Raptor-Coded Free-Space Optical Communication Experiment

RAPTOR-CODED FREE-SPACE OPTICAL COMMUNICATION EXPERIMENT

BY

MIN LU, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2015) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Raptor-Coded Free-Space Optical Communication Experiment
AUTHOR:	Min Lu B.Sc., (Electrical Engineering) Zhejiang University, Hangzhou, China
SUPERVISOR:	Dr. Steve Hranilovic
NUMBER OF PAGES:	xii, 109

To my family

Abstract

Free-space optical (FSO) communications are attractive in a host of applications, such as last-mile connections with backbone networks, due to their high data rates, license-free operation, low interference and ease-of-deployment. However, FSO communication systems suffer from availability issues mostly caused by bad weather conditions, thus preventing the widely deployment of FSO communication technology.

Raptor codes are a class of fountain codes which adapt to varying channel conditions. The rateless property of Raptor codes makes them ideal for channels with high vulnerability such as FSO channels. This work presents a performance evaluation of two classes of Raptor codes in FSO communications: Raptor 10 and RaptorQ codes. Both of the Raptor codes are capable of improving the reliability of FSO communications and adapting FSO data rates regardless of channel conditions. Compared with the Raptor 10, RaptorQ codes are capable of offering more robust communication performance.

This work presents the software implementation of soft-switching hybrid FSO/LAN link using a Raptor 10 code. Preliminary experimental measurements show that the hybrid link could exploit the bandwidth of both links and avoids availability issues caused by harsh weather conditions.

This work also includes an FPGA-based hardware implementation of a Raptor 10 coded FSO link with focus on the receiver architecture. A complete hardware architecture of Raptor 10 decoder is presented in this work. The hardware structure of a pipelined receiver is proposed and evaluated, showing the important role of Raptor codes in FSO communications. This work serves as the first step towards the FPGA-based hardware implementation of soft-switching hybrid FSO/RF link using Raptor codes.

Acknowledgements

During my two-year graduate study in Free-Space Optical Algorithms Laboratory at McMaster University, I learned a lot both in life and research from my supervisor Dr. Steve Hranilovic. He is always there to offer productive suggestions whenever I have trouble in my study. He is always there to encourage me whenever I feel lost in my research work. I owe my deepest acknowledge to my supervisor for his invaluable support and guidance.

I also wish to thank Mr. Ayman Mostafa and Miss. Linyan Liu and Dr. Mohamed A. El-Shimy who are senior members in my research group. Their sincere support help me solve a lot of problems regarding programming and simulation that I had during my research. I also want to thank all other group members: Mohammed S.A. Mossaad, Dingchen Zhang, Ahmed Darwish, Eric Edwards and Boxiao Han.

I would also like to thank administrative and technical staff in our department, especially Cheryl Gies, Kerri Hastings, Ron Harwood and Dave Acciaroli, for their consistent and thoughtful assistance.

And last, but no least, I would like to thank my parents for selfless love and endless support.

Contents

List of Figures

Chapter 1

Introduction

1.1 Free-Space Optical Communications

Free-space optical (FSO) communications utilizes light as the information carrier and air as the transmission medium [7]. The first experimental demonstration of the working principle behind FSO communications can be traced back to 1880 when Alexander Graham Bell and Charles Sumner Tainter successfully transmitted a voice signal around 200 meters using sunlight [8]. The introduction of laser technology in the 1960s boosted the development of FSO communications [9]. Considerable attention from both industrial and academic communities has been attracted to FSO communication technology. Numerous research papers are devoted to the study of the characteristics of FSO communications as well as potential applications. Commercial FSO communication systems are currently available from several companies including BridgeWave [10], fSONA [11] and LightPointe [12]. The market for FSO communications is currently worth around 116.7 million USD and is estimated to reach 940.2 million USD by 2020 [13].

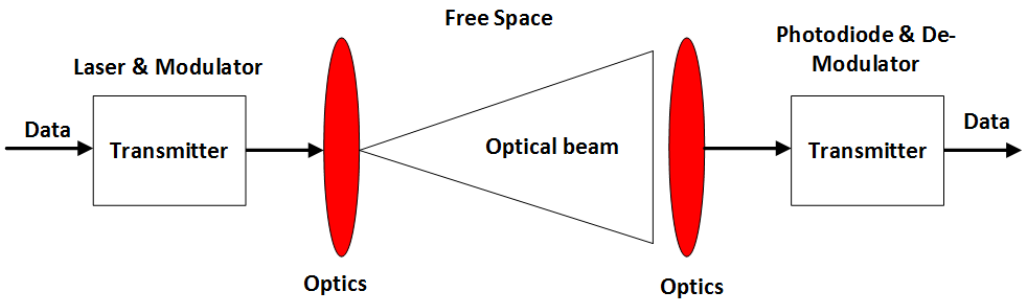

Figure 1.1: Diagram of FSO communication system adopting intensity modulation with direct detection (IM/DD)

Currently a majority of FSO communication systems adopt intensity modulation with direct detection (IM/DD) as the modulation method. Because implementation of phase or frequency modulation as the modulation method for FSO communications still remains a technical challenge [14]. Fig. 1.1 presents a typical IM/DD system. At the transmitter side, data are modulated with on-off keying (OOK) modulation and output to a laser diode or LED. The laser diode or LED converts the electrical signal to an optical intensity signal which propagates through the atmosphere. At the receiver, data are recovered through demodulation of the electrical current signal which is obtained from intensity signal via photodiode.

Using light intensities as the information carrier and the atmosphere as the transmission medium gives FSO communications many advantages. FSO communications are capable of offering high-bandwidth communications. Commercially available free-space optical communication systems are capable of offering data rates ranging from 100 Mbps to 10 Gbps with link distances within several kilometers [15]. For example, the NS10G CASCADE [16] from Northern Storm could support 10 gigabit ethernet with link distance up to 1500 meters under clear weather. Experimental demonstration of an approximately 200 meters

free-space optical link operating at the speed of 160 Gb/s is available in [17]. FSO communications typically uses electromagnetic wavelengths from 800 nm to 1550 nm [18]. Compared with most microwave communication, FSO communications are attractive since no spectrum license is required to operate a FSO link. Fiber deployment in urban areas is usually costly and disruptive due to digging tunnels and obtaining rights-of-way [19]. For example, construction of a FSO network with a total distance of 12.8 km in Istanbul's business district only costs around $1.4 million, much lower than the estimated cost of fiber deployment around $78.5 million [20]. FSO communication systems are easy to deploy. A typical FSO communication system is mainly compromised of two identical optical transceivers. Full duplex broadband communication can be achieved as soon as the optical transceivers are successfully deployed. Great flexibility of deployment makes FSO communications perfect for temporary network construction. Considering the fact that FSO communications are immune to electromagnetic interference, it can offer highly secure communications, thus FSO communications are suitable for applications concerning security-sensitive communication such as military communications [21, 22].

FSO communication systems can be applied in a wide range of scenarios such as satellite-to-ground [23], UAV-to-ground [24], ground-to-ground [25], satellite-to-satellite [26] communications, as shown in Fig. 1.2. The "last mile" problem deals with how to connect local users to fiber backbone efficiently and effectively [27]. It still remains a challenging and unresolved issue in network construction. FSO communications offer a flexible and cost-efficient solution to this problem [28]. Fig. 1.3 presents the FSO technology-based solution to the "last mile" problem. Building 1 is

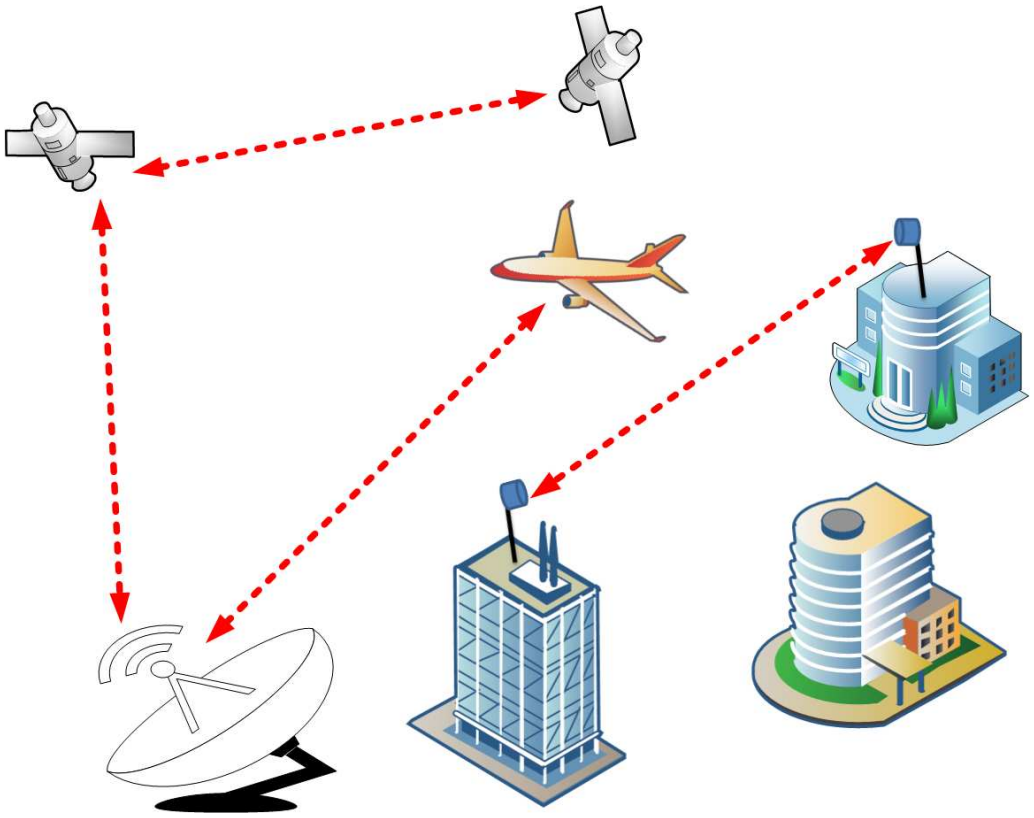

Figure 1.2: Applications of FSO communication

directly connected to the wide area network using optical fiber while Building 2 has indirect access wide area network via FSO link between Building 1 and Building 2. Using FSO communication technology, local area networks can be easily integrated to wide-area networks.

Since FSO communications transmit modulated laser beams through the atmosphere, they are impacted by environmental factors such as scattering, absorption and atmospheric turbulence. Light signals can be easily attenuated and distorted by atmospheric particles such as fog, rain or snow, whose influences on FSO communications have been intensively studied in [29–31]. Fog causes optical attenuation up to 500 dB/km in the 950 and 850 nm wavelength [32]. Unlike fiber

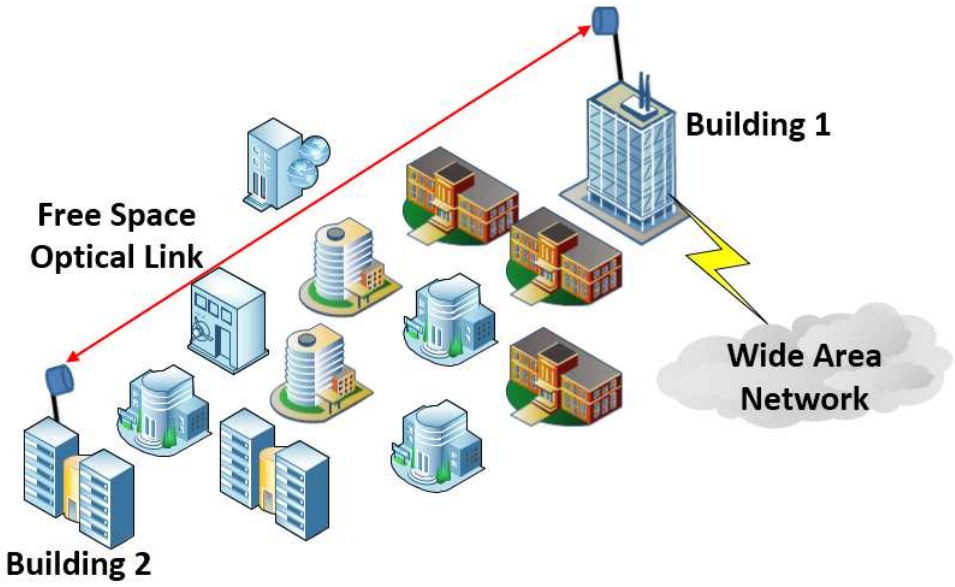

Figure 1.3: Solution to "last mile" problem by employing FSO link

optics, the atmosphere is very unstable. The refractive index along the transmission path does not stay constant due to unstable temperature and pressure , which is commonly known as atmospheric turbulence [33]. Pointing errors [34] also degrade the performance of FSO communications. Although the advantages of FSO communication are very attractive, its widespread deployment cannot be anticipated before solving the availability issue. The availability of the FSO communication systems is limited both by the link distance and weather conditions. Link availability analysis in [35] shows that it is difficult for FSO communication system to achieve carrier-class availability(99.999%) over a distance over 210 meters.

1.2 Raptor Codes

Fountain codes were introduced as a solution to multicast and broadcast data transmission in 1998 [36]. Fountain codes are of great importance for packet erasure channels, i.e, computer networks [37]. Given a source block of k source symbols, the encoder of the fountain codes generates potentially limitless encoding symbols, behaving similarly to a fountain. The capability of generating potentially limitless encoding symbols is commonly referred as the rateless property. The rateless property makes fountain codes capable of tracking channel change and adapting data rate accordingly without knowing channel states. The decoder of fountain code can be envisioned as a bucket that collects water drops from the fountain. As long as enough encoding symbols are received, the code is capable of recovering the source block correctly.

Luby Transform (LT) codes [38] were invented in 2002 as the first efficient and practical class of fountain codes. Although the invention of LT codes is quite innovative, the performance of LT codes is limited since it is difficult to obtain encoding and decoding algorithms with linear time complexities [39].

As a member of the rateless fountain family, Raptor codes [3] are ideal for channels with unknown erasure probability [40]. The computation complexities of encoding and decoding algorithms of Raptor codes are linearly achievable by combining pre-coding techniques with LT codes. There are currently two popular commercially available versions of Raptor codes, Raptor 10 code [41] and RaptorQ code [42]. Raptor 10 codes were invented in late 2000 and have been already adopted into many standards including 3GPP MBMS [43] and DVB [44]. RaptorQ code is currently commercially available from Qualcomm [45]. RaptorQ code was

invented as an improvement on the Raptor 10 code in terms of computation complexity and supporting larger source block. Using a software development kit from Qualcomm targeting all popular processors and operating systems, RaptorQ codes can be easily integrated into potential applications to ensure reliable and efficient data transmission [46].

There are many research papers available discussing features and possible applications of Raptor codes. Communications using Raptor codes under different channel models such as Guassian channels and binary memoryless symmetric channels have been well studied in [39, 47–49]. In [50], two popular fountain codes, Raptor codes and LT codes, are implemented and compared with each other in terms of overhead and computational complexity. In [4], the decoder of a Raptor 10 code is implemented with a soft-core embedded processor on Altera FPGA and different factors such as number of source symbols, symbol size are discussed in details. Later work [51] presents the software-based implementation of RaptorQ code on Altera Stratix FPGA. The performance investigation of multicast data service using RaptorQ code and Raptor 10 code is conducted in [52].

1.3 Raptor Coded FSO communication

The rateless property of Raptor codes makes them ideal for FSO communications which are unreliable and posses volatile channel conditions. Regardless of the variation of channel state, Raptor-coded FSO communications are capable of following the change of channel state, thus maximizing channel utilization. Raptor codes are utilized to mitigate the negative influence of the misalignment error of a mobile FSO link between the ground station and the unmanned aerial vehicle [53].

A simulation study of Raptor codes using a slant path FSO channel model is conducted in [54]. A simulation study of various LT and Raptor codes in simulated terrestrial turbulence is presented in [55]. In [56], the performance of Raptor codes for FSO channels is demonstrated using channel measurements from an FSO link and offline decoding. This work is extended in [57], where a systematic Raptor code is used to implement a rate-adaptive FSO link and performance of the proposed communication system is evaluated by offline processing using FSO channel measurements. Performance investigation of RaptorQ-coded FSO communication is conducted targeting an indoor free-space optical link with turbulence generated by heating equipments in [58]. A performance evaluation of various rateless codes including LT code, Raptor 10 code and RaptorQ code is conducted toward a FSO link between the satellite and the ground station in [59].

1.4 Hybrid FSO/ RF Links

Despite its many advantages such as high data throughout, ease-of-deployment, FSO communication suffers severely from availability issues. The reliability of FSO communication cannot always be guaranteed especially under adverse weather conditions. Raptor-coded FSO communication is helpful to mitigate the negative effects caused by unstable FSO link. However, severe weather conditions such as dense fog or heavy snow still can easily render FSO communications unavailable. One of many promising solutions to improve the availability is to deploy a hybrid link which compromises a free-space optical link and another reliable link usually a radio frequency (RF) link. The secondary link is required to operate normally under adverse weather conditions such as snow or fog. The radio frequency (RF) link can

be replaced with other links such as the fiber link. Atmospheric scintillation and scattering caused by fog could easily interrupt FSO communications while rain is an impairment for microwave links [60]. Because the size of rain droplets ranges from 1000 to 10,000 μm and the size of fog typically ranges from 1 to 20 μm [35]. Hybrid FSO/RF links could achieve the high throughput of the FSO links for most of the time and obtain improved availability from RF link. Depending on the configuration of FSO link and RF, this solution can be classified into two categories: hard-switching techniques and soft-switching techniques.

Works in [28, 35, 61–65] and commercially available hybrid products [15] mostly adopt hard-switching techniques as shown in Fig. 1.4. The basic idea behind hard-switching techniques are that data transmission switches between the FSO link and the secondary link using trigger signals which depend on weather conditions. If weather is favorable, data is transmitted through the FSO link which is capable of offering high data throughput. When the weather conditions become harsh such as snow or dense fog, a trigger signal is generated and data transmission switches to the other link to obtain continuous communication. In [63], the received optical signal level is utilized as trigger signal to switch data transmission between FSO and RF links. In [62], the performance of the hybrid FSO/RF link is evaluated under different kinds of weather conditions including fog, rain and dry snow. Simulation study in [65] analyzes the communication performance of a hard-switching hybrid FSO/RF link. Different kinds of hard-switching algorithms are considered and evaluated for the hybrid FSO/RF link in [61]. However, hard-switching techniques are not efficient in the sense that bandwidth of the RF link is not fully exploited under good weather conditions.

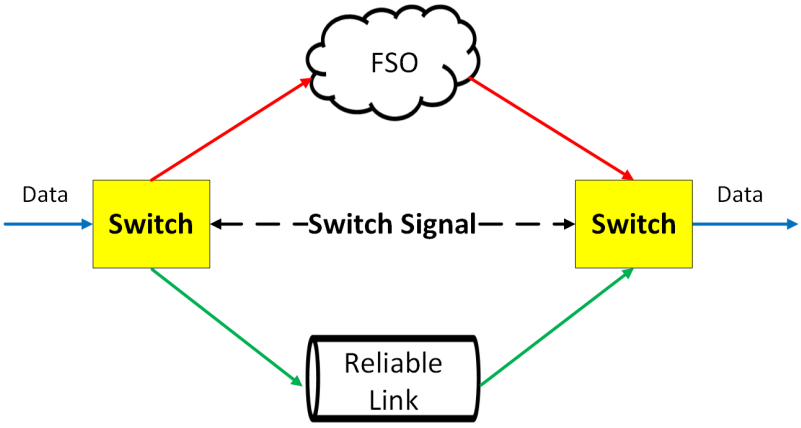

Figure 1.4: Hard-switching configuration for Hybrid FSO/RF links

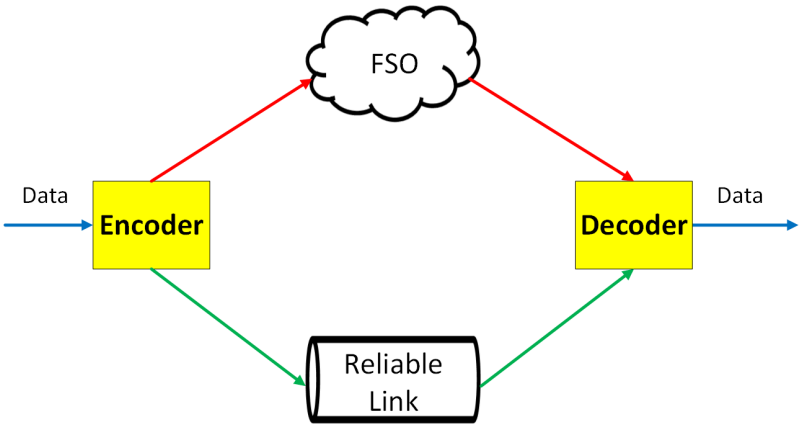

Figure 1.5: Soft-switching configuration for Hybrid FSO/ RF links

Motivated by improving communication efficiency, soft-switching techniques are proposed, as shown in Fig. 1.5. Soft-switching techniques utilize channel coding techniques to transmit data over two links concurrently instead of switching between two links according to weather conditions. Raptor codes [3] have been applied to enable soft-switching in hybrid FSO/RF recently due to its rateless property. In [66], a short length Raptor code is designed to encode symbols for both RF and FSO links. The decoder collects packets from both links and decodes when sufficient symbols are received. A simulation study was conducted using a 1Gbps FSO link and a 96Mbps WiMax RF link using climate data, however, no experimental verification in an FSO channel was performed. Simulation results in [67] using an off-the shelf Raptor code and classical channel models demonstrated that such hybrid FSO/RF soft-switching techniques are capable of approaching information theoretic limits in a variety of weather conditions.

1.5 Contributions of the Thesis

As an extension of [57], two versions of Raptor codes, Raptor 10 code [41] and the RaptorQ code [42], are considered for application in FSO communications. As guidance for future implementation, standard ethernet packets with User Datagram Protocol (UDP) employed at the transport layer are employed in this work, which differs from ideal theoretic packet used in [57]. Also a performance evaluation of Raptor codes is conducted targeting a 1.87 km FSO link deployed across McMaster University in different weather conditions. Instead of using mathematical models, the effects of the channel are considered using channel measurements from an experimental link [6]. Two conclusions can be obtained through the simulation

study. Firstly, Raptor-coded FSO communications help mitigate erasure influence of the FSO link and is capable of adapting rate regardless of channel conditions. Secondly, RaptorQ codes are capable of offering more robust communication performance compared with Raptor 10 code.

Earlier works regarding hybrid FSO/RF such as [66] and [68] mostly concentrate on mathematical analysis or numerical simulation using off-line channel information other than in-field demonstration over real-life hybrid links. This work presents an in-field experimental study of a Raptor-coded hybrid FSO/RF link. Due to limited experimental resources, a LAN link is used to represent the reliable RF link. In this work, a practical soft-switching hybrid FSO/LAN link employing Raptor 10 codes [41] is proposed and implemented in software. Link evaluation is conducted targeting a hybrid link consisting of a 1.87 km FSO link deployed across McMaster University and a 100 Mpbs campus network link. Though many improvements such as real-time data processing and feedback are still required, preliminary results show that Raptor-coded hybrid FSO/RF link is a viable solution to the availability issue of FSO communications.

Field-programmable gate arrays (FPGAs) are well known for their parallel processing ability. This work presents an FPGA-based hardware implementation of Raptor-coded FSO links. The most difficult tasks relate to the establishment of UDP communication and the implementation of a Raptor 10 decoder. UDP communications are established with the help of commercially available Quixtream Gigabit Ethernet Protocol [69] provided by TEK Microsystems, Inc. The implementation of the Raptor 10 decoder takes advantage of the matrix inversion algorithm proposed in [70] which balances the algorithms proposed in [71] and [72]

in terms of hardware usage and time complexity. Also a pipelined receiver architecture is designed to maximize link throughput at the application layer. To the best knowledge of the author, this is the first experimental investigation of FPGA-based implementation of Raptor-Coded FSO link [73].

