have open-sourced the design details for PurpleVLC boards and shared it via Github [13]. Users from all around the world can fetch the design and send the design files to manufacturing vendors for final PCB (printed circuit board) production. Then they can find a bill of materials (BOM) from vendors such as Sparkfun or Digi-Key and assemble them to make it fully functional. We ordered parts for 30 boards and assembled them. During classroom workshops, students only need to log in to the lab/personal computer and connect to the board to explore the functionality of the kit.

7.2 Classroom Workshops

We organized classroom workshops with OpenVLC1.0 and PurpleVLC in eight courses including Computer Organization and Architecture, Introduction to Computer Networks, Computer Networks, Microprocessor Systems, Embedded Microcomputer Systems, Optical Networking, Senior Design Lab, and Newwork Interconnection in US and Mexico. For each classroom workshop, we distributed one survey before and one survey after the workshop to understand how students learn from this experience. We list all courses where we conducted these workshops in Table 7.1.

- **Computer Organization and Architecture (COSC 2440)**: This is an undergraduate level course offered by the Department of Computer Science at the University of Houston. The course covers low-level computer design, basics of digital design, hardware/software interface, principles of pipelining and caching, instruction pipelining, and multiprocessor systems. Students enrollment for this course was 162 for Spring 2017, 177 for Fall 2017, and 178 for Spring 2018. There are about 27 lectures offered per semester. Each lecture

Table 7.1: Courses integrated with classroom workshops using OpenVLC1.0 and PurpleVLC.

Course Number	**Course Name**	**Enrollment**	**Country**
COSC 2440	Computer Organization & Architecture	517	US
COSC 4377	Introduction to Computer Networks	60	US
COSC 6377	Computer Networks	33	US
ECE 3436	Microprocessor Systems	45	US
ECE 4437	Embedded Microcomputer Systems	35	US
ELET 4208	Senior Project Lab	40	US
ELET 6307	Optical Networking	7	US
TC 2022	Network Interconnection	25	Mexico

is about 80 minutes. There are four lab sections for the course per semester. There are about 40-45 students enrolled in each lab section. We provided both OpenVLC1.0 and PurpleVLC platforms in each lab section so that all students for this course can benefit from this adoption. Students in the lab finished step-by-step instructions during the lab workshop. We organized three lab workshops in the Spring 2017, Fall 2017 and Spring 2018 semesters.

- **Introduction to Computer Networks (COSC 4377)**: This is an undergraduate level course offered by the Department of Computer Science at the University of Houston. The course covers data communications, network protocols and architecture, local and wide-area network, and internetworking. Enrollment for this course was 60. There are two lab sections for this course. We organized two lab workshops in the Spring 2017 semester.

- **Computer Networks (COSC 6377)**: This is a graduate level course offered by the Department of Computer Science Department at the University of Houston. The course covers concepts and technical foundations of computer networking from application layer to physical layer. The course then explores new frontiers in computer networking research by reading papers. Enrollment for this course was 33. The classroom workshop for this course was provided with a guest lecture introducing PurpleVLC as well as the in-class demo for the embedded visible light communication. Students explored OpenVLC1.0 and PurpleVLC in class and followed up with the slides on how to communicate via visible light. We organized one workshop in the Fall 2017 semester.

- **Microprocessor Systems (ECE 3436)**: This is an undergraduate level course offered by the Department of Electrical and Computer Engineering at the University of Houston. The course covers microprocessor architecture, instruction set, addressing modes, assembly/C language programming, debugging, I/O devices, programming and interfacing. Enrollment for this course was 45 students. We provided PurpleVLC platform to students in a guest lecture by providing step-to-step instructions with slides. We organized one workshop in the Fall 2017 semester.

- **Embedded Microcomputer Systems (ECE 4437)**: This is an undergraduate level course offered by the Department of Electrical and Computer Engineering at the University of Houston. The course covers the HW and SW interfacing methods, embedded software development, real-time debugging, interrupt synchronization, threads, timing generation and measurement, serial I/O devices, parallel port interfacing with sensors, displays and motors, data acquisition systems, and sampled signals, real-time operating systems. Enrollment for this course was 35. We organized one workshop in the Spring 2018 semester.