1.6 Thesis Structure

The thesis is organized as follows: Details of the experimental setup are presented in Chapter 2. Chapter 3 presents performance evaluation of Raptor codes in FSO communication. Chapter 4, an experimental study of Raptor coded soft-switching hybrid FSO/RF communication system is given. Chapter 5 presents works regarding hardware implementation of Raptor-Coded FSO link. The thesis concludes in Chapter 6 with direction for future work.

Chapter 2

Free-Space Optical Experimental Link

In this chapter, a brief introduction of the free-space optical link deployed across McMaster University is presented. More details of the free-space optical link can be found in [6]. Following is an experimental setup closely related to ongoing experiments conducted in this work. Note one major difference between the experimental setup in [6] and this work is the operating mode of the optical transceiver. In [6], the analog mode of the link is utilized while this work focuses on the digital mode.

2.1 SONAbeam 1250-M Optical Transceiver

The ongoing experiments take advantage of a customized SONAbeam 1250-M [74] link which is deployed across McMaster University and Westdale Village in west Hamilton area, as shown in Fig. 2.1. The free-space optical link operates at wavelength of 1550 nm [75]. The link is of great importance for both the theoretical study of FSO

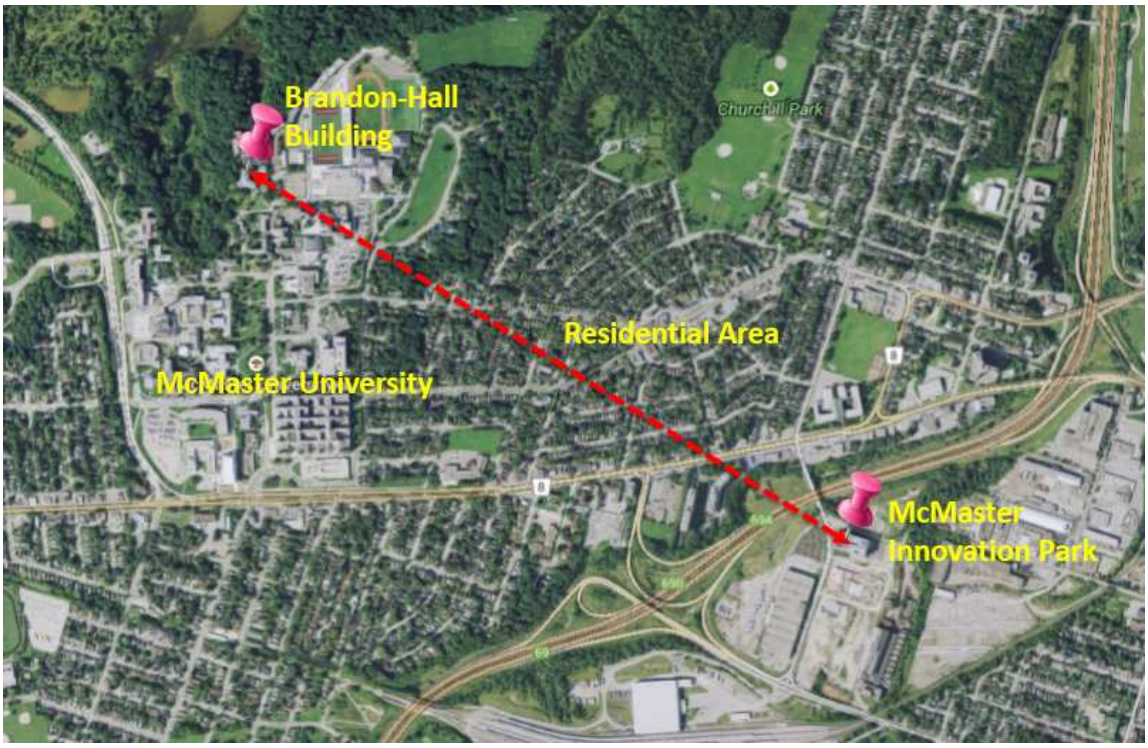

Figure 2.1: Snapshot of optical terminal installation sites, April 2014 (taken by Google Map ©2014 [1])

channels and experimental demonstration of FSO communication algorithms. Key components of the deployed FSO link are two identical full-duplex optical transceivers which are installed on the rooftops of McMaster Innovation Park (MIP) and Brandon-Hall (BH). The two installation locations are approximately 1.87 km distant from each other, as shown in Fig. 2.2. The optical transceiver can be configured to operate under either analog or digital mode according to application requirements.

As shown in Fig. 2.3, each optical transceiver consists of four laser transmitters including one analog transmitter, two standard digital transmitters and one custom digital transmitter and one dual-mode receiver. The operation of the optical transceiver is controlled both by Power & Control Assembly (PCA) and the

Figure 2.2: Pictures of site views from Brandon-Hall and McMaster Innovation Park of the experimental FSO links

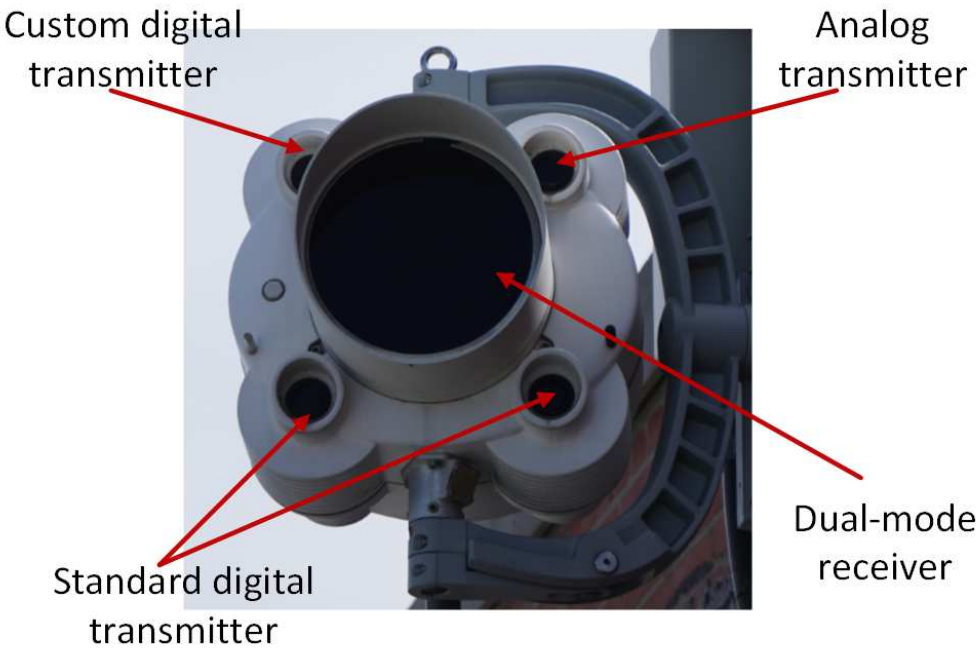

Figure 2.3: Customized SONAbeam-1250-M Optical Transceiver

SONAbeam Terminal Controller (STC) software. The PCA provides power conditioning for optical transceiver and serves as an interface between user hardware and the optical transceiver [76]. The STC software is provided by fSONA Networks Corporation to monitor and control the operation of the optical transceiver. Digital components of optical transceiver are briefly introduced in the following sections and more details can be found in [76] and [6].

2.1.1 Standard Digital Transmitters

Each optical transceiver is equipped with two standard digital transmitters. A standard digital transmitter is capable of offering data rates ranging from 100 Mbps to 1.6 Gbps. A standard digital transmitter accepts digital inputs through the standard interface, which is a 1310 nm SM fiber with SC connector. It supports a

Table 2.1: Laser power control characteristics [76]

Power level	Laser power (mW)
0	0.00
1	4.10
2	6.90
3	11.00
4	17.30
5	27.40
6	41.70
7	59.80

wide range of transmission standards including Gigabit Ethernet and Fiber Channel. The output power of a standard digital transmitter can be set as one of seven power levels listed in Table 2.1 via STC. There are two operating modes available for each standard digital transmitter, commercial mode and bypass mode [76]. Under commercial mode, transmitter would regenerate, reclock and then transmit data bits taken from standard interfaces. Under bypass mode, the transmitter simply regenerates and transmits binary data without reclocking.

2.1.2 Custom Digital Transmitter

The custom digital transmitter can either be configured to work in commercial mode or custom mode. The custom digital transmitter functions exactly as a standard digital transmitter under commercial mode. Under custom mode, users can drive the digital transmitter with an external data source. It supports the same data rates and uses the same power control as a standard digital transmitter.

2.1.3 Dual-Mode Receiver

Depending on the operating mode of the link, the dual-mode receiver can be configured as digital mode or analog mode. The dual-mode receiver in this work is configured to operate under digital mode to support binary data communication. The speed of the link ranges from 100 Mbps to 1.6 Gbps. The dual-mode receiver uses a single InGaAs APD photo-diode and its sensitivity is 2 μW. Same as the standard digital transmitter, the interface to the dual-mode receiver is 1310 nm SM fiber with SC connector.

2.2 Experimental Details

Ongoing experiments presented in Chapter 4 and Chapter 5 take advantage of the same free-space optical link in [6]. The two SONAbeam optical transceivers deployed at the roofs of McMaster Innovation Park (MIP) and Brandon Hall(BH) are configured to work under digital mode in this work. A block diagram of the experimental setup is shown in Fig. 2.4. The indoor setups at both locations are very similar and only the one at MIP is presented in Fig. 2.5. The experiment in Chapter 4 utilizes the PCs as the communication nodes while the experiment in Chapter 5 employs the Triton-V5 VXS boards as the communication nodes. An brief introduction to the PCs can be found in Section 2.3 and the description of the Triton-V5 VXS boards can be found in Section 2.4.

Each optical transceiver is connected to a PC via a serial RS232 port (DB9 connector) which enables the SONAbeam transceiver controlling software to adjust and monitor the operation of the optical transceiver. As shown in Fig. 2.6, it is

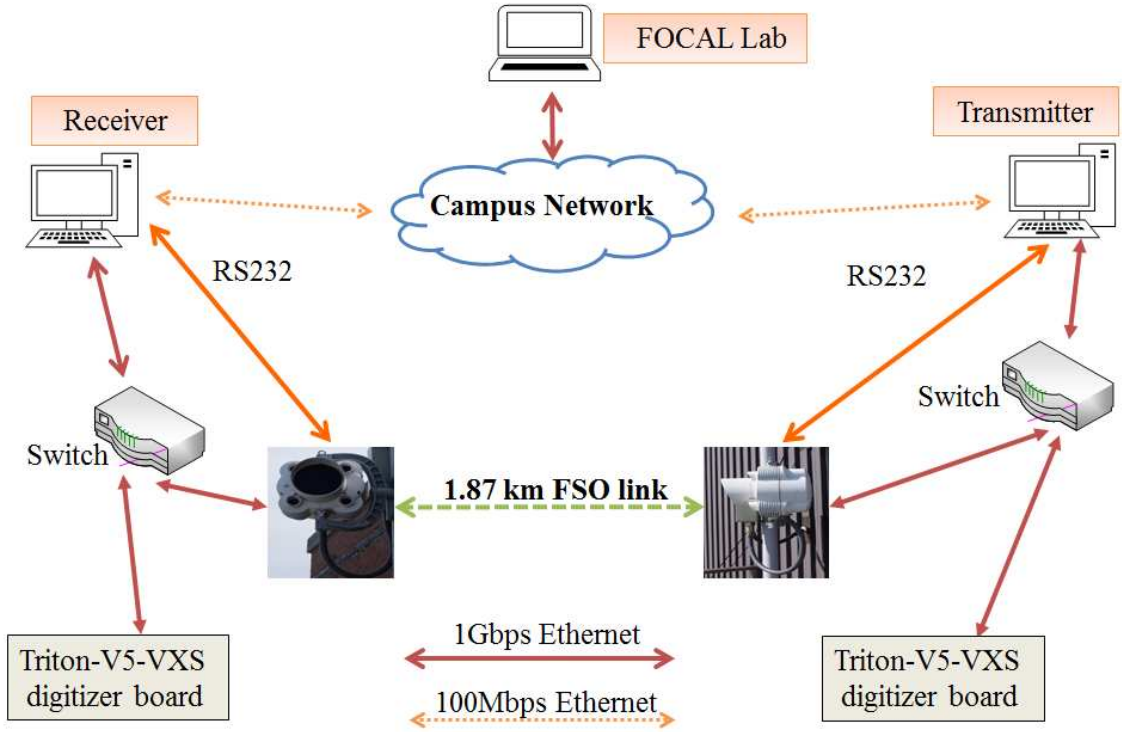

Figure 2.4: Block diagram of the experimental setup for free-space optical communications

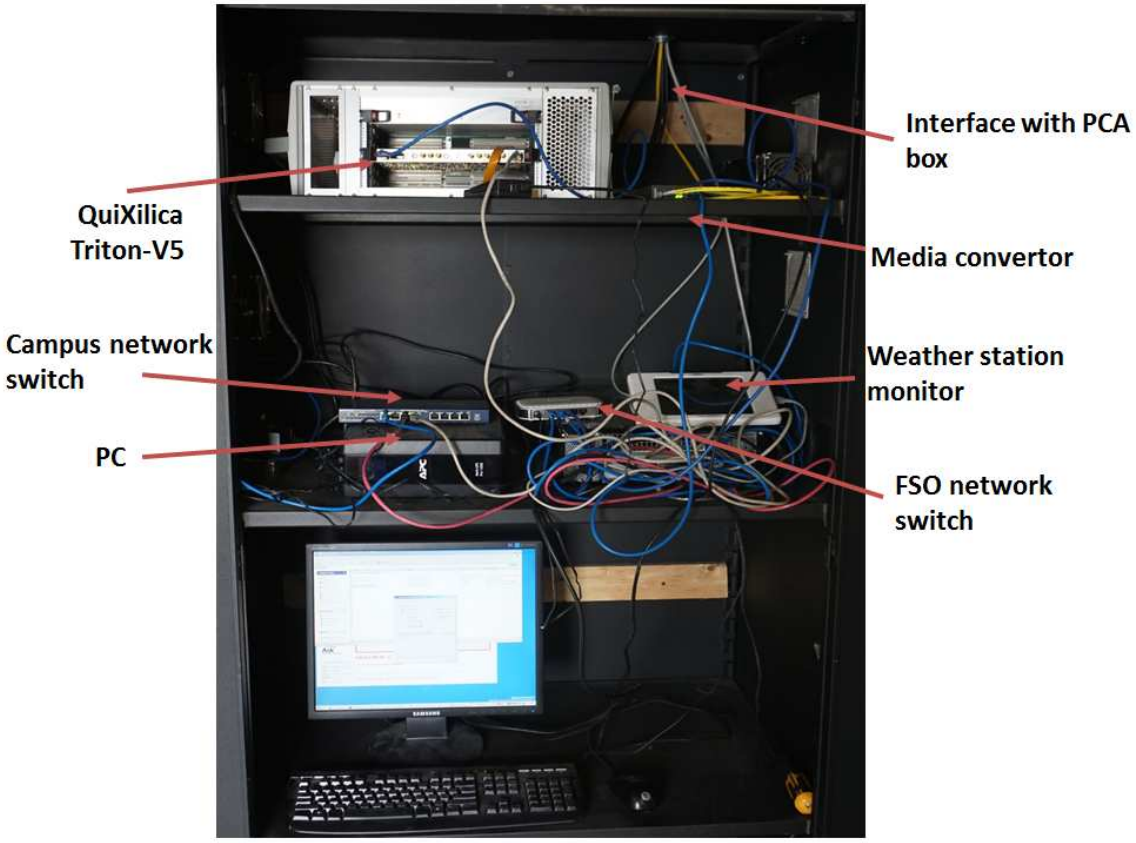

Figure 2.5: Indoor experimental setup at McMaster Innovation Park

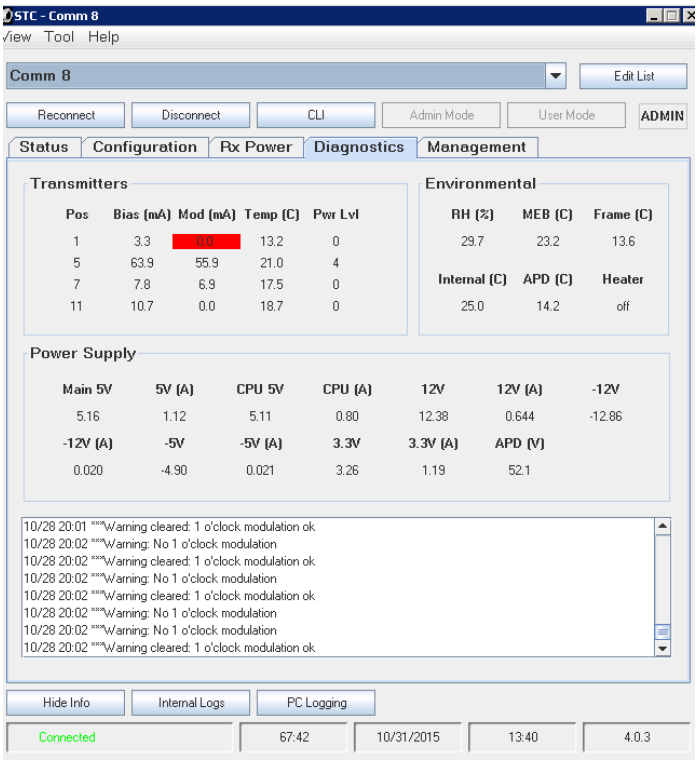

Figure 2.6: Details of terminal operating status

convenient to obtain comprehensive information regarding the operating status of the optical transceiver through transceiver controlling software. Such information includes transmitter power level, power supply and environmental temperature. Note this information is very important for maintaining and troubleshooting operation of the optical transceiver.

Also each optical transceiver is connected to a gigabit ethernet switch which includes multiple communication nodes such as PCs and FPGAs inside a local area network established by the free-space optical link. A gigabit SFP Media Converter [77] is used to solve the incompatibility of interfaces between 1310 nm MM fiber (interface to optical transceiver) and RJ45 (interface to gigabit ethernet switch). Each PC is equipped with two gigabit ethernet cards. One gigabit ethernet card is connected to the 100 Mbps campus network while the other one is connected to the gigabit ethernet switch. Additionally, each gigabit ethernet switch also connects one QuiXilica Triton-V5 VXS high-speed digitizer board. Two local area networks have been constructed with such a configuration. One local area network is established by the FSO link while the other one is established by campus network (fiber link). Each component of the communication could be remotely controlled via the PC located at the FOCAL Lab in the ITB building, McMaster University. The experimental setup here not only meets the requirements of Raptor-coded FSO communication link but also can be used for long term development of Raptor-coded hybrid FSO/RF communication system. In this work, communication nodes such PCs and FPGAs at MIP are chosen as transmitters while corresponding communication nodes at BH function as receivers.

For the ongoing experiments in Chapter 4 and Chapter 5, the configuration of

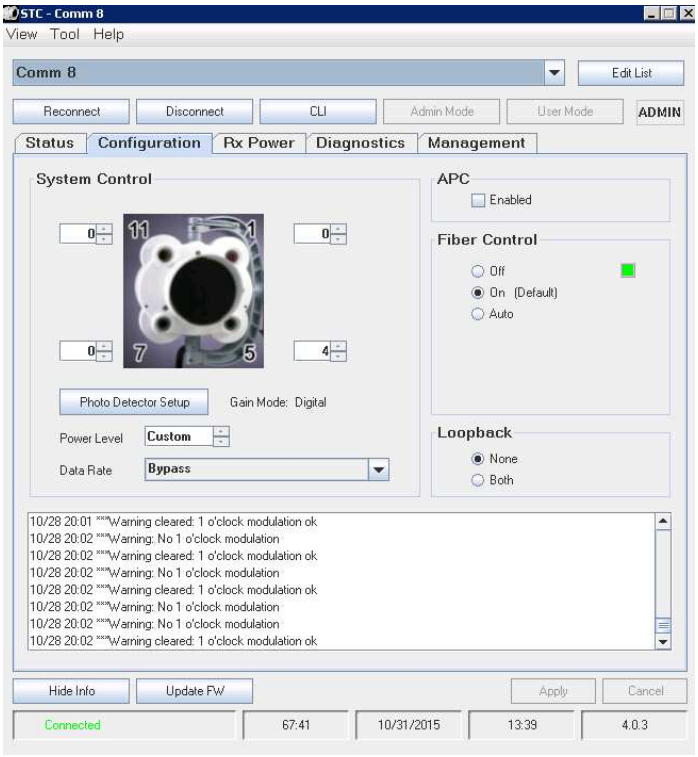

Figure 2.7: Optical transceiver configuration through STC

the optical transceiver is configured as shown in Fig. 2.7. Data rate is configured as "bypass" which indicates data rate is dominated the rate of digital input. Since the optical link functions under digital mode, the photo detector is configured in digital mode (auto gain).

2.3 Computer Resources

The CPUs of conventional computers are used as processing cores in the experiments conducted in Chapter 4. Network adapters of each computer are the interfaces connected to the physical layers of the FSO network and campus network. The two personal computers deployed at MIP and BH share the same hardware and

Table 2.2: Specifications for Personal Computer

CPU	Intel Core2 E8400 @ 3.00GHz
Memory	100Mbps to 1.6Gbps, software selectable
Memory	4GB (Kingston DDR2 800 MHz)
Hard Disk	Seagate ST3320613 AS (320GB/ 7200rpm)
Operating system	Windows XP Professional 32 bit SP3
Motherboard	Intel DQ45CB (Intel 4 Series Chip set- ICH10DO)
Network Adapter	Intel 82567LM-3 Gigabit Network Adapter, VIA VT3119 Velocity Family Gigabit Ethernet Adapter

software configurations. The features of the computers used are summarized in Table 2.2.

2.4 Triton-V5 VXS

QuiXilica Triton-V5 VXS high-speed digitizer board is utilized in Chapter 5 to implement the Raptor 10 decoder and establish UDP communications. Key components on the Triton-V5 VXS board are a 10-bit ADC (maximum speed 2.2 Gigasamples per second), a 12-bit DAC (maximum speed 4 Gigasamples per second) and three high-grade Xilinx Virtex-5 series FPGAs [2]. Fig. 2.8 presents a block diagram of the resource distribution on the Triton-V5 VXS board. Supported with high-end components, the Triton-V5 VXS board is designed for applications such as unmanned airborne and radar. Hardware features and specifications of the Triton-V5 board are summarized in Table 2.3. An overview of FPGA resources available on the Triton-V5 VXS digitizer board are shown in Table 2.4. Featured with three interconnected Xilinx Virtex-5 series FPGAs, the Triton-V5 VXS digitizer board is capable of advanced data processing. Two SFP+ ports on the

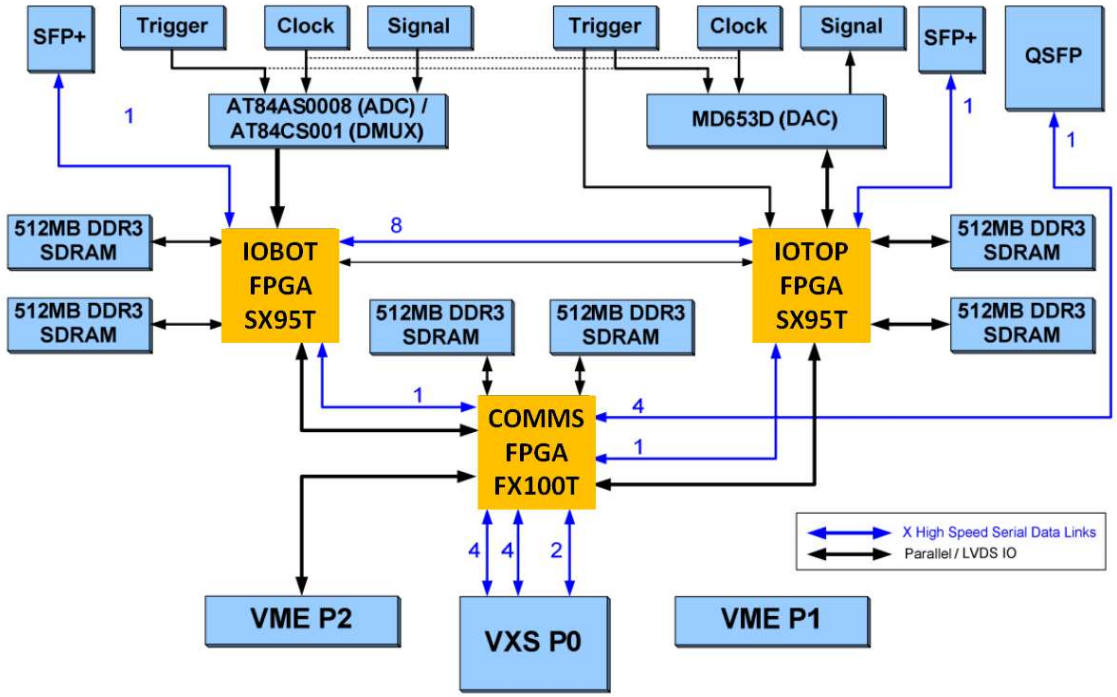

Figure 2.8: Block diagram of resources distribution on Triton-V5 board (modified based on [2])

front panel of the digitizer board facilitate standard communication such as gigabit ethernet and fiber channel [2]. The Triton-V5 VXS board also possesses 3 GB memory evenly distributed among six DDR3 SDRAMs. Sufficient memory resources are useful for applications involving data collection, such as channel measurement in [6].

Table 2.3: Hardware features and specifications of Triton-V5 board [78]

FPGAs	2× XCVSX95 (FPGAs 0, 2) 1× XC5VFX100 (FPGA 1)
FPGAs features (three combined)	1536 DSP slices (25 × 18 multipliers) 4280 integrated block RAMs (36 kbit) 2 PowerPC 440 RISC cores up to 550 MHz clock
Serial I/O	2× SPF+ (up to 3.75 Gbps) 1× QSPF (up to 4×3.75 Gbps
ADC	10-bits, 0.49 mV LSB resolution ±250 mV input signal level up to 2.2 GSa/s sampling rate
DAC	12-bits, 0.195 mV LSB resolution ±400 mV output signal level up to 4 GSa/s sampling rate
Memory	6×512-MB DDR3 SDRAM 64-bits data bus up to 400 MHz clock (DDR)
Clock sources	100, 125, 156.25, 212.5 MHz derivative of the external ADC & DAC sampling clock

Table 2.4: Main resources on Virtex-5 XC5VFX100 and XC5VSX95T FPGA [79]

Device	Configurable Logic Blocks (CLBs)			Block RAM Blocks			PowerPC Processor Blocks
	Array Row × Col	Virtex-5 Slices	Max Distributed RAM (Kb)	18 Kb	36 Kb	MAX Kb	
XC5V SX95T	160 × 46	14,720	1,520	488	244	8,784	N/A
XC5V FX100T	160 × 56	16,000	1,240	456	228	8,208	2

2.5 Summary

This chapter summarized the experimental setup of experiments carried out in Chapters 4 and 5. The experimental setup here can be used for both software and hardware implementation of Raptor-coded FSO link or Raptor-coded hybrid FSO/RF link.

Chapter 3

Raptor Codes for Free Space Optical Communication

In this chapter, a brief introduction to Raptor codes will be given first. A description of encoding and decoding algorithms of two versions of Raptor codes, Raptor 10 and RaptorQ codes are presented. Performance evaluation of Raptor 10 and RaptorQ codes in FSO communications is simulated using the FSO link introduced in Chapter 2. Instead of using mathematical channel models, simulations in this chapter are carried out using channel measurements recorded over the FSO link in different weather conditions.

3.1 Raptor Code Technology

3.1.1 Background

An introduction to fountain codes, LT codes and Raptor codes can be found in Section 1.2 of Chapter 1. Raptor codes can be considered as the improved version of LT codes. LT codes are usually characterized with the number of source symbols k and a degree distribution $\Omega(d)$ with $1 \leq d \leq k$. Degree distribution $\Omega(x)$ is defined as the probability that an encoding symbol is generated as the sum of x randomly and uniformly chosen source symbols [38]. Good design of the degree distribution is well-studied in [38]. Raptor codes are a concatenation of a high-rate pre-coder followed by a Luby transform (LT) coder. Raptor codes are usually defined by three parameters: the number of source symbols k, the pre-code C and a degree distribution $\Omega(x)$ [3]. As shown in Fig. 3.1, source symbols are first pre-coded into intermediate symbols. Encoding symbols are generated using intermediate symbols as new source source symbols according to the degree distribution $\Omega(x)$ of LT codes. Details of coding theory regarding Raptor codes can be found in [3]. This chapter considers two standardized versions of Raptor codes, Raptor 10 and RaptorQ codes.

3.2 Raptor 10 Code

Considering the fact that Raptor 10 code is a linear block code, the encoding and decoding algorithms of Raptor 10 code can be mathematically described from the perspective of matrix operations. A block diagram of encoding and decoding process of Raptor 10 code is shown in Fig. 3.2. This chapter only presents an overview of encoding and decoding procedures according to [4] and [41]. More details such as

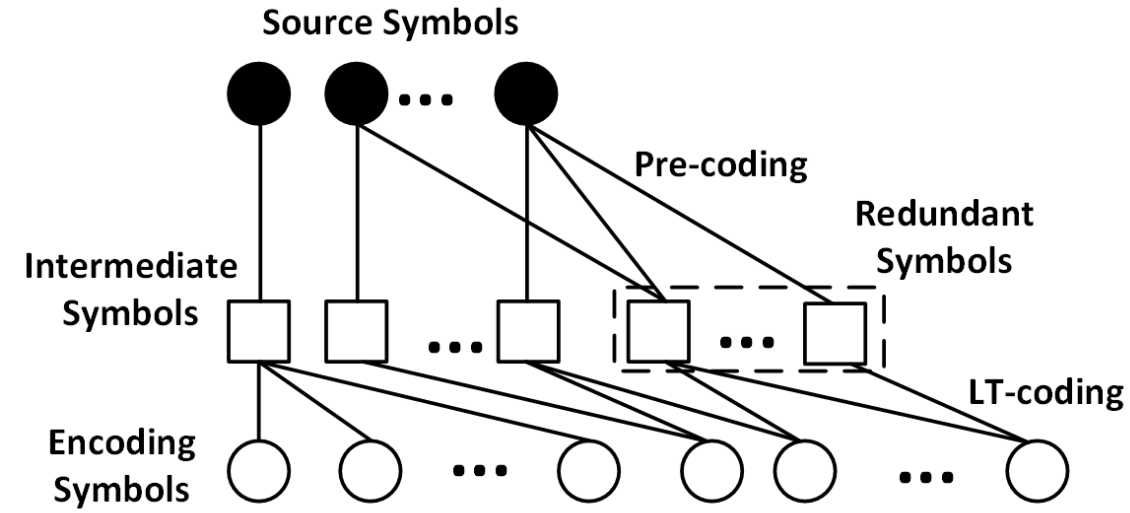

Figure 3.1: Factor graph of Raptor codes (modified on [3])

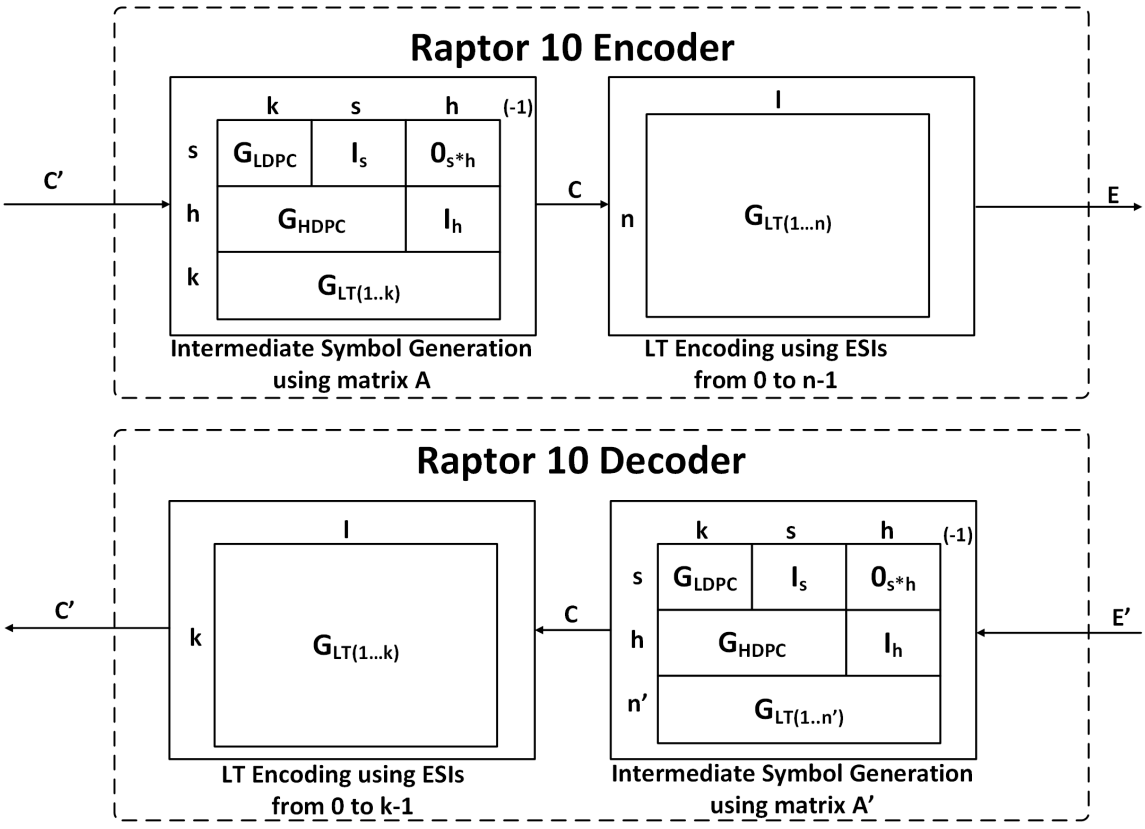

Figure 3.2: Block diagram of systematic Raptor 10 codes based on [4]

parameter calculation or matrix construction regarding Raptor 10 code can be found in [4] and [41].

3.2.1 Algorithm of Raptor 10 Encoder

Given a source block of k source symbols $C' = (C'[0], \cdots, C'[k-1])^T$ with symbol size ranging from 1 byte to 1024 bytes, the encoder of Raptor 10 code requires two steps to generate encoding symbols, intermediate symbol generation and LT encoding. Both of the two steps can be characterized by their corresponding generator matrices.

Intermediate Symbol Generation

The purpose of this step is to generate intermediate symbols from source symbols, which would be utilized to generate encoding symbols. In this step, the Raptor 10 encoder generates $l = k+s+h$ intermediate symbols denoted as $C = (C[0], \cdots, C[l-1])^T$ from source symbols $C' = (C'[0], \cdots, C'[k-1])^T$. The parameter s is the number of low density parity check (LDPC) symbols and h is the number of high density parity check (HDPC) symbols. As shown in Fig. 3.2, intermediate symbols are determined as:

$$C = A^{-1} \times D \tag{3.1}$$

where D denotes a column vector consisting of $s + h$ zero symbols and k source symbols $C' = (C'[0], \cdots, C'[k-1])^T$, matrix A is referred to as the pre-coding matrix and is an $l \times l$ matrix defined over GF(2). The composition of matrix A is presented in Fig. 3.2. The submatrices of A consist of several submatrices, namely, G_{LDPC}, G_{HDPC} and $G_{\text{LT}}(1, \cdots, k)$. G_{LDPC} is an $s \times k$ generator matrix of LDPC symbols defined over GF(2) while G_{HDPC} is an $h \times (k+s)$ generator matrix of HDPC symbols defined

over GF(256). $G_{\mathrm{LT}}(1, \cdots, k)$ is a $k \times l$ generator matrix of LT symbols defined over GF(2). The Raptor 10 code is designed to be systematic which means first k encoding symbols are identical to the the k source symbols $C' = (C'[0], \cdots, C'[k-1])^T$. The systematic property is achieved by $G_{\mathrm{LT}}(1, \cdots, k)$. Submatrices inside the pre-coding matrix A also include an $s \times s$ identity matrix I_s, an $s \times h$ identity matrix $0_{s \times h}$ and an $h \times h$ identity matrix I_h. Details of method to construct matrix A are given in [41, Sec.5].

LT Encoding

The purpose of this step is to generate potentially limitless encoding symbols from the intermediate symbols. Each encoding symbol represents a linear combination of intermediate symbols. In order for the decoder to retrieve intermediate symbols back from encoding symbols, each linear combination of of intermediate symbols should be uniquely defined. Also the linear combination relationship should be shared between encoder and decoder. For this purpose, each encoding symbol is assigned with a unique encoding symbol ID (ESI) starting from 0. For example, the first k encoding symbols are assigned with ESIs ranging from 0 to $k-1$. Given a fixed value k, each ESI is uniquely mapped to a triple (d, a, b) [41, Sec.5]. Triple (d, a, b) defines a subset of intermediate symbols which shall be summed together to obtain the encoding symbol indicated by ESI.

As shown in Fig. 3.2, encoding symbols E of size n are determined as:

$$E = G_{\mathrm{LT}(1,\cdots,n)} \times C \tag{3.2}$$

where matrix $G_{\mathrm{LT}(1,\cdots,n)}$ is an $k \times n$ generator of LT symbols defined over GF(2). ESIs

for encoding symbols E ranges from 0 to $n-1$. The i_{th} row of matrix $G_{\mathrm{LT}(1,\cdots,n)}$ is uniquely determined by the ESI associated with i_{th} encoding symbol in E. Encoding symbols with ESI starting from k are commonly referred as repair symbols. In the practical application of Raptor 10 code for data transmission, encoding symbols with sequential ESIs are typically grouped together with start ESI as a payload for data transmission. Cyclic redundancy check (CRC) codes are often employed to ensure correctness of data payload at the receiver. Thus, at the decoder each symbol is assumed to be error-free and the ESI known.

3.2.2 Raptor 10 Decoding Algorithm

Raptor 10 codes under consideration are systematic, i.e., the k source symbols are same as first k encoding symbols. Through checking ESI associated with each encoding symbol, the decoder of the Raptor 10 code can classify the received encoding symbols into two categories: source symbol and repair symbol. Specially, an encoding symbol with ESI ranging from 0 to $k-1$ is a source symbol while an encoding symbol with ESI starting from k is a repair symbol. If the first k encoding symbols are received, no decoding is required. Otherwise, the decoder would initiate a two-stage decoding process. As shown in Fig. 3.2, the Raptor 10 decoder will retrieve the intermediate symbols C first and then generate source symbols. Retrieval of intermediate symbols C are equivalent to solve a system of linear equations:

$$A' \cdot C = D' \tag{3.3}$$

where D' denotes a column vector compromised of $s + h$ zero symbols followed by n' received encoding symbols E'. Submatrices of matrix A' is presented in Fig. 3.2. Matrix A' shares a very similar structure with matrix A: first $s+h$ rows of matrix A' and matrix A are identical, remaining rows of matrix A are constructed using ESIs associated with received encoding symbols E' while those of matrix A are constructed using ESIs associated with the first k encoding symbols ranging from 0 to $k-1$. n' is usually slightly greater than k. The success of decoding is determined by the rank of A'. Decoding will be successful if and only if rank of A' is equal to l.

The essence of Equation 3.3 is matrix inversion. The matrix inversion algorithm utilized in this chapter is based on [4] and [41]. The matrix inversion algorithm requires that retrieving intermediate symbols C from D' takes place simultaneously with a three-phase matrix inversion scheme. Matrix inversion uses three matrix operations, row exchange, column exchange and row XOR (Exclusive-OR). Each time matrix A' completes one matrix operation, a corresponding symbol operation should be performed on C and D'. The corresponding operation relationship is established through two column vectors X and Y. X is initialized as $X[0] = 0, X[1] = 1, \cdots, X[l-1] = l-1$ while Y is initialized as $Y[0] = 0, Y[1] = 1, \cdots, Y[m-1] = m-1$. Each time row exchange occurs between row i and row j in pre-decoding matrix A', value exchange occurs between $Y[i]$ and $Y[j]$. Each time column exchange occurs between column i and column j in pre-decoding matrix A', value exchange occurs between $X[i]$ and $X[j]$. Each time row XOR occurs between row i and row j in pre-coding matrix A', symbol XOR occurs between $D'[Y[i]]$ and $D'[Y[j]]$. Once matrix inversion completes successfully, intermediate symbols would be correctly determined as $C[X[i]] = D'[Y[i]]$ where

$0 \leq i \leq l-1$.

The three-phase matrix inversion scheme includes three phases as follows:

Phase I

The purpose of this phase is to accelerate decoding speed by exploiting Belief Propagation (BP) algorithm [80]. It is this step that makes the matrix inversion scheme here that differs from traditional Gaussian Elimination. Two auxiliary variables, i and u (both initialized as 0), are used to help matrix manipulation. As shown in Fig. 3.3, matrix A' can be viewed as the collection of several submatrices in this phase, namely, I_i, $Z_{i*(l-i-u)}$, U_{m*u}, $Z_{(m-i)*i}$, $V_{(m-i)*(l-i-u)}$. Given $i = 0$ and $u = 0$ at very first stage of Phase I, only submatrix $V_{(m-i)*(l-i-u)}$ exists in matrix A'. This phase is carried out cycle by cycle with at most l cycles. During each cycle, one row with minimum number of 1 in $V_{(m-i)*(l-i-u)}$ would be selected and relocated to the first row of matrix $V_{(m-i)*(l-i-u)}$. Assume there are r columns in that row with 1. Then the first column with 1 in that row would be exchanged to the first column in $V_{(m-i)*(l-i-u)}$ and remaining $r-1$ columns would be relocated to the front of matrix U_{m*u}. After each cycle, i increases by 1 while u increases by $r-1$. In other words, the size of submatrix I_i increases by 1 while column size of U_{m*u} increases by $r-1$. This phase completes successfully if and only if submatrix $V_{(m-i)*(l-i-u)}$ disappears conditioned on $i+u=l$. Once this phase completes successfully, matrix A' shall be reduced to the form in Fig. 3.4. Otherwise a decoding failure occurs and more encoding symbols are required to receive for a successful decoding.

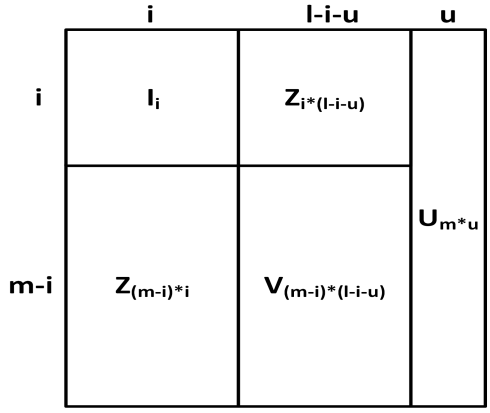

Figure 3.3: Matrix A' in first phase of matrix inversion

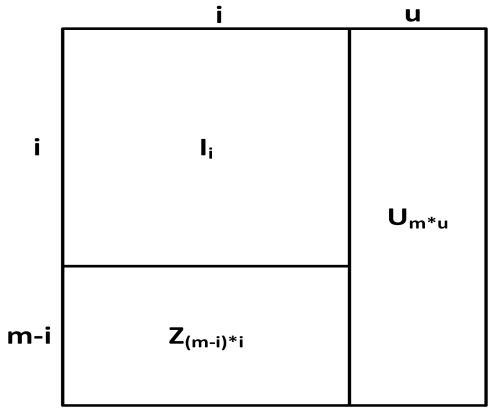

Figure 3.4: Matrix A' after the first phase of matrix inversion

Phase II

The purpose of this phase is to retrieve part of source symbols using Gaussian elimination. In this phase, submatrix U_{m*u} is partitioned into two submatrices, U_{i*u} consisting of first i rows and $U_{(m-i)*u}$ consisting of the last $m-i$ rows, as shown in Fig. 3.5. Standard Gaussian elimination is performed on submatrix $U_{(m-i)*u}$ to transform it into an identity matrix I_u. This phase completes successfully if and only if the submatrix $U_{(m-i)*u}$ can be transformed into an identity matrix. Otherwise, a decoding failure occurs and more encoding symbols are required for a successful decoding. Once the transformation completes successfully, the last $m-l$ rows of matrix A' would be discarded. At the end of this phase, the matrix A' should be in the form shown in Fig. 3.6. Note, a decoding failure can only occur during Phase I and Phase II.

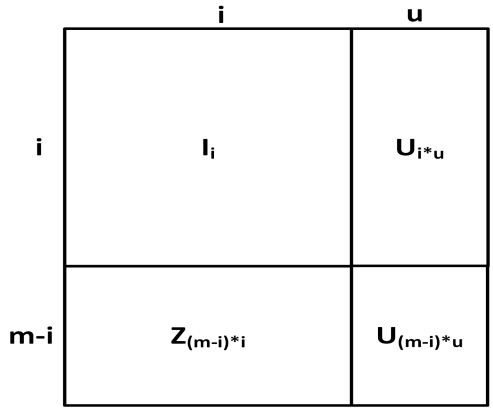

Figure 3.5: Matrix A' in second phase of matrix inversion

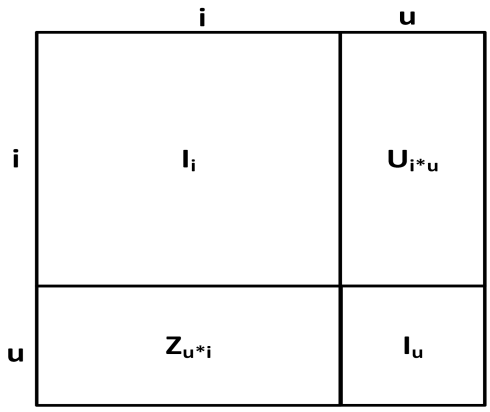

Figure 3.6: Matrix A' after second phase of matrix inversion

Phase III

The purpose of this phase is to retrieve source symbols that are not recovered in Phase II. In this phase, submatrix U_{i*u} would be transformed into a zero matrix Z_{i*u}. This is achieved by XORing each row in U_{i*u} with a certain number of rows in I_u. Matrix A' would be a identity matrix as shown in Fig. 3.7 at the end of this phase. This phase is guaranteed to finish successfully without generating a decoding error. As long as this phase completes successfully, the matrix inversion completes successfully and all intermediate symbols are recovered successfully.

Once intermediate symbols are retrieved in Equation 3.3, source symbols can be recovered as

$$C' = G_{\mathrm{LT}(1,\cdots,k)} \times C \tag{3.4}$$

where $G_{\mathrm{LT}(1,\cdots,k)}$ is a $k \times l$ generator matrix of first k encoding symbols with ESI ranging from 0 to $k-1$.

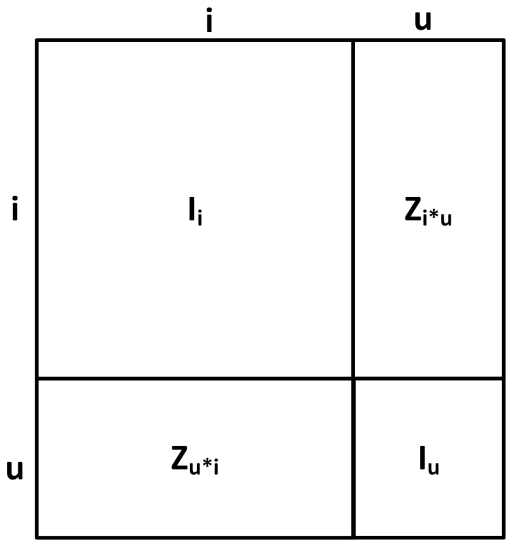

Figure 3.7: Matrix A' after third phase of matrix inversion

3.3 RaptorQ Code

Considering the fact that the RaptorQ code is linear, the encoding and decoding algorithms of RaptorQ codes can be characterized using matrix operations, as shown in Fig. 3.8. Note, the encoding and decoding algorithms of Raptor 10 and RaptorQ codes are very similar. This section only highlights the major differences between the two version of Raptor codes from perspective of matrix operation. More details of method to construct the RaptorQ code can be found in [42] and [81]. Compared with the two-step encoding algorithm of the Raptor 10 code, the encoding algorithm of the RaptorQ code includes one more step: Padding. Given a source block consisting of k source symbols denoted as $C'(1, \cdots, k) = (C'[0], \cdots, C'[k-1])^T$. Before generating intermediate symbols, RaptorQ code constructs an extended source block $C'(1, \cdots, k') = (C'[0], \cdots, C'[k'-1])^T$ first. The extended source block constructed by adding padding symbols (zero symbols) after original source block.