- **Senior Project Lab (ELET 4208)**: This is an undergraduate level course offered by the Department of Engineering Technology at the University of Houston. This course helps students design their senior projects in their final year of college. Enrollments for this course was 40. We organized one classroom workshop in the Spring 2018 semester.

- **Optical Networks (ELET 6317)**: This is a graduate level course offered by the Department of Engineering Technology at the University of Houston. The course covers fundamentals of optical communications, control and management, network survivability, access networks, photonic packet switching, and deployment considerations. There were seven students enrolled in this course. We organized one workshop in the Fall 2017 semester.

- **Network Interconnection (TC 2022)**: This is an undergraduate level course offered by the Department of Computer Technologies at the Monterrey Institute of Technology and Higher Education, Monterrey, Mexico. The course covers protocols, methods and standards of the network, transport, session and presentation layers of the OSI model, and the TCP/IP model, as well as a basic knowledge of network security. The course is instructed in both English and Spanish. Enrollment for this course was 25. The instructor and the top students helped organizing the classroom workshop in the Spring 2017 semester.

7.2.1 Workshop Format

We organize the workshops in the following sequences:

- Settle on a time and location for a guest lecture for the course. We contacted instructors who wanted to collaborate with us. We agreed to organize the guest lecture for the course as workshops. Each workshop took 90 minutes.
- Prepare for the workshops. We prepared 30 boxes. Each box contained one PurpleVLC/OpenVLC platform, one USB 2.0 A-Male to Mini-B cable, one LED, one Photodiode, and one micro SD card. We also printed physical copies of pre-surveys and post-surveys. We prepared instructions/slides for the workshop presentations. We requested the course instructors to send the instructions/slides electronically one day before the workshop. For a lab workshop, the instructor would send out instructions/slides one week before the lab so students could prepare, and we arrived at the lab location 30 minutes ahead of the lab to install the required software on the lab computers before the workshop.
- Distribute pre-surveys and collect them back in 5 minutes. At the beginning

of the workshop, we distributed the anonymous pre-surveys as a way to understand students' background and interest level toward the workshop and collected after about 5 minutes.

- Introduce and discuss single-board computer and embedded visible light communication with a presentation lasting about 10 minutes.
- Distribute hardware boxes to each student or student pairs. We described all required components inside the box and taught students how to plug them together and to their computer and how to power on the device.
- Follow the step-by-step instructions/slides. We demonstrated live how to connect, configure, edit, compile, and install sample program on embedded Linux using OpenVLC1.0/PurpleVLC.
- Observe LED patterns and help students debug the program to see expected results. We walked around the classroom to help students debug their program.
- Distribute post-surveys and collect them in 5 minutes. We distributed post-surveys at the end of the workshop in order to obtain students' feedback on the workshop.
- Ask students to disassemble the components, put them back into the box and return the hardware boxes. We collected all hardware boxes for future workshops.

- Convert the physical survey into electronic format. We analyzed survey result and modified instructions for future workshops.

In all but one workshop, the teaching team consisted of the guest lecturer (myself) and the regular TAs assigned to the course. In TC 2022, we included three top students as part of the teaching team. The students who helped learned technical and teaching skills and the other students received some peer-learning experience.

7.2.2 Workshop Challenges

We faced and overcome several challenges when we ran these workshops. Next, we discuss those challenges:

- **Lack of student's familiar with Linux**: The lab tutorials require basic C and shell programming knowledge in Linux as well as version control tools. Many students lacked this knowledge. It is really important for us to know the student's background before we proceed with the classroom workshops.
- **Limited number of TAs and long setup time**: The workshop requires the lab computer to be installed with several software tools, such as Putty, Beaglebone Black drivers, and micro SD card flashing tools. The TAs have to install the software before students start to work on the platform. This takes time since we have a limited number of TAs and we have around 40 lab computers requiring installation in sequence.

- **Limited number of kits**: Assembling OpenVLC1.0 and PurpleVLC is very labor intensive with around 200 soldering pads. We built 30 boards for the classroom/lab workshops. Because of the limited number of kits available, students had to work in pairs or we had to divide a lab into multiple sessions, each section with a fraction of students. With more kits, we would have been able to hold larger workshops or could have provided each student their own kit.