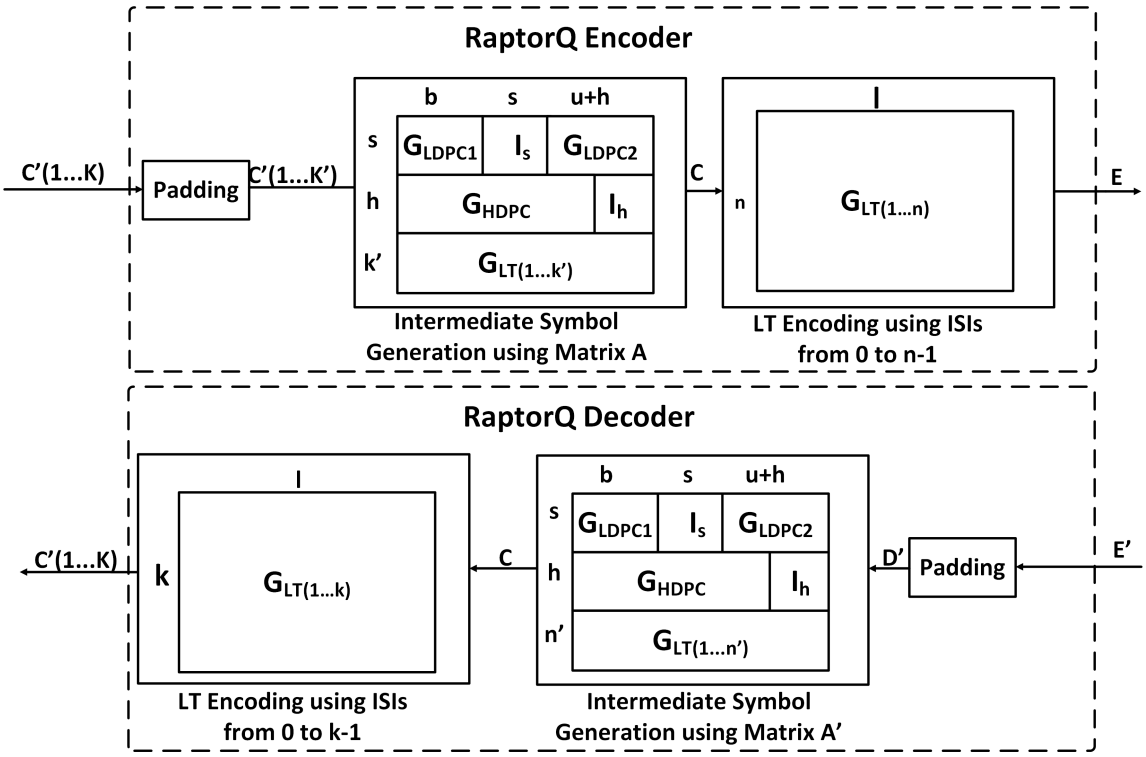

Figure 3.8: Block diagram of systematic RaptorQ codes

Specially, $k' - k$ padding symbols are added after $C'(1, \cdots, k)$ to form the extended source block $C'(1, \cdots, k') = (C'[0], \cdots, C'[k' - 1])^T$. The value of k' is uniquely determined by the value of k which can be found in [42, Sec.5]. Two advantages can be achieved by including the Padding step: improving computation speed and minimizing number of encoding symbols required for successful decoding [81]. Although the Raptor 10 code and RaptorQ code share similar pre-coding matrix, it should be noted that G_{HDPC} in the Raptor 10 code is defined over GF(2) while G_{HDPC} in the RaptorQ code is defined over GF(256). Thus, symbol operation and matrix operation in RaptorQ code are defined over GF(256) instead of GF(2).

3.4 Failure Probability

One of many important performance parameters regarding Raptor codes is the failure probability. The failure probability of Raptor codes is closely related to the overhead. The overhead o is defined as the difference between the number of encoding symbols used for decoding n' and the number of source symbols k. To numerically characterize the failure probability of Raptor codes, authors of [82] conducted an extensive decoding test for different combinations of k and o with different loss rate for both classes of Raptor codes. Details of the test result can be found in Appendix B of [82]. The overhead curve in [82] is a good reference for designing communication systems concerning Raptor codes. Compared with Raptor 10, the RaptorQ code requires less overhead to achieve the same level of decoding failure probability when the number of source symbols and loss rate are fixed. For example, the failure probability of the RaptorQ code can decrease to the order of 10^{-6} with only two overhead symbols when $k = 1000$ and the loss rate is 0.2.

However, the failure probability of the Raptor 10 is on the the order of 10^{-1} under same situation. To achieve a decoding failure probability on the order of 10^{-6} the overhead for the Raptor 10 code should be increased to 21. Note both Raptor 10 and RaptorQ codes are systematic which means they are sensitive to loss rate, i. e., no decoding probability occurs if loss rate is zero. The right choice of overhead should be the balance of failure tolerance and bandwidth utilization, especially for Raptor 10 code.

3.5 FSO Communications Employing Raptor Codes

3.5.1 Gigabit Ethernet Frame with UDP Datagram

In this work, Raptor codes are employed as forward error correction at the application layer to ensure reliable data transmission. User Datagram Protocol (UDP) is used as the transport protocol at the transport layer and Internet Protocol version 4 (IPv4) is the layer protocol at the Internet Layer. This assumption is practical and required since most real-life data communication happens over network. A diagram of Gigabit Ethernet (IEEE 802.3ab) Frame Structure with UDP Datagram is shown in the Fig. 3.9. Details of each frame component are listed in Table 3.1. Data transmission over a network is closely related to Internet protocol suite [83]. The network can be abstracted into four layers, application layer, transport layer, internet layer and link layer. In order for two communication nodes to exchange data with each, each layer employs specific network protocol to provide service for the layer above. For example, Internet Protocol version 4 (IPv4)

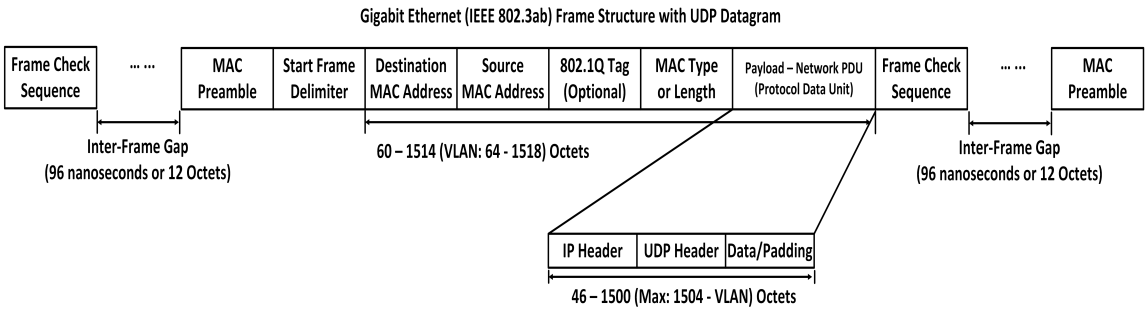

Figure 3.9: Gigabit Ethernet (IEEE 802.3ab) Frame Structure with UDP Datagram (modified based on [5])

at the Internet Layer provides packet routing service for User Datagram Protocol(UDP) at the transport layer. In this work, Raptor-coded data is first packetized into the UDP datagram which would be further encapsulated into an Ethernet packet. In this work, the total protocol overhead caused by different network layers is assumed to 54 bytes, ignoring inter-frame gap. Also this work does not consider IPv6 Jumbograms and IP fragmentation[84].

3.5.2 Packet Erasure Channel Model

As in [57], the FSO link is assumed to adopt IM/DD with OOK. The FSO channel is modelled as an additive white Gaussian noise (AWGN) channel:

$$y = hx + n \tag{3.5}$$

where y is the received signal, x is the transmitted signal, h represents the channel gain, and n is the additive white gaussian noise with zero mean and variance σ^2. The channel gain h considers the aggregated effects such as atmospheric loss, scintillation and geometric loss. The channel measurement test in [6] shows a coherence time of

Table 3.1: Gigabit Ethernet Frame Component Size With UDP Datagram (modified based on [5])

Frame Component	Size
MAC Preamble	7 Octets of: 10101010
Start Frame Delimiter	1 Octet of 10101011
Destination MAC Address	6 Octets
Source MAC Address	6 Octets
802.1Q VLAN TAG ID (Optional)	4 Octets
MAC Type or Length	2 Octets
IP Header in Payload	20 Octets
UDP Header in Payload	8 Octets
Data/Padding in Payload	18 - 1472 Octets
Frame Check Sequence (CRC)	4 Octets
Inter-Frame Gap	12 Octets

6.5 ms. Considering the fact data are transmitted at speeds of several Gbps, the FSO channel can be considered as slow-fading [6]. Details of the channel model can be found in [57] and [85].

Under OOK modulation, the bit error probability ξ is calculated as:

$$\xi = Q\left(\frac{h}{2\sigma}\right), \quad \text{where} \quad Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} * \exp\left(-\frac{t^2}{2}\right) dt \tag{3.6}$$

Note the time to transmit a single packet with packet size P ($P \leq 1526 \times 8$ bits) over a FSO link operating at 1Gbps is negligible. Thus, the channel state can be considered as constant for a single data packet. The packet erasure probability ε is defined as:

$$\varepsilon = 1 - (1 - \xi)^P \tag{3.7}$$

Since data are assumed to be transmitted as UDP datagrams in an ethernet frame, each ethernet frame transmitted through channel only has two possible

results: correctly received at receiver or not. Thus, the communication system can be modelled as the packet erasure channel (PEC) model same as in [57]. The PEC capacity C is defined as:

$$C = (1 - \xi)^P \tag{3.8}$$

The PEC capacity C does not consider the effects of overhead P_R inside each packet, it only characterizes the performance of the communication system at the physical layer. In order to describe the useful information exchange rate from user's side, the effective PEC channel capacity C_e is defined as:

$$C_e = (1 - \xi)^P * \frac{P - P_R}{P} \tag{3.9}$$

More details of packet erasure channel(PEC) model can be found in [57].

3.5.3 Simulation System Construction

A block diagram of the FSO communication system employing Raptor codes is shown in Fig. 3.10. At the transmitter side, the encoder of the systematic Raptor codes (either Raptor 10 [41] code or Raptor Q [42] code) generates encoding symbols on the fly from a source block consisting of k source symbols. The size of each source symbol is fixed as T bytes. Instead of using a conceptual data packet, this work considers a gigabit Ethernet frame introduced in Section 3.5.1. User Datagram Protocol UDP is employed as the transmission protocol at the transport layer to transmit data over the FSO link. Encoding symbols with sequential ESIs are typically grouped together with start ESI as a payload for the UDP datagram. The UDP datagram is further encapsulated into gigabit ethernet packet. As the case in most commercially

available FSO systems, on-off keying(OOK) modulation with bit-wise hard decision decoding is assumed to be adopted in the FSO communication system. Through OOK modulation, each ethernet packet is mapped to a sequence of optical intensities. It is assumed that channel gain which determines the threshold of bit-wise hard decision is available at the receiver. Demodulation at the receiver is the inverse process of the modulation at the transmitter. Note, a UDP datagram would either be correctly received or not. Through checking ESI associated with the encoding symbol, the decoder is able to decide whether the encoding symbol is a source symbol or a repair symbol. If all k source symbols are correctly received as encoding symbols, no decoding is required. Otherwise, the receiver needs to receive $k + o$ encoding symbols where o denotes the overhead for Raptor codes. Either the $k + o$ encodings leads to the success or the failure of decoding, the receiver would send a feedback signal to transmitter. The the transmitter and the receiver are synchronized with each other to move to the transmission of next source block. Such a feedback mechanism is useful especially when decoding time cannot be neglected as compared with link speed. System parameters adopted for the FSO communication system is listed in Table 3.2. The main purpose of the simulation is to investigate the performance of Raptor codes, Raptor 10 and RaptorQ code in FSO communications.

3.6 System Evaluation using Experimental Data

3.6.1 Measured FSO Channel Characteristics

Although there are many mathematical FSO models available for performance evaluation of FSO communication systems, they may still deviate from the actual

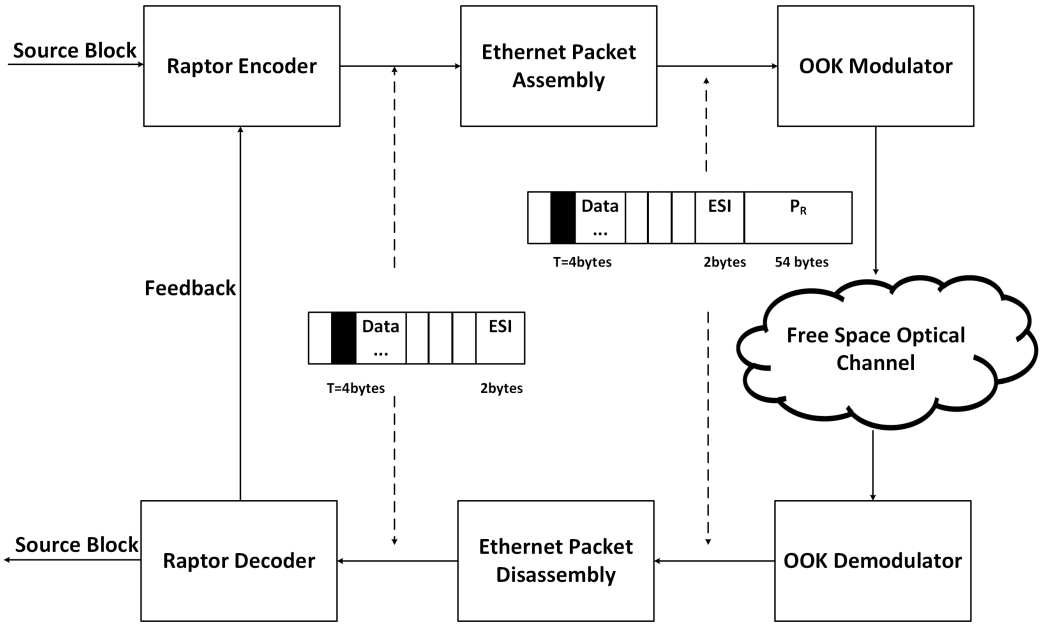

Figure 3.10: Block diagram of the Raptor-code-based FSO communication system

Table 3.2: System Parameters of Raptor-coded FSO communication system for simulation

Parameter	Symbol	Value
Signaling rate	R	1 Gbps
Packet size	P	Determined later
Protocol overhead	P_R	54 bytes
Encoding Symbol ID	ESI	2 bytes
Number of source symbols	k	64, 1000
Symbol size	T	1024 bytes

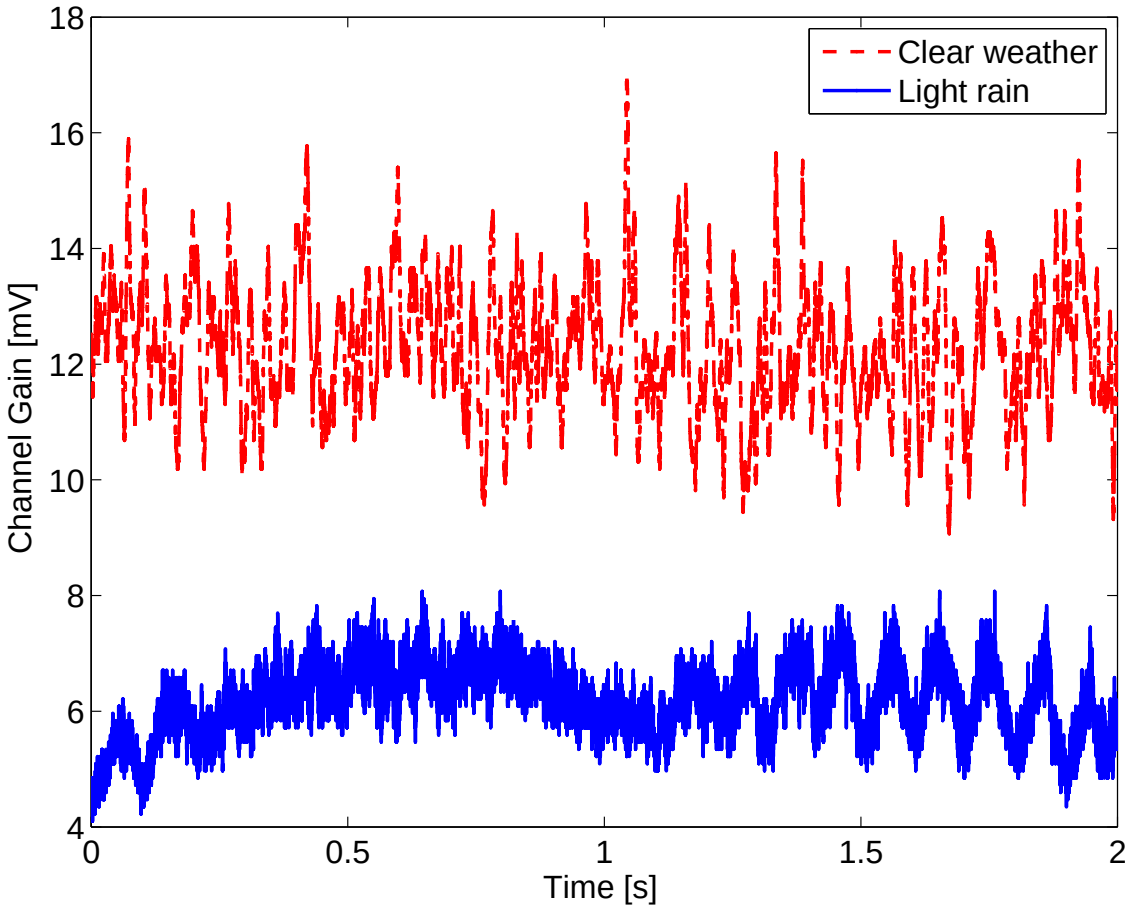

Figure 3.11: The experimental measurements of FSO channel gain recorded in clear weather at 04:13 PM, November 1st, 2011 and light rain at 08:36 AM, November 22nd, 2011 [6]

effects of FSO channel. Thus, performance evaluation in this work utilizes channel measurements collected from a 1.87 km FSO link in [6]. Channel measurements helps reproduce the effects of the FSO channel. Fig. 3.11 demonstrates the change of channel gain under two typical weather conditions, namely, clear weather and light rain. As expected, channel gain under light rain is much lower than channel gain under clear weather. Note the reduction in channel gain decreases the reliability of FSO communications.

3.6.2 Effects of Packet Size

The packet size in Table 3.2 is closely related to channel conditions. It should be determined according to channel instead of empirical assumption. To better characterize the effects of packet size, the channel capacity and effective channel capacity are evaluated under two typical weather conditions: clear weather and light rain, with different packet size 2^9, 2^{10}, 2^{11}, 2^{12} and 2^{13} bits. Channel capacity and effective channel capacity are averaged over a period of 162 seconds of channel measurements under the clear weather and the light rain separately. Simulation result of the average channel capacity $E\{\mathcal{C}\}$ and the effective channel capacity $E\{\mathcal{C}_e\}$ is listed in Table 3.3. $E\{\mathcal{C}\}$ indicates the channel condition since it directly reflects communication quality at the physical layer level. For example, $E\{\mathcal{C}\}$ under clear weather is same with different packet size because channel presents no packet erasure and channel capacity at physical layer should be same. Observing Equation 3.9, effective channel capacity would be determined mainly by packet size when packet erasure probability is zero. The bigger packet size is, the higher $E\{\mathcal{C}_e\}$ would be, as verified in Table 3.3. Thus, larger packet size such as 1024 bytes is preferred under clear weather.

As shown in Table 3.3, smaller packet size under light rain leads to larger $E\{\mathcal{C}\}$. Equation 3.7 indicates that larger packet would incur higher packet erasure probability when the bit erasure probability is fixed, resulting reduction in channel capacity. Although small packet size could achieve high throughput at physical layer, throughput at application layer would be limited due to effects of protocol overhead. For the FSO communication under consideration, 512 bytes is preferred as packet size under light rain. Considering $E\{\mathcal{C}\}$ with packet size 512 bytes is only

Table 3.3: Average channel capacity, $E\{\mathcal{C}\}$, and effective channel capacity, $E\{\mathcal{C}_e\}$, with various packet size P, different weather conditions

logP			9	10	11	12	13
1Gbps	Clear	$E\{\mathcal{C}\}$	1	1	1	1	1
		$E\{\mathcal{C}_e\}$	0.125	0.562	0.781	0.891	**0.945**
	Rain	$E\{\mathcal{C}\}$	0.787	0.725	0.659	0.590	0.520
		$E\{\mathcal{C}_e\}$	0.098	0.408	0.515	**0.526**	0.491

slightly greater than $E\{\mathcal{C}\}$ with packet size 1024 bytes, packet size in this work is chosen as value around 1024 bytes. Taking redundancy of protocol overhead and ESI, packet size in this work is determined as 1080 bytes. Note that packet overhead P_R consisting of 2 bytes ESI and 54 bytes network protocol overhead.

3.6.3 Performance of Raptor Coded FSO Communications

For the ongoing simulations, the throughput of the Raptor coded FSO communication system is calculated as the size of source block diving the time to receive enough UDP datagrams that leads to a successful decoding. Due to the rateless property of the Raptor codes, the throughput is the average level of link utilization over a small period of time. Thus, for better presentation of throughput of Raptor coded FSO communication system, the original throughput data is sampled with 101.791 kSa/s and then smoothed with a 0.01 s window. Effective packet erasure capacity (PEC) is calculated according to Equation 3.9. Simulation in this chapter utilizes the channel measurements from earlier work [6].

The RaptorQ code is considered as an improved version of the Raptor 10 code with better overhead-failure curve and computation complexities [82]. Performance of a RaptorQ coded FSO communication system is expected to be better than a

Raptor 10 coded communication system. This work studies the performance gap between RaptorQ code and Raptor 10 code in FSO communications. Two groups of simulations are conducted targeting a small and a large number of source symbols per source block, $k = 64$ and $k = 1000$. Each group of simulations includes several individual simulations to evaluate the effects of the overhead on the communication performance. Decoding failure probability and average throughput is calculated for each individual simulation. Performance evaluation of the Raptor 10 code or RaptorQ code continues for a period of approximately 145 s.

The FSO link in both groups of simulations is performed under light rain operating at 1Gbps. For each group of simulations, the optimal overhead is determined for the RaptorQ code and Raptor 10 code first. Then performance of RaptorQ and Raptor 10 codes is compared with each other. Note this work only considers the performance evaluation under light rain. Because the performance of the Raptor 10 and RaptorQ codes would be exactly the same due to the systematic property, given no packet erasure present in the FSO channel under clear weather.

Table 3.4 and Table 3.5 presents the effects of the overhead on the performance of the Raptor 10 code and RaptorQ code in FSO communications with $k = 64$. The performance the communication system is evaluated in two aspects, decoding failure probability and average throughput. The overhead of Raptor 10 code is set as 0, 4, 8 and 16 individually while the overhead of RaptorQ code is set as 0 and 1 separately. For Raptor 10 code, increasing overhead helps reduce decoding failure probability. However, it may also reduces the average throughput. Because not every successful decoding requires a higher overhead. The right choice should be a balance of the decoding failure probability and the overhead. For the RaptorQ code,

the overhead of one encoding symbol is enough to ensure successful decoding in the simulation. Error-free decoding comes at the cost of a slight decrease in average throughput. For the simulation under consideration, an overhead of 4 encoding symbols offers the highest average throughput for the Raptor 10 code while an overhead of 0 encoding symbol offers the highest average throughput for the RaptorQ code. Fig. 3.12 presents a period of 16 s of the instantaneous throughput of Raptor coded communication system along with the packet erasure channel capacity (PEC). For the simulations in the Fig. 3.12, the overhead of the Raptor 10 code is chosen 4 while the overhead of RaptorQ code is chosen as 0. As shown in Fig. 3.12, both Raptor 10 and RaptorQ codes closely follow the change of channel state and approach channel capacity regardless of channel conditions.

The same simulation is conducted for $k = 1000$. And corresponding results are listed in Table 3.6 and Table 3.7. For the simulation under consideration, an overhead of 8 encoding symbols offers the highest average throughput for Raptor 10 code while an overhead of 1 encoding symbol offers the highest average throughput for RaptorQ code. Fig. 3.13 presents a period of 16 s of the instantaneous throughput of Raptor coded communication system along with the packet erasure channel capacity (PEC). For the simulations in the Fig. 3.13, the overhead of the Raptor 10 is chosen 8 while the overhead of the RaptorQ code is chosen as 0. Same conclusion can be reached as above.

Table 3.4: Average throughput and decoding failure probability with various overhead o, $k = 64$, Raptor 10 code

Overhead	Decoding failure probability	Average throughput (Gbps)
0	0.6923	0.1881
4	0.0487	0.4471
8	0.0027	0.4436
16	0	0.4107

Table 3.5: Average throughput and decoding failure probability with various overhead o, $k = 64$, RaptorQ code

Overhead	Decoding failure probability	Average throughput (Gbps)
0	0.0062	0.4837
1	0	0.4803

Table 3.6: Average throughput and decoding failure probability with various overhead o, $k = 1000$, Raptor 10 code

Overhead	Decoding failure probability	Average throughput (Gbps)
0	0.7460	0.1419
4	0.0849	0.4451
8	0.0065	0.4793
16	3.7341×10^{-4}	0.4785

Table 3.7: Average throughput and decoding failure probability with various overhead o, $k = 1000$, RaptorQ code

Overhead	Decoding failure probability	Average throughput (Gbps)
0	0.0044	0.4839
1	0	0.4855

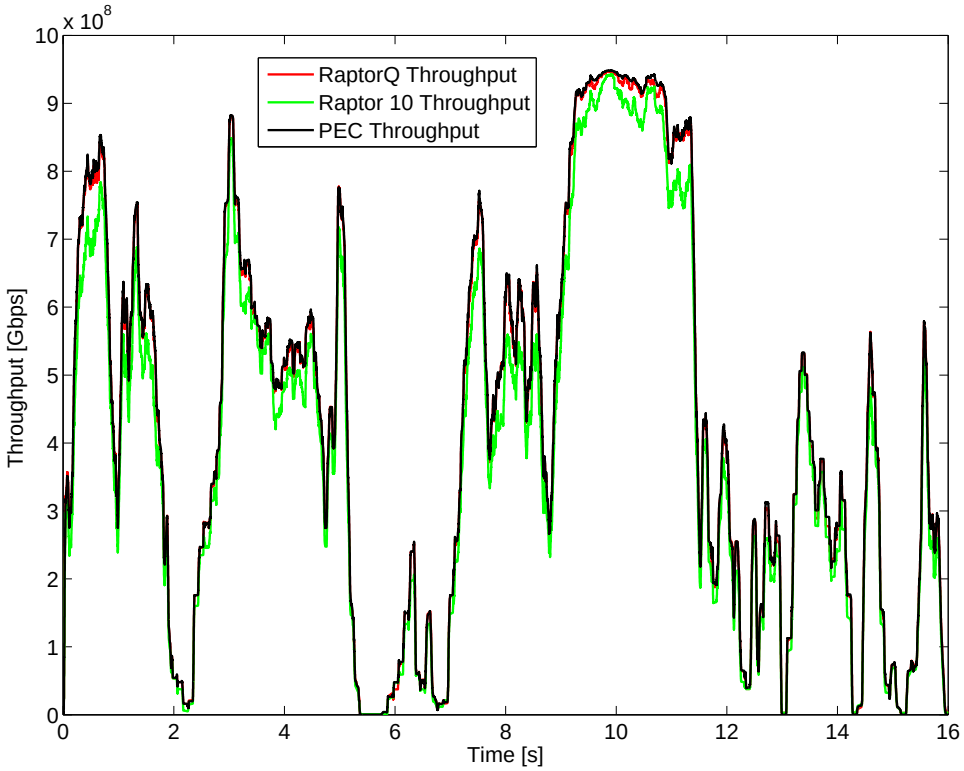

Figure 3.12: Raptor coded FSO communication system in rainy weather, 1Gbps signaling rate, $k = 64$

For the simulations under consideration, the RaptorQ code is slightly better than the Raptor 10 code. It should be noted the overhead for the Raptor 10 code should be carefully chosen in order to achieve a similar performance as the RaptorQ code. To maximize the average throughput of a FSO link, an evaluation of effects of different overhead should be conducted to choose the optimal overhead for the Raptor 10 code. Because different overhead leads to different communication performance. There is no such problem for the RaptorQ code. Because the overhead barely affects the average throughput. Additionally, the FSO link may require a negligible decoding failure probability for real-time data service. For the simulations under consideration, an

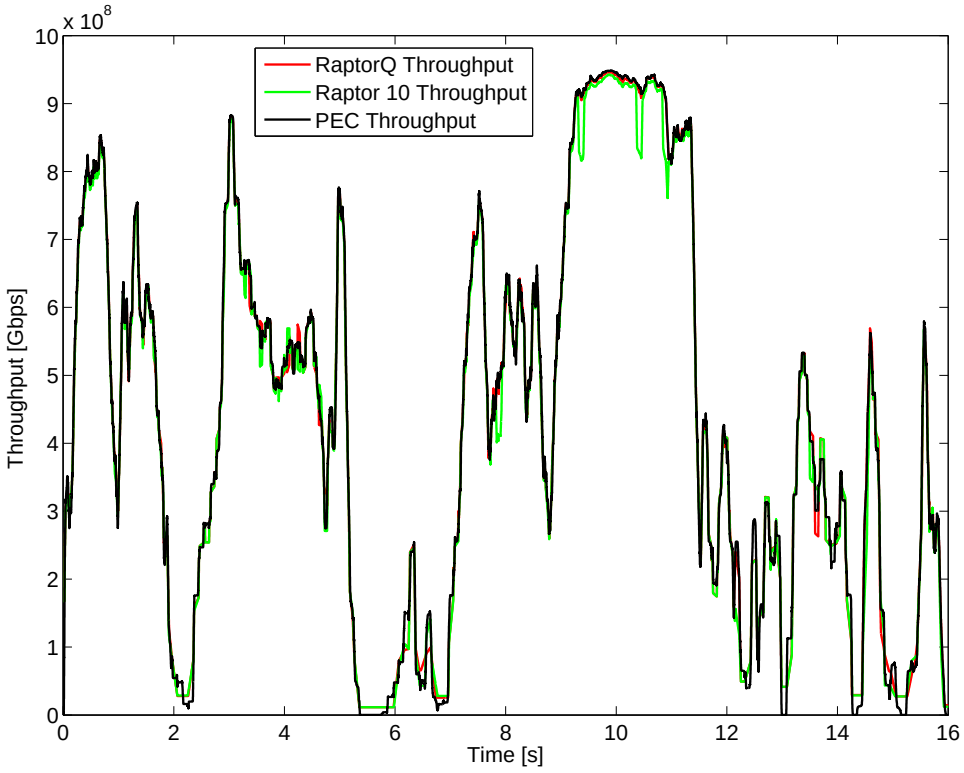

Figure 3.13: Raptor coded FSO communication system in rainy weather, 1Gbps signaling rate, $k = 1000$

overhead of at least 16 should be chosen for the Raptor 10 code and an overhead of 1 should be chosen for the RaptorQ code. Using larger overhead for the Raptor 10 code decreases the average throughput of the FSO link. An overhead of 1 leads to zero decoding failure probability for the RaptorQ code and the small overhead barely affects the average throughput. In conclusion, in order to achieve more efficient and robust communications over FSO link, the RaptorQ code is preferred.

3.7 Conclusion

In this chapter, the performance of the Raptor coded FSO link is investigated using experimental channel measurements collected over different weather conditions. The results shows that both of Raptor 10 and RaptorQ codes help FSO links follows the change of channel conditions and adapt rate. When an optimal overhead is chosen, the RaptorQ code is slightly better than the Raptor 10 code. However, the RaptorQ code is capable of achieving negligible decoding failure probability with an overhead of two, which is not possible for the Raptor 10 code. Thus, the RaptorQ code can offer more robust communication performance. For ease of implementation of Raptor codes, this work focuses on the Raptor 10 code leaving implementation of RaptorQ code to the future work.

Chapter 4

Soft-Switching Hybrid FSO/LAN link using Raptor codes

In this chapter, a Raptor-coded hybrid FSO/LAN communication system is designed and implemented. This work serves as an in-field demonstration of a hybrid FSO/RF link introduced in Chapter 1. Limited by equipment resources, the RF link is replaced with a low rate LAN link in this chapter. Compared with previous works which focus on mathematical analysis or simulation using off-line channel measurement, performance of the hybrid link is evaluated under practical weather conditions. The hybrid link under consideration is described in Section 2.2 in Chapter 2. Though the system is far from commercial application, preliminary measurement results demonstrate that Raptor-coded hybrid FSO/LAN link is an effective and efficient solution to improve the availability of FSO communication links.

4.1 Backbone Firmware

The details of experimental setup for the communication system in this Chapter are provided in the Section 2.2 of Chapter 2. Since this work focuses on a PC-based software implementation, the Triton-V5 VXS digitizer boards are not utilized. As mentioned in Chapter 2, each PC is equipped with two gigabit ethernet cards. One ethernet card is connected to a 100 Mbps campus network link while the other is connected to a 1 Gbps FSO link. The block diagram of the Raptor-coded hybrid FSO/LAN link is presented in Fig. 4.1. The Raptor 10 code, introduced in Chapter 3, is deployed at the application layer as a forward error control to guarantee reliable data transmission. Using channel coding techniques, the FSO link and LAN link combine with each other seamlessly. This is formed as a *soft-switching* scheme for hybrid link, as introduced in Chapter 1. Soft-switching hybrid links are expected to fully utilize the bandwidths of both links. User Data Protocol(UDP) is selected as a transport layer protocol to establish a connectionless data communication between transmitter (PC at MIP) and receiver (PC at BH). UDP communications between transmitter and receiver are driven by socket programming in Java [86]. At the transmitter side, source blocks are first encoded by a Raptor 10 encoder to generate potentially limitless encoding symbols. Encoding symbols with sequential ESIs are grouped together along with start ESI as a payload for UDP datagram. In order for the UDP datagram to transmit through the network, each UDP datagram is further encapsulated into an ethernet packet, as shown in Fig. 4.1. Considering the limited processing capability of CPU and the high speed of FSO link, real-time encoding and decoding of Raptor 10 code and feedback mechanism between transmitter and receiver is not implemented. The approach taken in this work is to

Table 4.1: System Parameters of hybrid FSO/LAN Link

Parameter	Symbol	Size
Number of source symbols	k	1000
Overhead of encoding symbols	o	30
Payload size		1000 bytes
Encoding symbol ID	ESI	4 bytes
Symbol size	T	32 bytes
Reserved filed	Index	4 bytes

use memory to buffer ethernet packets that would be transmitted or have been received. The encoding and decoding is conducted off-line. Multi-threading programming is utilized to transmit ethernet packets through the FSO link and LAN link simultaneously. Several transmitter threads are employed to send ethernet packets through the FSO link while one thread is employed to send packets through LAN link. The reasons for employing multiple threads send packets through FSO link will be explained in Sec. 4.2.2. At the receiver side, the same number of threads are deployed to receive packets into a common packet buffer. Due to lack of a feedback mechanism, synchronization between transmitter and receiver is assumed to happen instantly as long as receiver captures enough ethernet packets which guarantees successful decoding of a source block. Software implementation is conducted under Microsoft Windows XP [87] using Eclipse IDE [88]. The programming language is Java and Java SE development kit under use is Version 6 Update 26 [89].

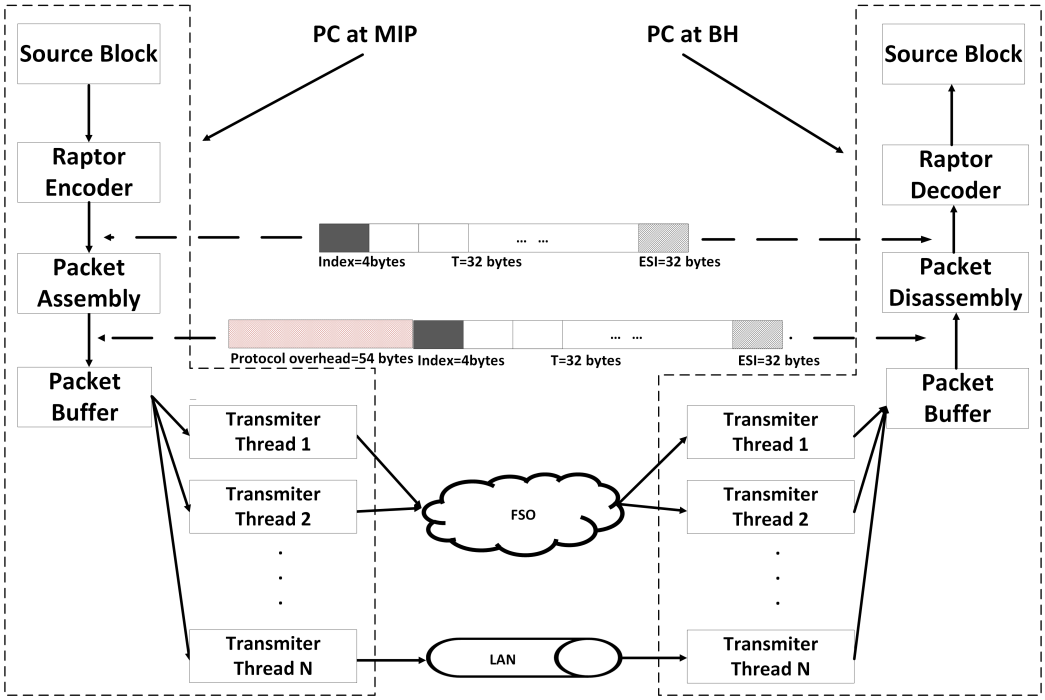

Figure 4.1: Block diagram of the Raptor coded hybrid FSO/LAN link

4.2 Communication System Construction

4.2.1 System Parameters

The experimental parameters for the hybrid link are summarized in Table 4.1. In this work, a source block consists of $k = 1000$ source symbols with symbol size equal to $T = 32$ bytes. The payload size of each UDP datagram is configured as 1000 bytes. Each payload comprises one start Encoding Symbol ID (ESI), 31 encoding symbols and one index. The index field is a reserved field set as 4 bytes. It is used for design of a feedback mechanism in future work. The overhead of encoding symbols is selected based on test results in [82, 90] and practical test. For $k = 1000$, [82] indicates an overhead of 22 encoding symbols could render decoding failure probability less than 10^{-6} and [90] claims that an overhead of 17 encoding symbols ensures 100%

of successful decoding. In this work, the overhead of encoding symbols is set as 30. A practical decoding test is conducted for approximately 10^6 times with random encoding symbols input to decoder and no decoding error is observed. Note, ethernet packets transmitted through the FSO link or LAN link have both payload and protocol overhead caused by Internet protocols.

4.2.2 *Byte Sent Throughput* versus number of Threads

Byte Sent Throughput under Windows system is defined as the percentage of connection bandwidth used by traffic sent from the machine in the polling time [91]. Utilization of network bandwidth can be roughly estimated by observing this parameter in Windows task manager, as shown in Fig. 4.2. For the 100 Mbps campus network under consideration, one transmitter thread is able to utilizing around 97% of the network bandwidth. However, one transmitter thread only occupies around 25% of the network bandwidth of FSO link. Thus, network bandwidth of the FSO link is wasted if only one transmitter thread is deployed to transmit data via the FSO link. Practical tests reveal that increasing number of transmission threads help increase utilization of network bandwidth of FSO link. Table 4.2 shows the *Byte Sent Throughput* versus the number of transmitter threads for FSO link. Though employing 4 or 5 threads still could increase Byte Sent Throughput, the increase of Byte Sent Throughput is only approximately 2% compared with employing 3 threads. Additionally, deploying multiple threads would increase the burden on CPU. In this test, three transmitter threads are deployed to transmit ethernet packets over the FSO link concurrently.

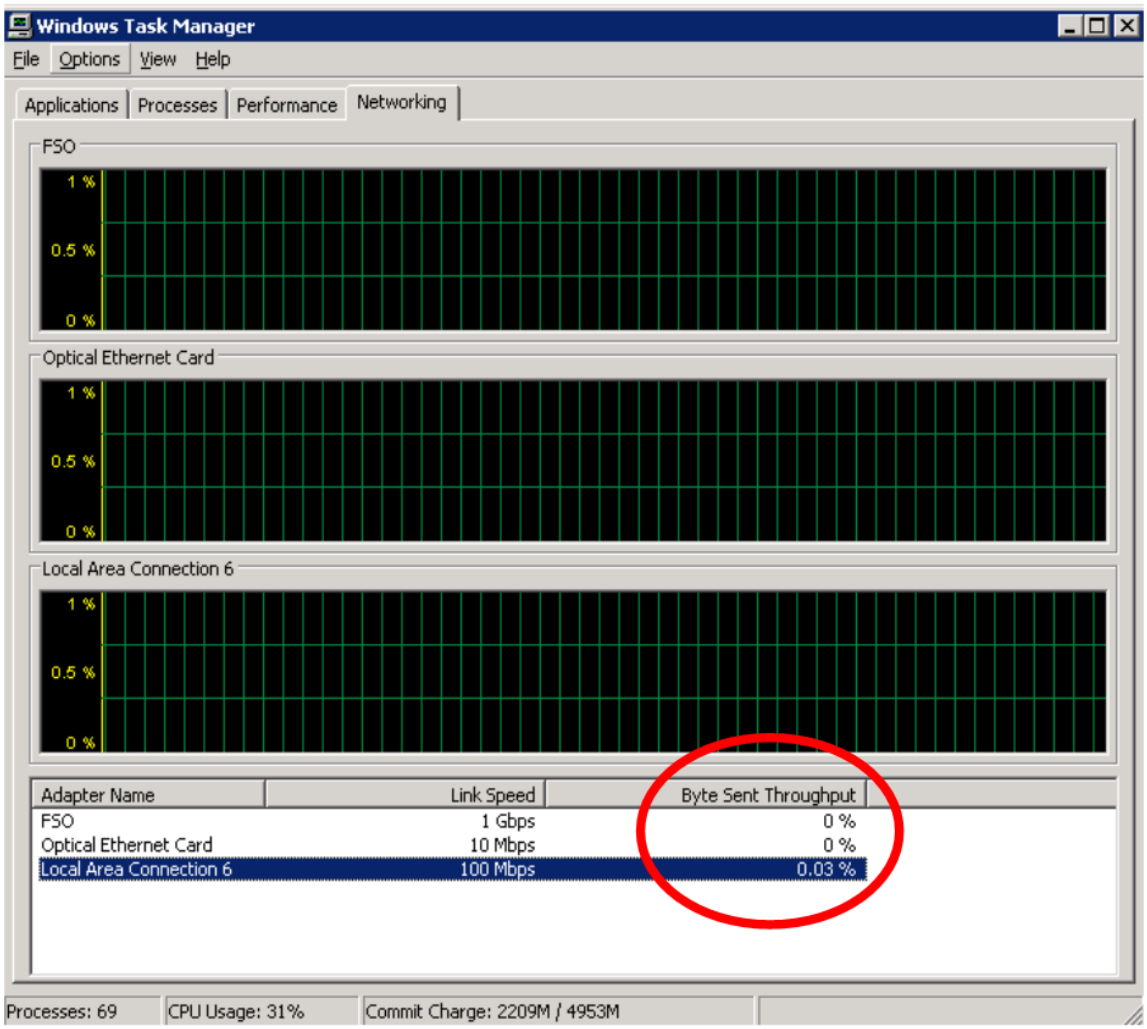

Figure 4.2: *Byte Sent Throughput* in Windows task manager

Table 4.2: *Byte Sent Throughput* versus Thread Number

Thread Number	Byte Sent Throughput
1	25%
2	53%
3	69%
4	71%
5	73%

4.3 Experimental results

To characterize the performance numerically, two parameters are defined, data rate and reliability. Since the decoding of data is conducted off-line, the data rate of the hybrid link is calculated as the size of a source block over the time to receive enough encoding symbols that guarantees successful decoding. It characterize the the number of useful information bits successfully delivered to the receiver via the hybrid link per unit time. The data rate discussed here considers information exchange rate at the application layer instead of the physical layer. Reliability is an important parameter to characterize the availability of a communication system. As in [66], the reliability in this chapter is defined as the probability that data rate is greater than a given value R_0 as

$$\text{Reliability} = \Pr(\text{data rate} \geq R_0) \times 100\% \tag{4.1}$$

In order to demonstrate the performance of Raptor coded hybrid FSO/LAN link, three categories of tests are carried out: *LAN only*, *FSO only* and *FSO and LAN*. The weather conditions for all the three categories of tests are mixed with clear weather and snow. In test *LAN only*, data packets are only transmitted through the LAN link. In test *FSO only*, data packets are only transmitted through the FSO link. In test *FSO and LAN*, data packets are transmitted over both FSO and LAN links concurrently. The three categories of tests aims to characterize the performance of LAN link, FSO link and hybrid link.

Table 4.3: Statistical observation of measurement results for LAN only

Statistics	Rate (MB/s)
Maximum	10.736
Minimum	4.997
Average	9.987
Standard deviation	0.662
Reliability=99.99 %	4.997
Reliability=99 %	8.007
Reliability=90 %	8.799
Reliability=80 %	9.702
Reliability=70 %	9.964

4.3.1 LAN only

The purpose of the test is to investigate the performance of the LAN link. The test is conducted from 8:54PM, November 16, 2014 with a duration of around 30 minutes. Pre-generated ethernet packets are transmitted through LAN link repeatedly. Note, LAN link under consideration is a 100 Mbps campus link connecting McMaster Innovation Park and main campus of McMaster University and it is shared by many campus users. The instantaneous data rate and distribution of instantaneous data rates of the LAN link is shown in Fig. 4.3 and Fig. 4.4. Statistical observation of the measurement result is listed in Table 4.3. As demonstrated by experimental results, LAN link under consideration is a relatively low-speed link with average data rate 9.987 MB/s and maximum data rate only 10.796 MB/s. The LAN link is very reliable since data rate is above 4.997 MB/s for 99.99% of the test duration. In conclusion, the LAN link under consideration is a low speed link with high reliability.