- **Network Connectivity**: Some students had difficulty connecting the kits to their computers or accessing the Internet from the kit to download the software modules for the workshop. Network configuration on the kit was not as robust as we would have liked in the initial workshops.

- **Driver issues**: Our workshop required students to bring their personal laptops to connect to the board. We need to provide links for the students to download and install drivers on their computers. This sometimes caused driver-compatibility issues.

7.3 Lessons Learned

We summarize the lessons learned from these classroom workshops and the future improvements in this section.

- Students liked the hands-on learning. During most of the classroom workshops, we taught students how to edit, compile, and install software on a single board computer. Most students were able to finish the instructions from the workshop. They were excited to be engaged and make things work.

- The instructions need to be clear from the students' point of view. The students found screenshots about the step by step procedure very useful.

- The instructions should consider the students' background, especially, their knowledge with embedded systems. The time-consuming part in the lab was formatting the micro SD cards, and connecting to the board through SSH. Some students had difficulty working with git. A brief tutorial about git and other basic Linux topics would be useful. Students with an EE background are more comfortable with our platform compared to those with a CS background.

- It is a good idea to incorporate these workshops in the courses with a clear-grading strategy. One approach is to treat these workshops as a standard homework and the students can explore the kit further for bonus credit.

- A seprate tutorial on Linux before the workshop will improve the student experience with the labs. We found out students from both engineering and computer science departments at the University of Houston lack basic Linux skills.

- The example projects need to be customized to create more interest in the students. Based on our survey, students who enrolled in a senior design project class show the highest interest toward building hardware and software integrated systems.

- Command line-based control of the kit and experimentation is not user-friendly. We need to design better intuitive online tutorials and easier-to-use scripts or graphical interfaces for some parts of device configuration.

- Limited audiences. We organized classroom workshops physically in eight courses in the US and Mexico during one year. Logistics and platform distribution took a lot of time. In the future, we want to develop example course projects based on PurpleVLC to offer online courses so that we can distribute our educational practice for global access.

Chapter 8

Conclusion

In this dissertation, we first study the wireless interference in RF spectrum, demonstrating wireless interference on unlicensed bands. We understand that cross-technology devices interfere each other since they share the same unlicensed RF spectrum. We propose two technique to utilize and avoid this interference for better reliability and robustness. One technique is to utilize the interference by creating a channel for information transmission. The other technique is to avoid the interference by exploiting low-cost, low-complexity but robust and high-speed embedded visible light communication. To be specific, we make the following contributions:

Understanding Wireless Interference. We designed experiments and conducted wireless-performance measurements in a practical environment and made the following observations: Channels on 2.4 GHz sometimes experience high levels of 802.11

interference. Channels on 900 MHz band provides a more reliable connectivity. Different from previous approaches, our experimental study was done on a large-scale dual-band testbed.

Interference Utilization: Cross-Technology Communication. We propose a new technology called cross-talk communication (CTC) that can utilize the interference in a way to build a channel for information transmission. We implemented the proposed technique on real WiFi and 15.4 devices and performed extensive experimental validation of the technique in both controlled and uncontrolled environments. We achieved a data rate of 2 bytes per second with less than 10% bit error rate in uncontrolled environments.

Interference Avoidance: Robust Embedded VLC. We present the circuit and physical-layer designs, which form the ambient-light cancellation primitive to enable visible light communication with high-reliability and robustness in bright and challenging ambient-light conditions. We also prototyped the system and performed extensive evaluations to understand the effectiveness of the system in challenging scenarios.

Interference Avoidance: Reliable and Fast Embedded VLC. To improve the limited data rate of current state-of-the-art embedded VLC, we presented Purple VLC, a fully open-source embedded VLC platform that achieves 50 kbps effective throughput at a distance of up to 6 m, 99% reliability under normal lighting conditions and 100 kbps aggregate throughput under full duplexing. This is at least 6-7x improvement over the state-of-the-art. We evaluated the system in both physical-and link-layer and provided a full Linux driver/firmware implementation of our approach. We open sourced the hardware and software of Purple platform.