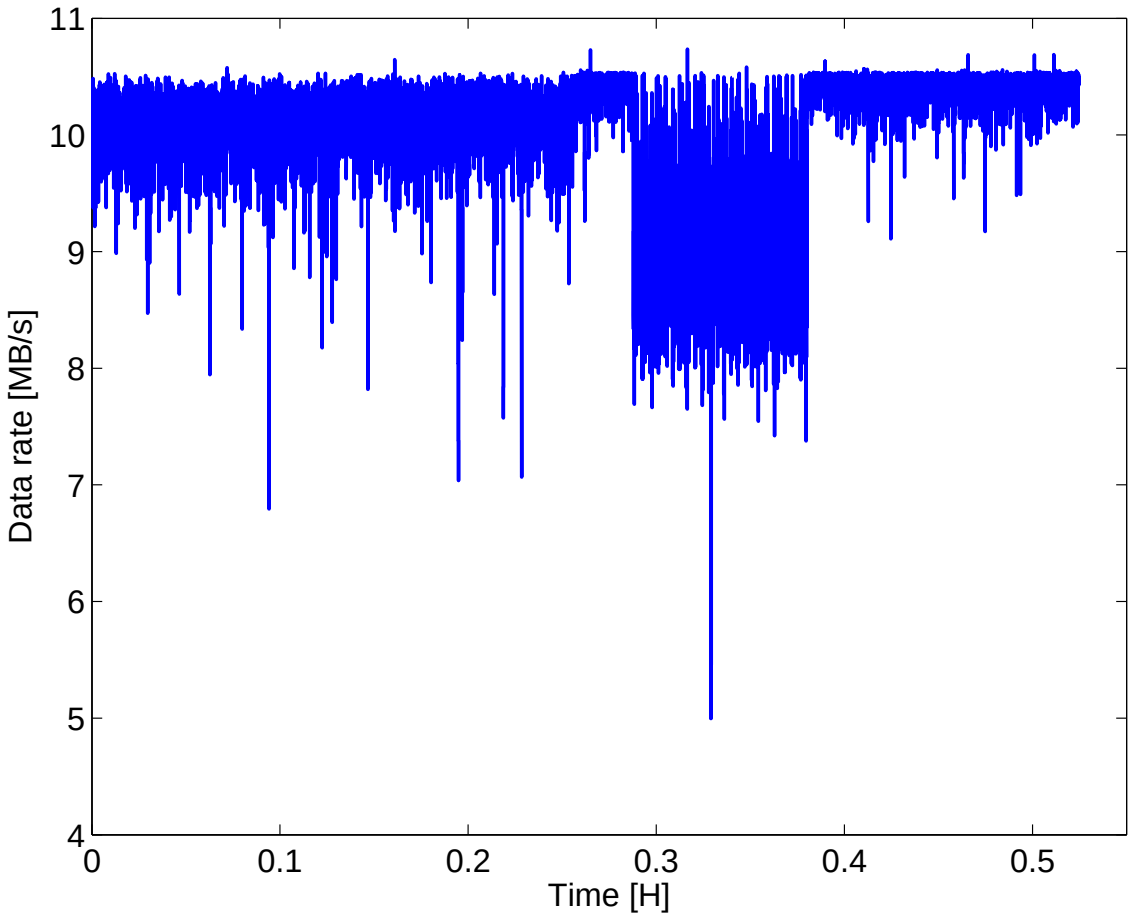

Figure 4.3: Instantaneous data rate for LAN link measured from 8:54PM, November 16, 2014 a duration of around 30 minutes

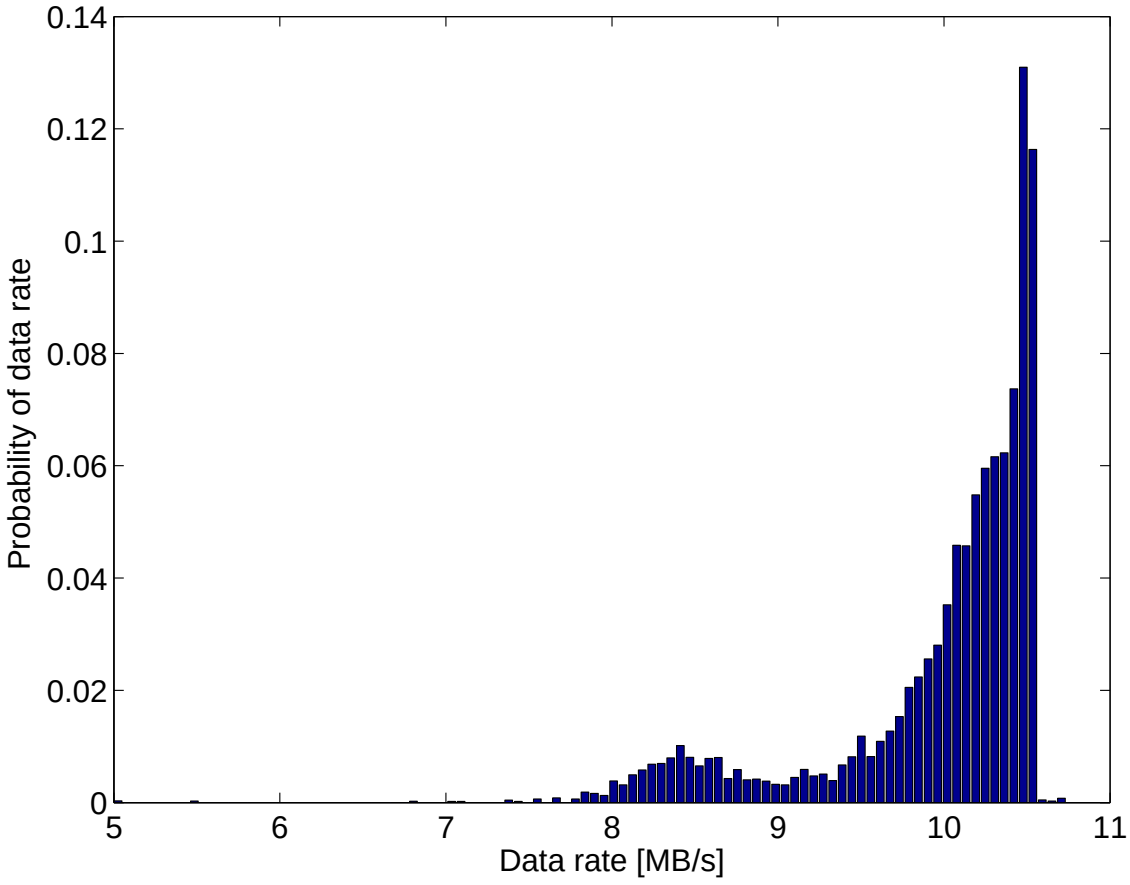

Figure 4.4: Histogram of data rates for LAN link measured from 8:54PM, November 16, 2014 a duration of around 30 minutes

4.3.2 FSO only

The aim of this test is to explore the merits and limitations of free space optical link. The test is conducted from 10:00PM November 17, 2014 with a duration around 0.8 hours. The weather condition is dominated by snow. In this test, pre-generated ethernet packets are transmitted over FSO link alone. The FSO link, as introduced in Chapter 2, is deployed across McMaster University campus and configured to operate at operate at 1.5 Gbps. The instantaneous data rate and distribution of instantaneous data rate of the Raptor-coded FSO link are shown in Fig. 4.5 and Fig. 4.6. Statistical observation of the measurement results is listed in Table 4.4. Compared with the LAN link, the FSO link is a relatively high-speed link with average data rate 49.004 MB/s and maximum data up to 76.056 MB/s. For 70% of the test duration, data rate is above 50.556 MB/s. However, availability is an issue since R_0 corresponding Reliability= 99% is only 0.019 MB/s. Also, it can be observed that link is unavailable for 19% of the time in this weather condition. Although the rateless property of Raptor 10 code could help mitigate negative effects of random erasure caused by link and increase link utilization, communication over FSO link will still be unavailable under difficult weather conditions. In conclusion, the FSO link is capable of offering high bandwidth communication but it suffers from availability issue especially under adverse weather conditions.

4.3.3 FSO and LAN

The purpose of this test is to evaluate the communication performance of the hybrid FSO/LAN link using Raptor 10 codes. The test is conducted from 11:25AM, November 18, 2014 with a duration of around 8 hours. Weather is dominated by

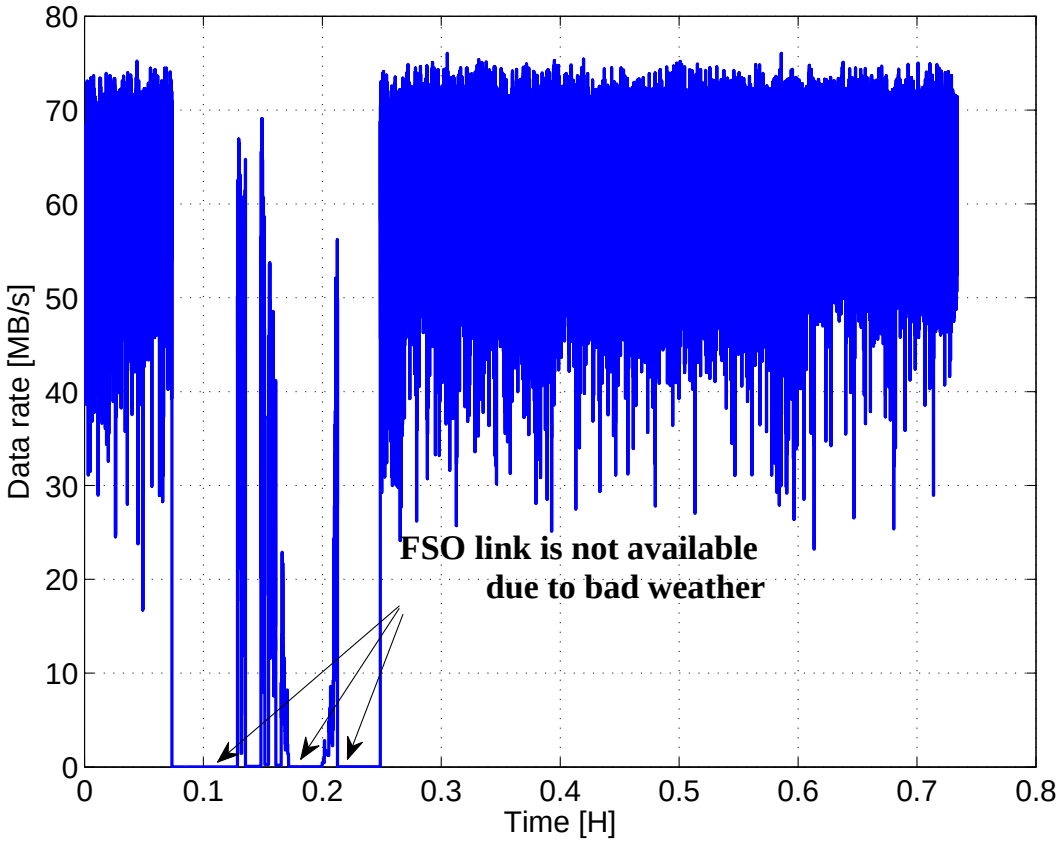

Figure 4.5: Instantaneous data rate for FSO link measured from 10:00PM November 17, 2014 with a duration around 0.8 hours under snow

Table 4.4: Statistical Observation of Measurement Result for FSO only

Statistics	Rate (MB/s)
Maximum	76.056
Minimum	0.0154
Average	49.004
Standard deviation	26.812
Reliability=99.99%	0.019
Reliability=99%	0.019
Reliability=90%	0.027
Reliability=80%	1.7985
Reliability=70%	50.556

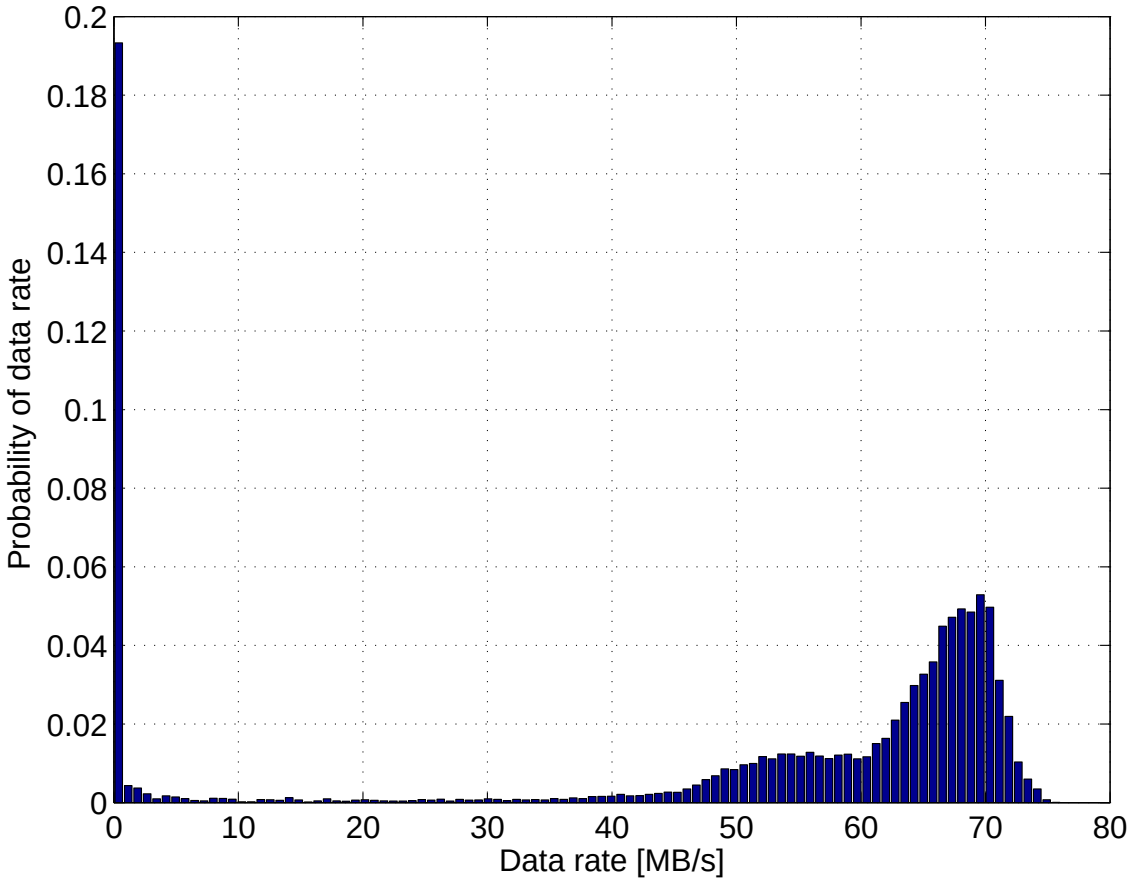

Figure 4.6: Histogram of data rates for FSO link measured from 10:00PM November 17, 2014 with a duration around 0.8 hours under snow

Table 4.5: Statistical Observation of Experimental Result for FSO and LAN

Statistics	Rate (MB/s)
Maximum	88.313
Minimum	6.480
Average	53.696
Standard deviation	29.2619
Reliability=99.99%	7.696
Reliability=99%	8.252
Reliability=90%	9.201
Reliability=80%	9.332
Reliability=70%	43.817

blowing snow. In this test, data are transmitted over FSO and LAN links simultaneously. The instantaneous data rate and distribution of instantaneous data rates of the hybrid link is shown in Fig. 4.7 and Fig. 4.8. Statistical observation of the measurement result is listed in Table 4.5. Soft-switching technique by Raptor 10 code enables the hybrid link to fully utilize the bandwidth of both links. Thus, data rate sometimes could reach as high as 88.313 MB/s. The hybrid link is capable of offering average data rate around 53.696 MB/s for the test duration. For 70% of the test duration, data rate of the link is above 43.817 MB/s which suggests that high data throughput is possible for most of the time. The availability of the hybrid link can always be guaranteed due to existence of LAN link. Although communication over the FSO link is interrupted during adverse weather conditions such snow, the LAN link remains functional to provide data communications. In conclusion, the Raptor-coded FSO/LAN link is a robust and efficient solution to circumvent the disadvantages of FSO links. On the one hand, it could combine two links seamlessly without wasting any bandwidth. On the other hand, it guarantees the availability of the overall communication system using low-speed link as “backup”.

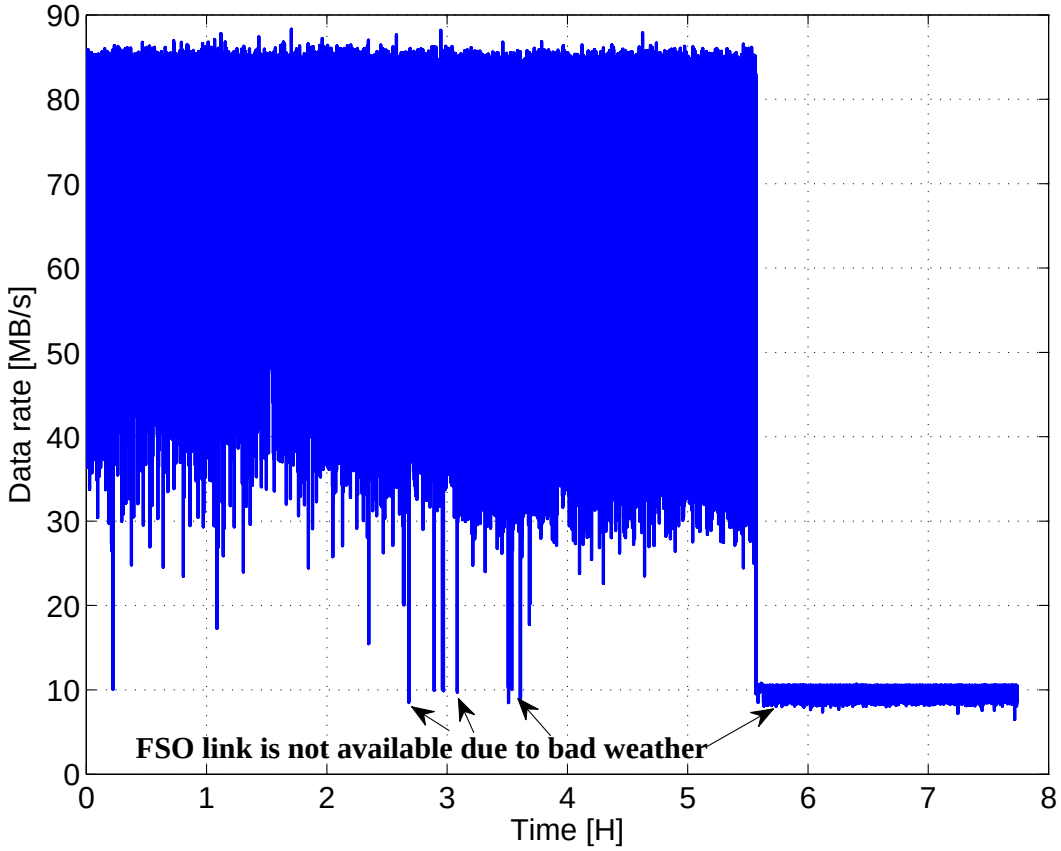

Figure 4.7: Instantaneous data rate for the hybrid link measured from 11:25AM, November 18 2014 with a duration of around 8 hours under blowing snow

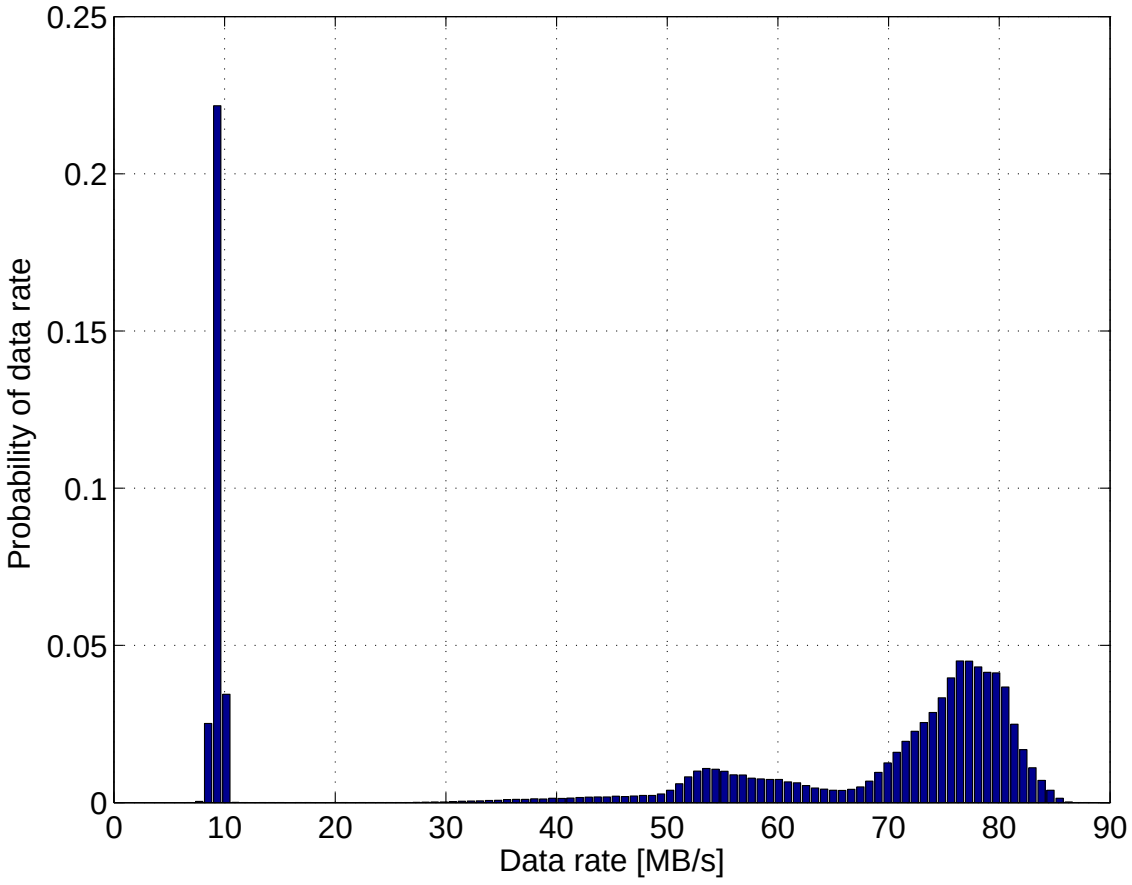

Figure 4.8: Histogram of data rates for the hybrid link measured from 11:25AM, November 18 2014 with a duration of around 8 hours under blowing snow

4.4 Conclusions

In this chapter, the experimental study of a Raptor-coded hybrid FSO/LAN link is conducted. To characterize the performance of the hybrid communication system, several measurement tests are carried out in late 2014. Weather conditions for the test duration are dominated by snow. Though many improvements are still required before link could be put into practical use, preliminary measurement results demonstrate that Raptor-coded hybrid FSO/LAN link is an efficient and effective solution to the availability issue of FSO communications. Considering the fact that free-space optical links usually operate at high speeds of several Gbps, it is difficult for software-based FSO communication system to achieve real-time decoding and high bandwidth utilization at the same time. This is the main reason to explore the FPGA-based implementation of Raptor-coded FSO link in Chapter 5.

Chapter 5

Hardware Implementation of Raptor 10 coded FSO Link

Considering the fact that the FSO link usually operates at speeds of several Gbps, the decoding speed of the Raptor 10 decoder would greatly affect the performance of the Raptor-coded FSO link, especially real-time throughput. Compared with software implementation, an FPGA-based hardware implementation of a Raptor 10 decoder is capable of offering much higher decoding speeds due to the parallel processing capability of FPGA. This chapter presents an FPGA-based hardware implementation of a Raptor 10 coded FSO link. Since the structure of transmitter is very similar to receiver, this chapter only focuses on the receiver design. The receiver mainly consists of two parts, a Raptor 10 decoder and UDP communication module.

5.1 Raptor 10 Decoder

5.1.1 Background

A brief introduction to the Raptor 10 code can be found in Section 1.2 of Chapter 1. An overview of the encoding and decoding algorithms of the Raptor 10 code can be found in Section 3.2 of Chapter 3.

The most challenging work regarding practical application of Raptor 10 code is the design and implementation of the decoder. As long as the decoder is successfully obtained, only minor changes are required to transform the decoder into an encoder. Given n' received encoding symbols, decoding is initiated by recovering intermediate symbols C from a known pre-coding matrix A', which depends on the ESIs of received encoding symbols, and D', which consists of $s+h$ zero symbols followed by n' received encoding symbols. As in [41, Sec.5], retrieval of intermediate symbols is equivalent to solve a system of linear equations:

$$A'_{m\times l} \times C = D' \tag{5.1}$$

with

$$A'_{m\times l} = \begin{bmatrix} G_{\text{LDPC}} & I_s & \mathbf{0}_{s\times h} \\ & G_{\text{Half}} & I_h \\ & & G_{\text{LT}(1,\cdots,n')} \end{bmatrix} \tag{5.2}$$

and where $m = s + h + n'$, submatrices I_s and I_h are identity matrices, $\mathbf{0}_{s\times h}$ is an $s \times h$ zero submatrix, G_{LDPC} is a $s \times k$ generator matrix of LDPC symbols, G_{Half} is a $h \times (k + s)$ generator matrix of HDPC symbols, G_{LT} is $n' \times l$ generator matrix of LT symbols with i_{th} row is determined by ESI for i_{th} received encoding symbols. For

successful decoding, the rank of A' must be l. Given intermediate symbols C from Equation 5.1, source symbols are computed as

$$C' = G_{\mathrm{LT}(1,\cdots,k)} \times C \tag{5.3}$$

FPGA-based hardware implementation of Raptor 10 decoder mainly deals with Equation 5.1 and Equation 5.3. Equation 5.3 is easy to achieve since it represents XOR combinations of a subset of intermediate symbols. Basic FPGA components such as adders and selectors could be exploited to implement Equation 5.3. The key to implement Equation 5.1 is matrix inversion over GF(2), which requires efforts to design from scratch. Also implementation of Equation 5.1 on FPGA should take both time-efficiency and resource-efficiency into consideration.