8.1 Looking forward

Our work on cross-talk communication inspired various follow-up research. Some follow-up work [45, 100] proposed a better solution for wireless coexistence. Some work [38, 108, 109, 114] proposed better modulation schemes to improve system reliability. Others [29, 54, 57] focused on system performance with faster data rates. Enabling communication between various wireless technology without a gateway has invited a lot of research and commercial interest. Reliability and robustness have been the bottleneck to implement an end-to-end IoT system. We envision new benchmarking techniques to be proposed for cross-talk communication system evaluation in the future.

Wireless research is becoming more important as billions of devices are connected online with wireless interface. We envision more standardized and proprietary wireless connectivity options to come up in the next several years. With more devices connected in a limited spectrum space, we need robust and reliable technical solution to optimize the wireless connection performance. Thus, designing efficient and reliable wireless technology to utilize wireless interferences still remains an open problem. Meanwhile, exploring the unexplored spectrum, especially visible light, based on open source hardware like BeagleBone Black to provide wireless connectivity is important since visible light can exist in any environment, like a tunnel or underwater. Embedded visible light communication can be a complementary wireless technology to provide connectivity in harsh environments where RF does not perform well. Companies like Philips Lighting are integrating LiFi technology into the light bulbs. Performance issues, such as reliability and robustness, need to be improved to enable large-scale adoption and deployment of this technology. We provided initial technical work to improve reliability and robustness for embedded visible light communication, we envision better work may push the boundary and improve reliability and robustness of these systems. Imagining the flash light on smartphones as not only an illumination tool but as an wireless connectivity besides cellular, Bluetooth, and WiFi, inspires new set of future applications. We envision more work can be done for mobile computing with embedded visible light communication. The potential use for our work in this dissertation could be extended to wireless debuggers for resource-constrained systems as well as indoor-localization techniques using visible light.

Bibliography

[1] 3gpp initiative. http://www.3gpp.org/.

[2] Am335x technical reference manual. http://www.ti.com/lit/ug/spruh73o/spruh73o.pdf.

[3] Atmel AT86RF212 Datasheet. http://www.atmel.com/images/doc8168.pdf.

[4] Atmel AT86RF231 Datasheet. http://www.atmel.com/images/doc8111.pdf.

[5] Beaglebone black. http://beagleboard.org/BLACK.

[6] Bluetooth technology. https://www.bluetooth.com/.

[7] Electromagnetic spectrum. https://en.wikipedia.org/wiki/Electromagnetic_spectrum.

[8] Laser diode 5mm 650nm red. https://www.adafruit.com/product/1054.

[9] Manchester coding basics application note. http://www.atmel.com/images/atmel-9164-manchestercodingbasicsapplication-note.pdf.

[10] Photosensing with ambient background. http://www.edn.com/electronics-blogs/bakers-best/4364018/Photosensing-with-ambient-background.

[11] Pru software supported package. http://www.ti.com/tool/pru-swpkg.

[12] Prudaq repository. https://github.com/google/prudaq/wiki.

[13] Purplevlc github repository. https://github.com/gnawali/purple.

[14] Sfh213 datasheet. http://www.farnell.com/datasheets/1672048.pdf.

[15] Thread group. https://www.threadgroup.org/.

[16] Tlc2274 datasheet. http://www.ti.com/lit/ds/symlink/tlc2274.pdf.

[17] Wi-fi alliance. https://www.wi-fi.org/.

[18] Wi-spy datasheet. http://files.metageek.net/marketing/data-sheets/MetaGeek_Wi-Spy-Chanalyzer_DataSheet.pdf.

[19] Xenomai: a versatile free software framework for implementing and migrating real-time applications. https://xenomai.org.

[20] Zigbee alliance. http://www.zigbee.org/.

[21] 6lowpan. https://tools.ietf.org/html/rfc4944, Sept. 2007.

[22] N. Baccour, A. Koubaa, L. Mottola, M. A. Zuniga, H. Youssef, C. A. Boano, and M. Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(4):34, 2012.

[23] J. Baranda, P. Henarejos, and C. G. Gavrincea. An sdr implementation of a visible light communication system based on the ieee 802.15. 7 standard. In *Telecommunications (ICT), 2013 20th International Conference on*, pages 1–5. IEEE, 2013.

[24] R. Birkett and L. L. Need hard real time. Enhancing real-time capabilities with the pru. 2014.

[25] C. Boano, T. Voigt, C. Noda, K. Romer, and M. Zuniga. Jamlab: Augmenting sensornet testbeds with realistic and controlled interference generation. In *10th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 175–186, 2011.

[26] N. M. Boers, I. Nikolaidis, and P. Gburzynski. Sampling and classifying interference patterns in a wireless sensor network. *ACM Transactions on Sensor Networks (TOSN)*, 9(1):2, 2012.

[27] C.-L. Chan, H.-M. Tsai, and K. C.-J. Lin. Poli: Long-range visible light communications using polarized light intensity modulation. In *MobiSys '17*. ACM, 2017.

[28] K. Chebrolu and A. Dhekne. Esense: Communication through energy sensing. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, MobiCom '09, pages 85–96, New York, NY, USA, 2009. ACM.

[29] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu. B2w2: N-way concurrent communication for iot devices. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, SenSys '16, pages 245–258, New York, NY, USA, 2016. ACM.

[30] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. Achieving single channel, full duplex wireless communication. In *MobiCom '10*. ACM, 2010.

[31] P. Dietz, W. Yerazunis, and D. Leigh. Very low-cost sensing and communication using bidirectional leds. In *Ubicomp'03*. Springer, 2003.

[32] W. Du, J. C. Liando, and M. Li. Softlight: Adaptive visible light communication over screen-camera links. In *INFOCOM '16*. IEEE, 2016.

[33] V. L. Erickson and A. E. Cerpa. Thermovote: Participatory sensing for efficient building hvac conditioning. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '12, pages 9–16, New York, NY, USA, 2012. ACM.

[34] A. Frye, M. Goraczko, J. Liu, A. Prodhan, and K. Whitehouse. Circulo: Saving energy with just-in-time hot water recirculation. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pages 16:1–16:8, New York, NY, USA, 2013. ACM.

[35] G. Gao and K. Whitehouse. The self-programming thermostat: Optimizing setback schedules based on home occupancy patterns. In *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '09, pages 67–72, New York, NY, USA, 2009. ACM.

[36] D. Giustiniano, N. O. Tippenhauer, and S. Mangold. Low-complexity visible light networking with led-to-led communication. In *Wireless Days (WD), 2012 IFIP*, pages 1–8. IEEE, 2012.

[37] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the rf smog: making 802.11 n robust to cross-technology interference. *ACM SIGCOMM Computer Communication Review*, 41(4):170–181, 2011.

[38] X. Guo, X. Zheng, and Y. He. Wizig: Cross-technology energy communication over a noisy channel. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.

[39] H. Haas, L. Yin, Y. Wang, and C. Chen. What is lifi? *Journal of Lightwave Technology*, 2015.

[40] V. Handziski, A. Köpke, A. Willig, and A. Wolisz. Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 63–70. ACM, 2006.

[41] J.-H. Hauer, A. Willig, and A. Wolisz. Mitigating the effects of rf interference through rssi-based error recovery. In *Wireless Sensor Networks*, pages 224–239. Springer, 2010.

[42] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-Å. Norden, and P. Gunningberg. Sonic: classifying interference in 802.15. 4 sensor networks. In *Proceedings of the 12th international conference on Information processing in sensor networks*, pages 55–66. ACM, 2013.

[43] K. Hewage, A. Varshney, A. Hilmia, and T. Voigt. modbulb: A modular light bulb for visible light communication. In *VLCS '16*. ACM, 2016.

[44] M. Heydariaan, S. Yin, O. Gnawali, D. Puccinelli, and D. Giustiniano. Embedded Visible Light Communication: Link Measurements and Interpretation . In *MadCom '16*. ACM, 2016.

[45] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy. Crosszig: combating cross-technology interference in low-power wireless networks. In *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*, page 10. IEEE Press, 2016.