5.1.2 Matrix Inversion over GF(2)

As mention earlier, the essence of Equation 5.1 is matrix inversion over GF(2). Theoretically, any algorithm applicable to solve linear equations could be used to implement Equation 5.1 on FPGA. However, matrix inversion algorithm should be carefully chosen since digital system design on FPGA should consider both time complexity and resource complexity. Traditional Gaussian Elimination(GE) algorithm is capable of achieving time complexity of $\mathcal{O}(N)$ at the expense of excessive usage of hardware resources. In this work, matrix inversion algorithm proposed in the FOCAL lab [70] is utilized to implement Equation 5.1. The matrix inversion algorithm in [70] is built on two previous works [71] and [72] while adding several innovative features such as row swap operation. It achieves the balance of [71] and [72] in terms of time complexity and hardware resource. In [71], hardware

resource efficiency is achieved while time complexity is still a issue with worst time complexity $\mathcal{O}(N^2)$ and average time complexity $\mathcal{O}(N)$. Time complexity is improved to $\mathcal{O}(N)$ in later work [72] but with the expense of excessive memory resources. Work in [70] manages the balance between [71] and [72] and is capable of inverting a matrix with size $N \times N$ within N to $2N$ clocks by consuming hardware resources between [71] and [72].

5.1.3 Hardware Structure

Both the encoder and decoder for Raptor 10 codes include similar structures. In this work, we focus on the implementation of decoder which is the challenging part of the implementing Raptor 10 codes. Figure 5.1 presents the hardware architecture of Raptor 10 decoder, which consists of three modules: Matrix Load, Matrix Inversion and Source Symbol Generation. The three modules correspond to a three-phase decoding process to recover a source block. For the purpose of illustration, it is assumed that RAM Encoding_symbol stores D', i.e., $s + h$ zero-valued check symbols and n' encoding symbols. Additionally, the memory block RAM ESI stores the ESIs for the n' received encoding symbols. In the application of this decoder, the content of RAM Encoding_symbol and RAM ESI are updated as received encoding symbols and their corresponding ESIs are collected from FSO links.

Matrix Load

The main function of Matrix Load module is to generate the matrix $A'_{m\times l}$ in (5.2) and store it in appropriate memory cells in module Matrix Inversion. At the end of this phase, each cell in Matrix Inversion stores the values of the augmented matrix

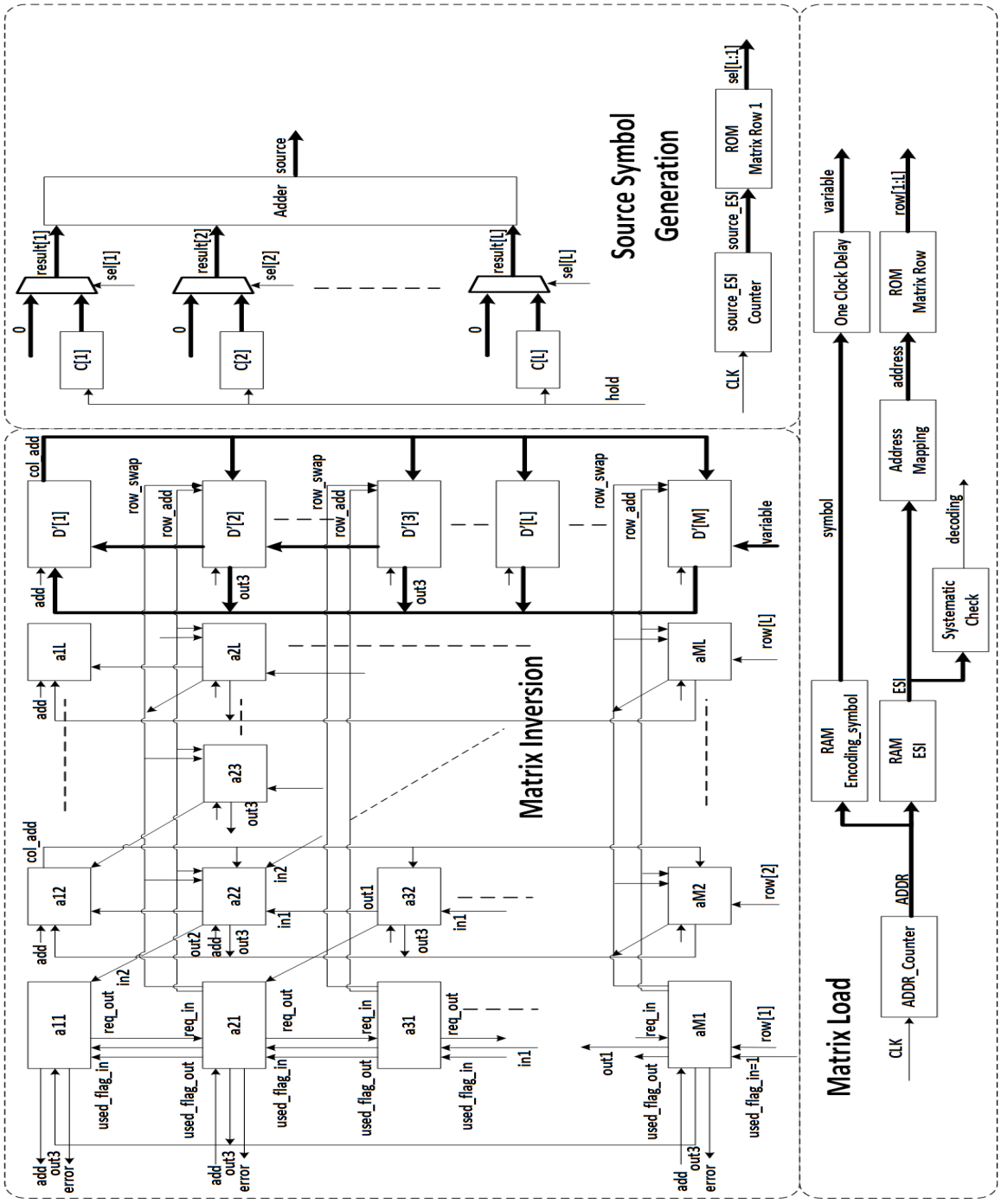

Figure 5.1: Hardware structure of proposed Raptor 10 decoder

$[A'_{m\times l}|D']$ defined in (5.1). Cell aij stores bit-value a_{ij} corresponding to $A'_{m\times l}[i][j]$ and cell D'[i] stores symbol $D'[i]$, where $i = 1, \cdots, m$ and $j = 1, \cdots, l$.

Matrix Load sequentially generates the rows of $A'_{m\times l}$, i.e., row[1:l], and D', i.e., variable. The counter ADDR_Counter outputs ADDR ranging from 0 to $m-1$ which addresses both RAM Encoding Symbol and RAM ESI. The memory block RAM ESI stores place holders for $s+h$ check symbols and the ESIs for the n' received encoding symbols. The Address Mapping module generates addresses to fetch the appropriate row in the pre-computed memory ROM Matrix Row corresponding to the rows of $A'_{m\times l}$ in (5.2). A one clock delay is inserted to signal variable to synchronize it to the corresponding row[1:l] due to the latency in addressing the ROM.

The Systematic Check block monitors whether the first k encoding symbols are received correctly. If all source symbols are received as encoding symbols, no further decoding is required due to the systematic property of Raptor 10 code and the global signal decoding is set to 1 to halt the decoding process and move to the next source block.

Matrix Inversion

The Matrix Inversion block retrieves intermediate symbols from $A'_{m\times l}$ and D' via Equation 5.1. The implementation of Matrix Inversion module in this work takes advantage of the matrix inversion algorithm in [70]. Matrix inversion algorithm in [70] is built on earlier matrix inversion approaches in [71] and [72] and balances hardware complexity with speed of inversion operation. Details of the matrix inversion algorithm can be found in [70].

To better illustrate the work flow of the Matrix Inversion module, a numerical

example is given to explain how the algorithm works in decoding. This example should only be considered for illustrative purposes. Assume we have an augmented matrix as shown in Step 1 in Fig. 5.2 . Matrices with dashed brackets indicate intermediate steps while the solid brackets indicate the final result at the end of each clock cycle. The matrix in Step 2 denotes the result after row elimination in which the first row is XORed with all rows with a leading "1". A corresponding diagonal shift operation are then performed in Step 2 for all elements. Step 3 shows the diagonal shifted matrix where the last row has already been processed and no longer participates in row swap for the following steps. Since the pivot element is in Step 3 "0", an additional clock cycle is needed for row swapping, as shown in Step 4. Step 5 is an intermediate step during which XORs occur after which a diagonal shift takes place to yield Step 6. Decoding proceeds in the same manner until Step 9 where the matrix has been successfully inverted and intermediate symbols are recovered in the last column.

Source Symbol Generation

The Source Symbol Generation module calculates source symbols from the computed intermediate symbols using (5.3). Once Matrix Inversion completes successfully, the l intermediate symbols appear at last l locations of D'. As long as hold is asserted, the data in cells D'[m-l+i], $i = 1, \ldots, l$ are transferred to C[i]. The signal source_ESI addresses a ROM which contains the pre-computed rows of matrix $G_{\mathrm{LT}(1,\cdots,k)}$. Matrix multiplication for each row of $G_{\mathrm{LT}(1,\cdots,k)}$ with C is via a series of multiplexers which implement multiplication by a single bit. It should be noted that the data paths for signals in this block are equivalent to T, the size of the encoding symbols.

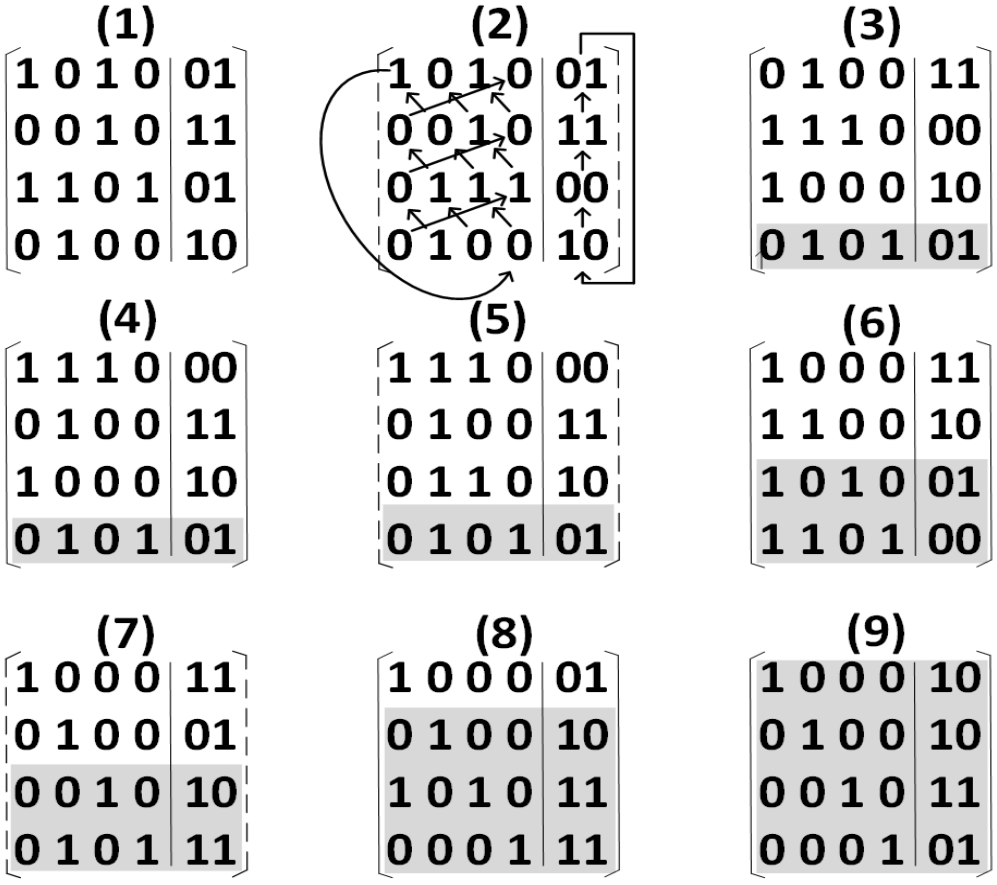

Figure 5.2: Illustrative example of matrix inversion: successful decoding

5.2 UDP communication module

5.2.1 Quixtream Gigabit Ethernet Protocol

UDP communication between transmitter and receiver is established using third-party network protocol, Quixtream Gigabit Ethernet Protocol [69] which is developed TEK Microsystems Inc. based on standard User Datagram Protocol(UDP) [92]. Quixtream Gigabit Ethernet Protocol can be employed both in software and FPGA-based hardware. The network protocol is fully compatible with the hardware utilized in this experiment including PC and QuiXilica Triton-V5 VXS high-speed digitizer board. There are three categories of communication channel available in the network protocol, stream-based channel, block-based channel or Remote Direct Memory Access(RDMA) channel [69]. Manipulation of input data and output lies at the application layer while the network protocol deals with all the underlying network layers to transmit or receive data. Currently, users cannot modify the underlying communication mechanism inside Quixtream Gigabit Ethernet UDP/IP protocol. Potential users could only access a number of synthesized netlists of channel controllers. Details of the ethernet protocol should be referred in [69]. With Quixtream Gigabit Ethernet Protocol, communication nodes such as PCs and FPGAs inside the same local network could exchange information with each other.

5.2.2 Encoding Symbol and ESI Recovering Module

The main function of Encoding Symbol and ESI Recovering module is to retrieve each encoding symbol and its corresponding ESI from Quixtream protocol interface into

Table 5.1: Important signals inside UDP communication module [69]

Signal Name	Description
R_Clk	Clock to read data, set as 31.25 MHz
R_Rst	Asynchronous active-high reset
R_En	Active high when reading data from output FIFO
R_Rdy	Active high when data is available in output FIFO
R_Data	Output data bus, width is 32 bits
R_EOS	Active high when data is the last 32 bits of the UDP datagram
Encoding_symbol	Encoding symbol, width is 32 bits
ESI	Encoding Symbol Identity, width is 32 bits

memories. The interface between the Raptor 10 decoder and the communications protocol can be viewed as an asynchronous FIFO. In this work, data are clocked out from the UDP interface using a 31.25 MHz clock with data width 32 bits. For ease of FPGA implementation, the symbol size is limited to 4 bytes and the UDP datagram size is chosen as 68 bytes which includes 4 bytes for ESI and 16×4 bytes of data payload. At least 17 clocks (31.25 MHz) are required to receive a single UDP datagram at application level, where the Raptor 10 decoder operates. Given UDP communication protocol and Encoding Symbol and ESI Recovering module, UDP communication module is implemented as shown in Fig. 5.3. Important signals inside UDP communication module are listed in Table 5.1.

5.3 Pipelined Receiver Architecture

In order to fully exploit link bandwidth, a pipelined architecture is required to ensure the decoder is not sitting idle waiting for encoding symbols. That is, a pipelined architecture enables the concurrent reception of symbols and decoding. Although this section only discusses a pipelined receiver, the same principle could be applied

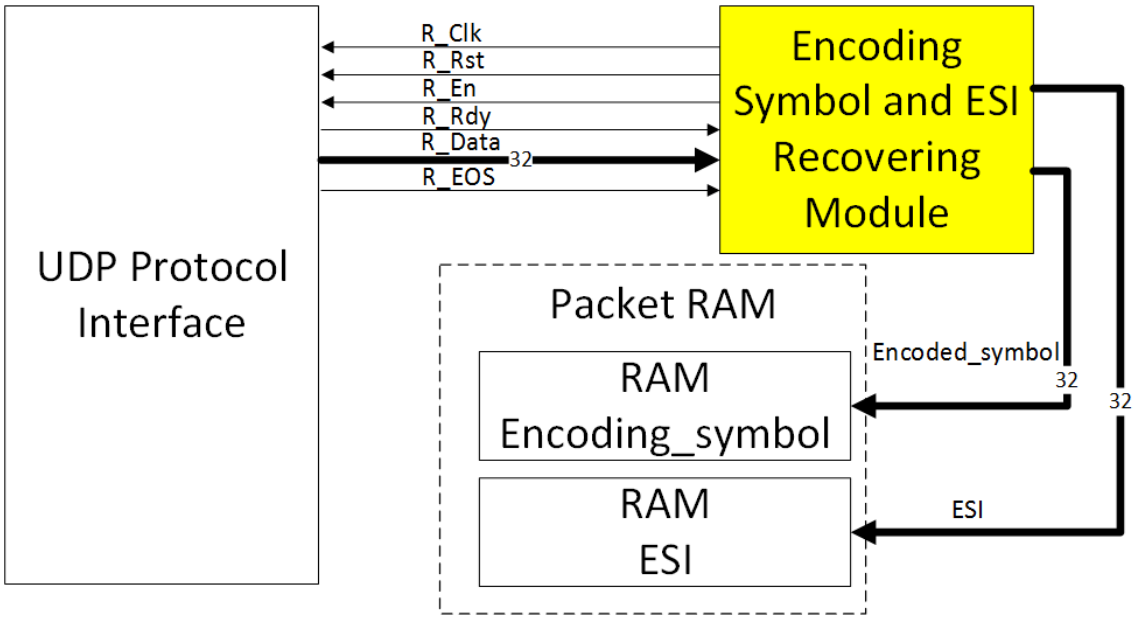

Figure 5.3: Architecture of UDP communication module

to the encoder. In addition, the receiver discussed here is useful for both FSO links as well as hybrid FSO/RF links.

Given the implemented Raptor 10 decoder, a pipelined receiver architecture is proposed in Fig. 5.4. It is assumed in the discussion that encoding symbols and ESIs are transmitted through UDP packets, although other protocols are possible. Two identical Raptor 10 decoders are implemented and a finite state machine (FSM) is developed to control concurrent packet collection and decoding as shown in Fig. 5.4. The FSM has 8 states, denoted State 0 to State 7, with the following functions:

- State 0: Store UDP packets to RAM 0, Decoder 0 recover source block from RAM 1
- State 2: Store UDP packets to RAM 2, Decoder 1 recover source block from RAM 3

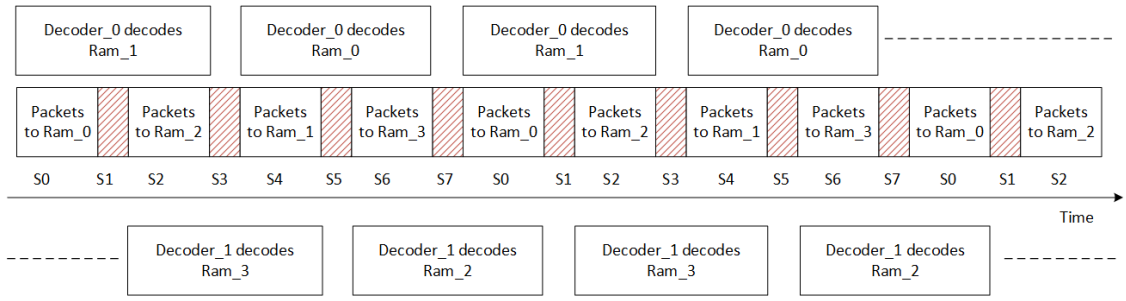

Figure 5.4: Block diagram of pipelined receiver

- State 4: Store UDP packets to RAM 1, Decoder 0 recover source block from RAM 0
- State 6: Store UDP packets to RAM 3, Decoder 1 recover source block from RAM 2
- State 1, 3, 5, 7: Intermediate states where transmitter and receiver are synchronized with each other to move to next source block

Depending on the ratio of decoding speed to the time for packet collection, the number of states in the FSM can vary. For the proposed pipelined receiver, the decoding speed should be at least half of the receiving speed. With a fully functional pipelined receiver, the bottleneck of link throughput is delegated to time necessary to collecting sufficient encoding packets.

5.4 Performance of Hardware Implementation of Raptor-coded FSO Link

The parameters of the implemented Raptor 10 decoder and the communication system are summarized in in Table 5.2. In this work, each source block consists of 64 source

Table 5.2: System Parameters of Raptor-coded FSO link

Parameter	Symbol	Value
Number of source symbols	k	64
Column size of matrix A	l	86
Row size of matrix A	m	102
Symbols size	T	32 bits
UDP clock frequency	N/A	31.25 MHz
Decoding clock frequency	CLK	62.5 MHz
Encoding Symbol ID	ESI	32 bits
Payload in UDP datagram	N/A	68 bytes

symbols with symbol size of 4 bytes. Based on [82, 90] and practical software-based decoding tests, the number of overhead symbols to ensure negligible decoding failure is chosen as 16. The decoder is driven by a 62.5 MHz clock. Since the receiver and transmitter share similar hardware architectures, here we focus only on the receiver side while the behaviour of transmitter is simulated by transmitting pre-generated network packets repeatedly.

The performance of the system is evaluated from two perspectives: decoding performance and link throughput. The *decoding performance* measures the rate and which the Raptor 10 decoder can retrieve source symbols from the received encoding symbols. Sec. 5.4.1 measures the decoding performance of the Raptor 10 decoder presented here. In contrast, *link throughput* is evaluated at the application layer considering the impact of protocol overhead and FSO channel conditions. This aspect of our experiment improves on earlier literature which typically consists of simulations with negligible protocol overheads and idealized channel models.

Although the FSO link operates at 1 Gbps, the maximum achievable throughput at the application layer in this experiment is far less than 1 Gbps. The reduction in

link throughput arises due to the protocol overhead in the particular implementation selected [69]. Practical test reveal an idle time period of approximately 29 cycles (31.25MHz) between the arrival of two consecutive packets at the application level, which corresponds to a protocol overhead of 116 bytes. For the data payload under consideration, the maximum link throughput at the application layer is 0.370 Gbps. Future work may consider improved transport protocols using larger packet sizes to increase throughput at application layer. For example, application level throughput can be made as high as 0.926Gbps if the packet size is chosen as 1452 bytes. Section 5.4.2 measures the link throughput of of Raptor 10 system under a variety of weather conditions.

5.4.1 Decoding Performance

Due to the systematic property of the Raptor 10 code, the decoder is not run if the first k encoding symbols are received. Thus, this section analyzes the decoding performance of the implemented Raptor 10 decoder when the systematic property is not utilized (i.e., when some of the first k encoding symbols are not received).

For the Raptor 10 decoder under consideration, m clocks are required by Matrix Load to load values from RAM to the Matrix Inversion module. To successfully invert, the Matrix Inversion requires between l to $2l$ clock cycles to retrieve the intermediate symbols. Also, k clocks cycles are required to generate k source symbols via the Source Symbol Generation module. Since a new round of decoding can be initiated without waiting source symbol generation, due to buffer C[i] in Fig. 5.1, the k clock cycles needed by Source Symbol Generation can be omitted in evaluating decoding performance. Theoretically, it takes between $l + m = 188$ to $2l + m = 274$ clocks

(62.5 MHz) to recover a source block, which gives theoretical decoding speeds between 0.467 and 0.681 Gbps.

To measure the decoding performance of the implemented Raptor 10 decoder, random encoding symbols along with their corresponding ESIs are input to the decoder to measure the decoding time. Fig. 5.5 presents the distribution of decoding times for 400161 successful decoding events. For the decoding test under consideration, decoding times range from 233 to 260 clock cycles with an average decoding time 250 cycles and standard deviation approximately 3.29 clock cycles. Thus, the hardware decoder is capable of offering average decoding speeds 0.512 Gbps with highest decoding speed of 0.549 Gbps and lowest decoding speed 0.492 Gbps. The decoding speed in the architecture in Fig. 5.1 can be theoretically be improved by increasing the symbol size since inversion time dominates the decoder. For example, increasing the symbol size to 64 bits doubles the decoding speed at the cost of extra hardware resource.