[46] A. Hithnawi, H. Shafagh, and S. Duquennoy. Tiim: Technology-independent interference mitigation for low-power wireless networks. In *To appear in the Proceedings of the 14th ACM International Conference on Information Processing in Sensor Networks (IPSN '15). Seattle, WA, USA*, Apr. 2015.

[47] T. W. Hnat, E. Griffiths, R. Dawson, and K. Whitehouse. Doorjamb: Unobtrusive room-level tracking of people in homes using doorway sensors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 309–322, New York, NY, USA, 2012. ACM.

[48] P. Hu, P. H. Pathak, X. Feng, H. Fu, and P. Mohapatra. Colorbars: Increasing data rate of led-to-camera communication using color shift keying. In *CoNEXT '15*. ACM, 2015.

[49] J. Huang, G. Xing, G. Zhou, and R. Zhou. Beyond co-existence: Exploiting wifi white space for zigbee performance assurance. In *Proceedings of the The 18th IEEE International Conference on Network Protocols*, ICNP '10, pages 305–314, Washington, DC, USA, 2010. IEEE Computer Society.

[50] J. Hui, D. Culler, and S. Chakrabarti. 6lowpan: Incorporating ieee 802.15. 4 into the ip architecture. *IPSO Alliance White Paper*, 3, 2009.

[51] O. D. Incel, S. Dulman, P. Jansen, and S. Mullender. Multi-channel interference measurements for wireless sensor networks. In *Proceedings of the 31st IEEE Conference on Local Computer Networks (LCN)*, pages 694–701, 2006.

[52] V. Iyer, F. Hermans, and T. Voigt. Detecting and avoiding multiple sources of interference in the 2.4 ghz spectrum. In T. Abdelzaher, N. Pereira, and E. Tovar, editors, *Wireless Sensor Networks*, volume 8965 of *Lecture Notes in Computer Science*, pages 35–51. Springer International Publishing, 2015.

[53] Y.-H. H. James Davis and H.-C. Lee. Humans perceive flicker artifacts at 500mhz. In *Scientific Report*, Feb. 2015.

[54] W. Jiang, Z. Yin, R. Liu, Z. Li, S. M. Kim, and T. He. Bluebee: a 10,000 x faster cross-technology communication via phy emulation. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, page 3. ACM, 2017.

[55] R. Jurdak, K. Klues, B. Kusy, C. Richter, K. Langendoen, and M. Brunig. Opal: a multiradio platform for high throughput wireless sensor networks. *IEEE Embedded Systems Letters*, 3(4):121–124, 2011.

[56] D. Karunatilaka, F. Zafar, V. Kalavally, and R. Parthiban. Led based indoor visible light communications: State of the art. *IEEE Communications Surveys Tutorials*, 17(3):1649–1678, 2015.

[57] S. M. Kim and T. He. Freebee: Cross-technology communication via free side-channel. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 317–330. ACM, 2015.

[58] L. Klaver and M. Zuniga. Shine: A Step Towards Distributed Multi-Hop Visible Light Communication. In *MASS '15*. IEEE, 2015.

[59] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 447–458, New York, NY, USA, 2014. ACM.

[60] B. Kusy, C. Richter, W. Hu, M. Afanasyev, R. Jurdak, M. Brunig, D. Abbott, C. Huynh, and D. Ostry. Radio diversity for reliable communication in wsns. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 270–281, April 2011.

[61] H. Lee, A. Cerpa, and P. Levis. Improving wireless simulation through noise modeling. In *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 21–30, 2007.

[62] K. S. Lee, H. Wang, and H. Weatherspoon. Phy covert channels: Can you see the idles? In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 173–185, Seattle, WA, Apr. 2014. USENIX Association.

[63] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao. Epsilon: A visible light based positioning system. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 331–343, Seattle, WA, Apr. 2014. USENIX Association.

[64] Q. Li, D. Han, O. Gnawali, P. Sommer, and B. Kusy. Demo Abstract: Twonet - Large-Scale Wireless Sensor Network Testbed with Dual-Radio Nodes . In *Proceedings of the 11th ACM Conference of Embedded Networked Sensor Systems (Sensys)*, Nov 2013.

[65] S. Li, A. Pandharipande, and F. M. J. Willems. Unidirectional visible light communication and illumination with leds. *IEEE Sensors Journal*, 16(23):8617–8626, Dec 2016.

[66] S. Li, A. Pandharipande, and F. M. J. Willems. Two-way visible light communication and illumination with leds. *IEEE Transactions on Communications*, 65(2):740–750, Feb 2017.

[67] T. Li, C. An, Z. Tian, A. T. Campbell, and X. Zhou. Human sensing using visible light communication. In *MobiCom '15*. ACM, 2015.

[68] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving wi-fi interference in low power zigbee networks. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 309–322, 2010.

[69] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Proceedings of the 12th international conference on Information processing in sensor networks*, pages 153–166. ACM, 2013.

[70] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith. Ambient backscatter: wireless communication out of thin air. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 39–50. ACM, 2013.

[71] D. Molloy. *Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux*. Wiley, 2014.

[72] C. Noda, S. Prabh, M. Alves, C. A. Boano, and T. Voigt. Quantifying the channel quality for interference-aware wireless sensor networks. *SIGBED Rev.*, 8(4):43–48, Dec. 2011.

[73] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM, 2004.

[74] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press.

[75] N. Rajagopal, P. Lazik, and A. Rowe. Visual light landmarks for mobile devices. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, IPSN '14, pages 249–260, Piscataway, NJ, USA, 2014. IEEE Press.

[76] N. Rajagopal, P. Lazik, and A. Rowe. Visual light landmarks for mobile devices. In *IPSN '14*. ACM, 2014.

[77] J. Ranjan, E. Griffiths, and K. Whitehouse. Discerning electrical and water usage by individuals in homes. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, BuildSys '14, pages 20–29, New York, NY, USA, 2014. ACM.

[78] T. Rusak and P. Levis. Burstiness and scaling in the structure of low-power wireless links. *Mobile Computing and Communications Review*, 13(1):60–64, 2009.

[79] S. Schmid, T. Bourchas, S. Mangold, and T. R. Gross. Linux light bulbs: Enabling internet protocol connectivity for light bulb networks. In *VLCS '15*. ACM, 2015.

[80] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross. Led-to-led visible light communication networks. In *MobiHoc '13*. ACM, 2013.

[81] S. M. Schmid. *Software-Defined Low-Complex Visible Light Communication Networks*. PhD thesis, ETH Zurich, 2016.

[82] M. Sha, G. Hackmann, and C. Lu. Multi-channel reliability and spectrum usage in real homes: empirical studies for home-area sensor networks. In *Proceedings of the Nineteenth International Workshop on Quality of Service (IWQoS)*, pages 39:1–39:9, 2011.

[83] Z. Shelby and C. Bormann. *6LoWPAN: the wireless embedded internet*, volume 43. John Wiley & Sons, 2011.

[84] Q. L. Shengrong Yin and O. Gnawali. Interconnecting WiFi Devices with IEEE 802.15.4 Devices without Using a Gateway. In *Proceedings of the 11th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2015)*, June 2015.

[85] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. An empirical study of low-power wireless. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):16, 2010.

[86] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. An empirical study of low-power wireless. *ACM Trans. Sen. Netw.*, 6(2):16:1–16:49, Mar. 2010.

[87] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis. The β-factor: measuring wireless link burstiness. In *Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys)*, pages 29–42, 2008.

[88] I. Tamzeed, I. Bashima, and N. Shahriar. Soundsifter: Mitigating overhearing of continuous listening devices. In *MobiSys'17*. ACM, 2017.

[89] Z. Tian, K. Wright, and X. Zhou. The darklight rises: Visible light communication in the dark. In *MobiCom '16*. ACM, 2016.

[90] Z. Tian, K. Wright, and X. Zhou. Lighting up the internet of things with darkvlc. In *ACM HotMobile '16*. ACM, 2016.

[91] D. Tsonev, S. Videv, and H. Haas. Towards a 100 gb/s visible light wireless access network. *Opt. Express*, 23(2):1627–1637, Jan 2015.

[92] A. Varshney, A. Soleiman, L. Mottola, and T. Voigt. Battery-free visible light sensing. In *VLCS '17*. ACM, 2017.

[93] J.-P. Vasseur and A. Dunkels. *Interconnecting smart objects with ip: The next internet*. Morgan Kaufmann, 2010.

[94] J. Vucic and K.-D. Langer. High-speed visible light communications: State-of-the-art. In *Optical Fiber Communication Conference*. Optical Society of America, 2012.

[95] Q. Wang. *Visible light and device-to-device communications: system analysis and implementation*. PhD thesis, 2016.

[96] Q. Wang, D. Giustiniano, and O. Gnawali. Low-cost, flexible and open platform for visible light communication networks. In *HotWireless '15*. ACM, 2015.

[97] Q. Wang, D. Giustiniano, and D. Puccinelli. Openvlc: Software-defined visible light embedded networks. In *Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems*, pages 15–20. ACM, 2014.

[98] Q. Wang, D. Giustiniano, and D. Puccinelli. An open source research platform for embedded visible light networking. *IEEE Wireless Communications*, 22(2):94–100, 2015.

[99] Q. Wang, S. Yin, O. Gnawali, and D. Giustiniano. Demo: OpenVLC1.0 Platform for Research in Visible Light Communication Networks . In *MobiCom '15*. ACM, 2015.

[100] U. Wetzker, I. Splitt, M. Zimmerling, C. A. Boano, and K. Römer. Troubleshooting wireless coexistence problems in the industrial internet of things. In *Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), 2016 IEEE Intl Conference on*, pages 98–98. IEEE, 2016.

[101] X. Xie and X. Zhang. Does full-duplex double the capacity of wireless networks? In *INFOCOM 2014*. IEEE, 2014.

[102] Y. Yang, J. Hao, and J. Luo. Ceilingtalk: Lightweight indoor broadcast through led-camera communication. *IEEE Transactions on Mobile Computing*, 2017.

[103] Z. Yang, Z. Wang, J. Zhang, C. Huang, and Q. Zhang. Wearables can afford: Light-weight indoor positioning with visible light. In *MobiSys '15*. ACM, 2015.

[104] S. Yin and O. Gnawali. Towards embedded visible light communication robust to dynamic ambient light. In *GlobeCom '16*. IEEE, 2016.

[105] S. Yin, O. Gnawali, P. Sommer, and B. Kusy. Concurrent wireless channel survey on dual band sensor network testbed. In *Mobile Adhoc and Sensor Systems (MASS), 2014 IEEE 11th International Conference on*, Oct 2014.

[106] S. Yin, O. Gnawali, P. Sommer, and B. Kusy. Multi Channel Performance of Dual Band Low Power Wireless Network. In *Proceedings of the 11th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2014)*, October 2014.

[107] S. Yin, N. Smaoui, M. Heydariaan, and O. Gnawali. Purple VLC: Accelerating Visible Light Communication in Room Area through PRU Offloading. In *EWSN '18*. ACM, 2018.

[108] Z. Yin, W. Jiang, S. M. Kim, and T. He. C-morse: Cross-technology communication with transparent morse coding. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.

[109] Z. Yu, C. Jiang, Y. He, X. Zheng, and X. Guo. Crocs: Cross-technology clock synchronization for wifi and zigbee. 2018.

[110] S. Zhai, L. Guo, X. Li, and F. X. Lin. Decelerating suspend and resume in operating systems. In *HotMobile '17*. ACM, 2017.

[111] C. Zhang and X. Zhang. Litell: Robust indoor localization using unmodified light fixtures. In *MobiCom '16*. ACM, 2016.

[112] P. Zhang, P. Hu, V. Pasikanti, and D. Ganesan. Ekhonet: High speed ultra low-power backscatter for next generation sensors. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 557–568, New York, NY, USA, 2014. ACM.

[113] Y. Zhao and J. Vongkulbhisal. Design of visible light communication receiver for on-off keying modulation by adaptive minimum-voltage cancelation. *Engineering Journal*, 17(4), 2013.

[114] X. Zheng, Y. He, and X. Guo. Stripcomm: Interference-resilient crosstechnology communication in coexisting environments. In *Proceedings of ACM INFOCOM*, 2018.

[115] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma. Zifi: wireless lan discovery via zigbee interference signatures. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 49–60. ACM, 2010.