5.4.2 Measured link throughput

To measure the throughput of the developed Raptor 10 system, pre-generated packets were repeatedly transmitted at one end of the link and decoded in real time with the pipelined receiver architecture. Each UDP datagram carries a starting ESI and 16 encoding symbols. A feedback signal is assumed to be generated under two conditions. Firstly, the receiver collects four UDP datagrams containing consecutive ESIs. That is, the systematic source symbols are correctly received and the decoder need not be run. Secondly, if the systematic portion of the message is not received, the receiver collects an additional UDP datagram (i.e., 5 in total) before initiating decoding.

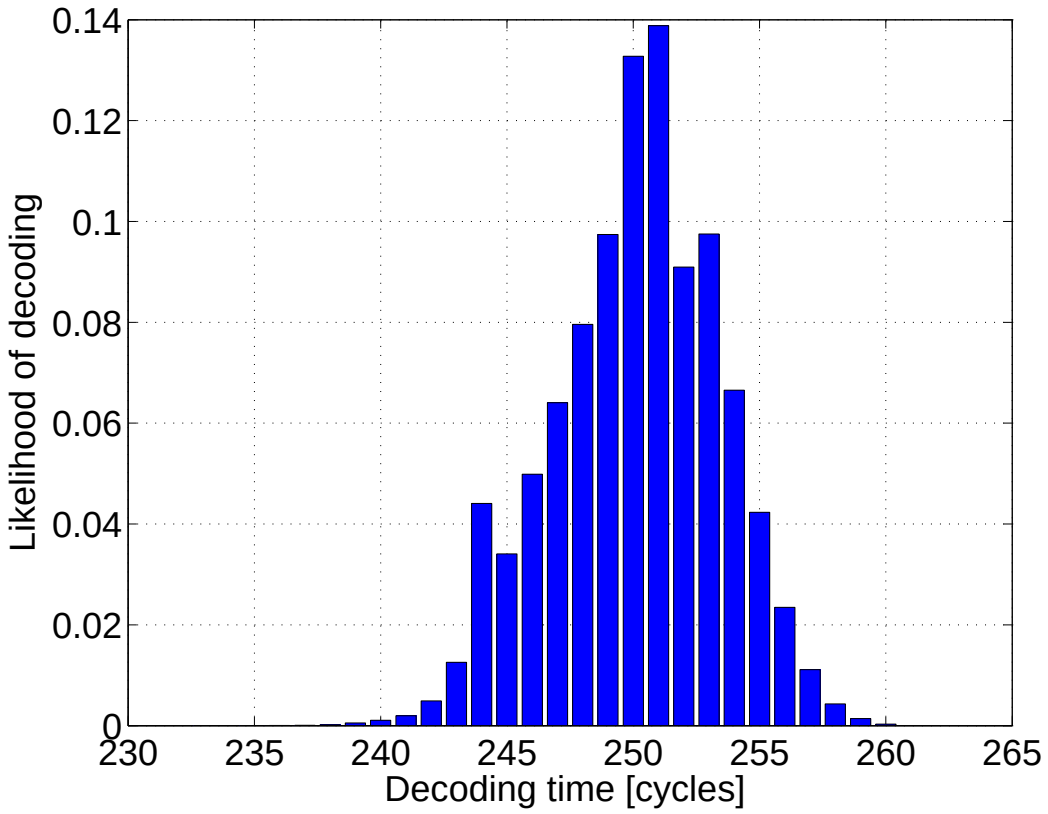

Figure 5.5: Distribution of decoding time (in clock cycles) for the implemented Raptor 10 decoder. In the experiment the decoder is clocked at 62.5 MHz.

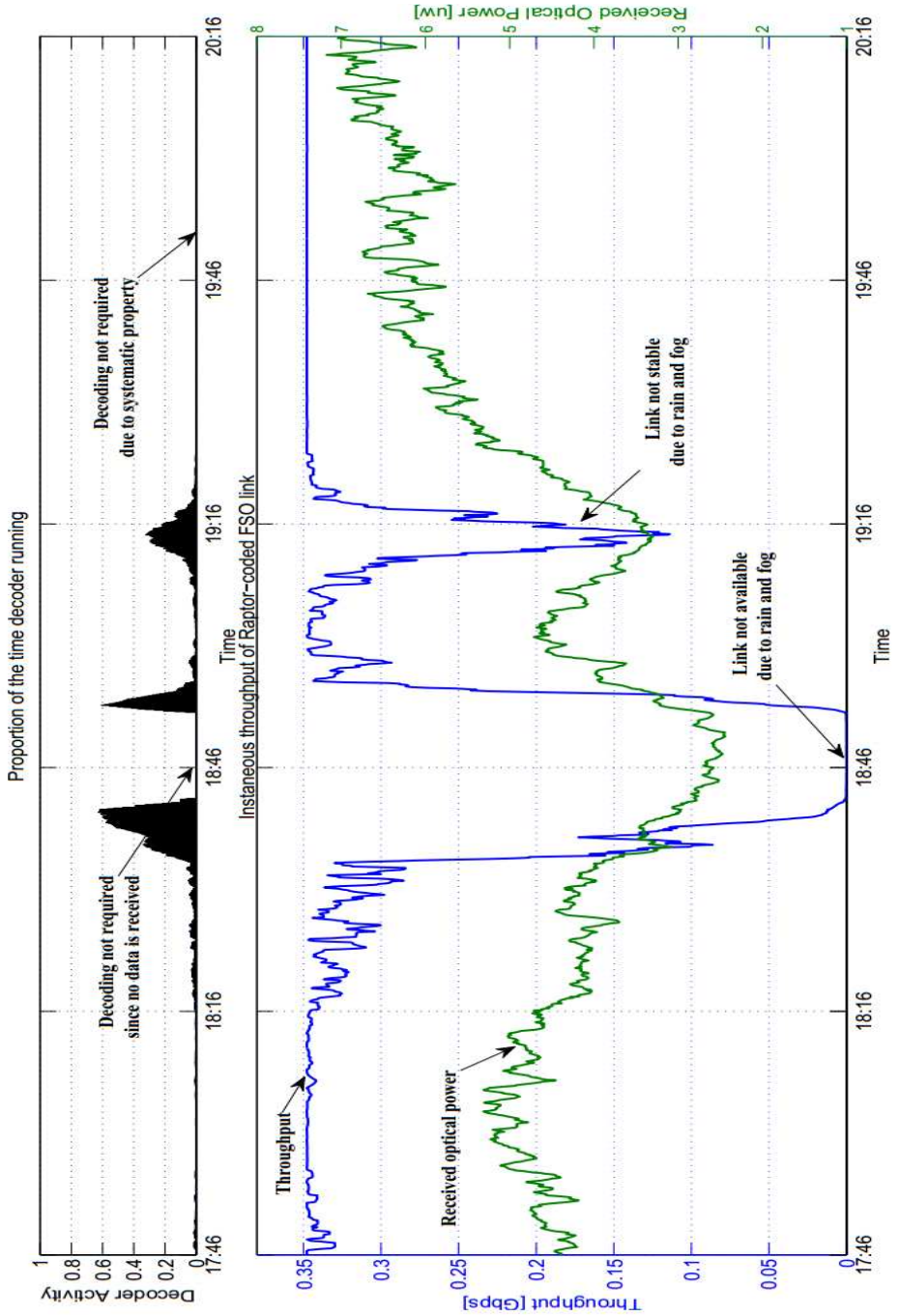

Figure 5.6: Instantaneous throughput of Raptor-coded FSO link and received optical power measured from 17:46 November 10, 2015 to 20:16 November 11, 2015.

Both the time to decode and the event of a decoding failure are recording in the hardware decoder. Note , the bottleneck of the throughput lies in the receiving speed at application level instead of the decoding speed.

Figure 5.6 presents the real-time throughput of the Raptor 10 coded FSO link along with the received optical power under practical weather conditions. In addition, the activity of the decoder is also plotted as indicated the proportion of the time decoder ran in a 5 sec window (i.e., the proportion of time all systematic source symbols are not received for a given transmission). The measurement experiment is conducted from 12:46 to 21:02 on November 10, 2015 with a continuous duration around 8 hours and only a 3 hour window is presented in the figure for clarity. During this period, the weather conditions were mixed with clear periods interspersed with periods of varying rain intensity.

The decoder can be observed to be operating in three regimes during the measurement period under consideration: link stable, link not stable and link not available.

The FSO link is stable under favourable weather conditions (i.e., clear or light rain) such as the time duration from 19:46 to 20:16. Notice that in this case the received optical power is high and packets are convey across the channel with little erasure. As evidenced by the low decoder activity, decoding is not required in a majority of message arrive using solely the systematic property of the Raptor 10 code. The link throughput at application layer is stable at 0.348 Gbps which is the limit limited due to protocol overhead.

When weather conditions are harsh during fog or heavy rain, such as the time duration around 18:46, the received optical power is low. In this case the link is not

available and the FSO link cannot support any data communication. In these cases, the decoder is also idle waiting to receive encoding symbols.

In intermediate weather conditions, such as from 19:10 to 19:30, with light to moderate rain conditions, the is termed to be not stable. The throughput varies greatly due to weather conditions which similarly impact the received optical power. In this regime, the systematic portion of the message is typically not received and the decoder is required to operate as evidenced by the high decoder activity. The rateless property of Raptor 10 enables the link to track the change of channel conditions, thus maximizing the channel utilization. Since decoding is initiated once a fixed number of symbols is received, there remains a finite probability of decoding failure. Using the fixed overhead of 16 symbols yielded a measured probability of decoding failure on the order of 10^{-5} during this period. A higher overhead could be applied to yield a lower decoding failure probability [82]. As shown in Fig. 5.6, the link throughput closely follows the trend of the received optical power. This indicates that Raptor-coded FSO link can only help maximize channel utilization and that Raptor-coded hybrid FSO/RF link are a viable the solution to availability issues of FSO communications.

5.5 Conclusions

This chapter presents the first in-field demonstration of an FPGA-based Raptor coded FSO link on a 1.87 km experimental FSO link. The work in this chapter focuses on the design and implementation of the receiver of the Raptor coded FSO link. The components of the receiver mainly consists of two functional modules, Raptor 10 decoder and UDP communication module. The Raptor 10 decoder is implemented based on the matrix inversion algorithm proposed in [70]. The Raptor 10 decoder is

currently capable of offering average decoding speed around 0.512 Gbps. The UDP communication module is implemented based on the Quixtream Gigabit Ethernet Protocol [69]. A pipelined structure of receiver, which is responsible for coordination between Raptor 10 decoder and UDP communication box, is proposed to maximize the real-time throughput of the link. Although the implemented link still cannot be used for commercial purpose, the preliminary measurement results do prove the important role of Raptor codes in FSO communications.

Chapter 6

Conclusions and Future Directions

The work in this thesis focuses on the application of the Raptor codes in FSO communications. It was shown in the Chapter 3 that Raptor codes help the FSO communication system closely follows the change of channel states and adapt the rate accordingly. Though FSO communications are attractive in a host of applications, such as last-mile connections with backbone networks, due to their high data rates, license-free operation, low interference and ease-of-deployment. However, a key limitation in the widespread deployment of FSO links is their inherent sensitivity to atmospheric effects and especially fog [35]. A promising direction to improve FSO reliability is to bond a complementary radio frequency (RF) channel to an FSO link to yield a *hybrid FSO/RF* link. Whereas FSO links are severely degraded by fog, RF links are able to maintain a persistent connection. Similarly in light rain, RF links are heavily attenuated while FSO links may still be operable. For the purpose of investigation of the performance of the hybrid FSO/RF system, a soft-switching hybrid FSO/LAN link using Raptor code is proposed and implemented using software. Details are presented in Chapter 4.

Limited by equipment resources, the implemented hybrid communication cannot be put into in practical application directly. However, the measurement results demonstrates that hybrid link is able to combine the bandwidths of both FSO and LAN links, thus maximizing link utilization. Additionally, the hybrid link eliminates availability issues of FSO links under adverse weather conditions.

Chapter 5 presents the first in-field demonstration of real-time Raptor-coded FSO communication over a 1.87 km urban link. This work serves as an important initial step toward FPGA-based implementation of Raptor-coded hybrid FSO/RF link. The hardware architecture of a Raptor 10 decoder based on a novel resource- and time-efficient matrix inversion algorithm is presented and implemented in an FPGA platform. The receiver is implemented in a pipelined architecture to maximize link utilization. The throughput of the Raptor-coded FSO link is measured both experimentally and numerically under real-life weather conditions for a continuous period of 8 hours. Experimental results demonstrate that Raptor-coded FSO links are capable of tracking changes in the channel state to maximize link utilization.

6.1 Future Work

This work focuses the experimental demonstration of Raptor codes in FSO communications, representing an important step form simulation study to practical application. Considering the fact that FSO links usually operate at a very high speed like several Gbps, the decoding performance of the Raptor decoder is critical to the overall performance of the FSO links or hybrid FSO/RF links. The FPGA-based hardware implementation of Raptor codes are worth exploring since

parallel processing capability of FPGA helps boost the coding performance. For the implementation of Raptor 10 decoder in Chapter 5, two possible directions can be explored to improve the decoding performance, the optimization of the matrix inversion algorithm in [70] and increasing operating frequency.

As a powerful extension of the Raptor 10 code, RaptorQ code embraces many improvements such as supporting larger source block, better overhead-failure curve and exceptional computation complexities. As indicated in the simulation work in Chapter 3, the RaptorQ code is more robust robust than the Raptor 10 code in FSO communications offering much better communication performance. It is worth the time and energy to pursue the hardware implementation of the RaptorQ code. The most challenging part of implementation of RaptorQ code stills lies in the implementation of the decoder. Note, the decoder of RaptorQ code operates over GF(256). The decoder can be implemented using the hardware architecture similar to the one proposed in Chapter 5.

As indicated in Chapter 4, the Raptor-coded hybrid FSO/RF link is an efficient and effective solution of the availability issues of FSO communications. Future work could consider the FPGA-based hardware implementation of a RaptorQ coded hybrid RF/FSO link. Performance evaluation of the hybrid communication system under different weather conditions are of great importance to the study of FSO communications. To make the FPGA program more extensible, the design of the communication component and RaptorQ code should be separated from each other.

Bibliography

[1] Google, "Google Map," [Available: https://www.google.ca/maps].

[2] TEK Microsystems Inc, "Triton V5 VXS Hardware Reference Manual," [Available: http://www.tekmicro.com].

[3] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[4] T. Mladenov, S. Nooshabadi, and K. Kim, "Implementation and evaluation of Raptor codes on embedded systems," *IEEE Transactions on Computers*, vol. 60, no. 12, pp. 1678–1691, 2011.

[5] Network Security Toolkit (NST) Wiki, "Gigabit Ethernet Using UDP," [Available: http://wiki.networksecuritytoolkit.org/nstwiki/index.php].

[6] A. Mostafa and S. Hranilovic, "Channel measurement and Markov modeling of an urban free-space optical link," *Journal of Optical Communications and Networking*, vol. 4, no. 10, pp. 836–846, 2012.

[7] FreeSpaceOptics.org , "History of Free Space optics," [Available: http://www.freespaceoptics.org/freespaceopticshome.html].

[8] A. G. Bell, "On the production and reproduction of sound by light," *American Journal of Science*, no. 118, pp. 305–324, 1880.

[9] fSONA Networks Corp., "Laser Guide," [Available: http://www.fsona.com/prod/SONAbeam_M.pdf].

[10] BridgeWave Communications, "BridgeWave Communications," [Available: http://www.bridgewave.com/].

[11] fSONA Networks Corp., "fSONA: Free Space Optical Networking Architecture," [Available: http://www.fsona.com/].

[12] LightPointe Communications, "Lightpointe Communications," [Available: http://www.lightpointe.com/about.html].

[13] WhaTech, "Telecommunications Mobile Network Wireless Market Research News," [Available: https://www.whatech.com/market-research/telecommunications].

[14] T. Song and P. Kam, "Background radiation cancellation for free-space optical communications with IM/DD," *arXiv preprint arXiv:1508.07390*, 2015.

[15] System Support Solutions, "Model Selector," [Available: http://systemsupportsolutions.com/model-selector/].

[16] System support solutions, "NSG10G CASCASE," [Available: http://www.northernstormdirect.com/model/ns10g/].

[17] G. Nykolak, G. Raybon, B. Mikkelsen, B. Brown, P. F. Szajowski, J. J. Auborn, and H. M. Presby, "160-Gb/s free-space transmission link," in *Information*

Technologies 2000. International Society for Optics and Photonics, 2001, pp. 11–13.

[18] J. Pan, M. Evans, T. Euler, H. Johnson, and F. DeNap, "Free-space optical communications: opportunities and challenges from a carrier's perspective," in *2002 Asia-Pacific Optical and Wireless Communications*. International Society for Optics and Photonics, 2002, pp. 58–72.

[19] LaserOptronics, "FSO network applications," [Available: http://www.laseroptronics.com/index.cfm/id/57-67.htm].

[20] Lightwave, "Gigabit-speed FSO network serves Istanbul's business district," [Available: http://www.lightwaveonline.com/articles/print/volume-21/issue-7/applications/case-by-case/gigabit-speed-fso-network-serves-istanbuls-business-district-53906792.html].

[21] H. Willebrand and B. S. Ghuman, *Free space optics: enabling optical connectivity in today's networks*. SAMS publishing, 2002.

[22] M. A. Khalighi and M. Uysal, "Survey on free-space optical communication: a communication theory perspective," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2231–2258, 2014.

[23] N. Perlot, M. Knapek, D. Giggenbach, J. Horwath, M. Brechtelsbauer, Y. Takayama, and T. Jono, "Results of the optical downlink experiment KIODO from OICETS satellite to optical ground station oberpfaffenhofen (OGS-OP)," in *Lasers and Applications in Science and Engineering*. International Society for Optics and Photonics, 2007, pp. 645 704–645 704.

[24] G. G. Ortiz, S. Lee, S. P. Monacos, M. W. Wright, and A. Biswas, "Design and development of robust ATP subsystem for the Altair UAV-to-ground lasercomm 2.5-Gbps demonstration," in *High-Power Lasers and Applications*. International Society for Optics and Photonics, 2003, pp. 103–114.

[25] K. Wakamori, K. Kazaura, and I. Oka, "Experiment on regional broadband network using free-space-optical communication systems," *Journal of Lightwave Technology*, vol. 25, no. 11, pp. 3265–3273, 2007.

[26] H. Hemmati, *Deep space optical communications*. John Wiley & Sons, 2006.

[27] I. I. Kim, B. McArthur, and E. J. Korevaar, "Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications," in *Information Technologies 2000*. International Society for Optics and Photonics, 2001, pp. 26–37.

[28] S. Bloom and W. S. Hartley, "The last-mile solution: hybrid FSO radio," *Whitepaper, AirFiber Inc*, 2002.

[29] S. S. Muhammad, B. Flecker, E. Leitgeb, and M. Gebhart, "Characterization of fog attenuation in terrestrial free space optical links," *Optical Engineering*, vol. 46, no. 6, pp. 066 001–066 001, 2007.

[30] M. Achour, "Simulating atmospheric free-space optical propagation: rainfall attenuation," in *High-Power Lasers and Applications*. International Society for Optics and Photonics, 2002, pp. 192–201.

[31] M. S. Awan, L. C. Horwath, S. S. Muhammad, E. Leitgeb, F. Nadeem, and

M. S. Khan, "Characterization of fog and snow attenuations for free-space optical propagation," *Journal of communications*, vol. 4, no. 8, pp. 533–545, 2009.

[32] M. Gebhart, E. Leitgeb, S. S. Muhammad, B. Flecker, C. Chlestil, M. A. Naboulsi, F. de Fornel, and H. Sizun, "Measurement of light attenuation in dense fog conditions for FSO applications," in *Optics and Photonics 2005*. International Society for Optics and Photonics, 2005, pp. 58 910K–58 910K.

[33] S. M. Navidpour, M. Uysal, and M. Kavehrad, "BER performance of free-space optical transmission with spatial diversity," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 2813–2819, 2007.

[34] D. K. Borah and D. G. Voelz, "Pointing error effects on free-space optical communication links in the presence of atmospheric turbulence," *Journal of Lightwave Technology*, vol. 27, no. 18, pp. 3965–3973, 2009.

[35] I. I. Kim and E. J. Korevaar, "Availability of free-space optics (FSO) and hybrid FSO/RF systems," in *ITCom 2001: International Symposium on the Convergence of IT and Communications*. International Society for Optics and Photonics, 2001, pp. 84–95.

[36] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 4. ACM, 1998, pp. 56–67.

[37] D. J. MacKay, "Fountain codes," *IEE Proceedings-Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.

[38] M. Luby, "LT codes," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 2002, pp. 271–271.

[39] O. Etesami and A. Shokrollahi, "Raptor codes on binary memoryless symmetric channels," *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 2033–2051, 2006.

[40] P. Pakzad and A. Shokrollahi, "Design principles for Raptor codes," in *Information Theory Workshop, 2006*. IEEE, 2006, pp. 165–169.

[41] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder, "Raptor forward error correction scheme for object delivery," RFC 5053 (Proposed Standard), Tech. Rep., 2007.

[42] M. Luby and A. Shokrollahi and M. Watson and T. Stockhammer and L. Minder, "RaptorQ Forward Error Correction Scheme for Object Delivery (RFC 6330)," *IETF Request For Comments*, 2011.

[43] Multimedia Broadcast, "Multicast Service (MBMS); Architecture and functional description (Release 8), 3GPP TS 23.246. v8. l. 0," 2007.

[44] ETSI TS 102 472 v1.2.1, "IP Datacast over DVB-H: Content Delivery Protocols," March 2006, Technical Specification. [Available: http://www.dvb-h.org].

[45] Qualcomm Inc., "Qualcomm Inc." [Available: https://www.qualcomm.com/].

[46] Qualcomm Inc., "RaptorQ forward error correction technology," [Available: https://www.qualcomm.com/products/raptorq/technology].

[47] R. Palanki and J. S. Yedidia, "Rateless codes on noisy channels," in *IEEE International Symposium on Information Theory*. Citeseer, 2004, pp. 37–37.

[48] K. Hu, J. Castura, and Y. Mao, "Performance-complexity tradeoffs of Raptor codes over Gaussian channels," *IEEE Communications letters*, vol. 11, no. 4, pp. 343–345, 2007.

[49] B. Sivasubramanian and H. Leib, "Fixed-rate raptor code performance over correlated rayleigh fading channels," in *Canadian Conference on Electrical and Computer Engineering, 2007*. IEEE, 2007, pp. 912–915.

[50] P. Cataldi, M. P. Shatarski, M. Grangetto, and M. Enrico, "Implementation and performance evaluation of LT and Raptor codes for multimedia applications," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP'06.* IEEE, 2006, pp. 263–266.

[51] T. Mladenov, K. Kim, and S. Nooshabadi, "Forward error correction with RaptorQ code on embedded systems," in *2011 IEEE 54th international midwest symposium on Circuits and systems (MWSCAS)*. IEEE, 2011, pp. 1–4.

[52] C. Bouras, N. Kanakis, V. Kokkinos, and A. Papazois, "Evaluating RraptorQ FEC over 3GPP multicast services," in *2012 8th International wireless communications and mobile computing conference (IWCMC)*. IEEE, 2012, pp. 257–262.

[53] W. Zhang and S. Hranilovic, "Short-length Raptor codes for mobile free-space optical channels," in *IEEE International Conference on Communications, 2009.* IEEE, 2009, pp. 1–5.

[54] L. Liu, A. Dang, and H. Guo, "Rateless codes for slant path free space optical channels," in *2010 IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE, 2010, pp. 985–989.

[55] A. Andò, S. Mangione, L. Curcio, S. Stivala, G. Garbo, R. Pernice, and A. Busacca, "Recovery capabilities of rateless codes on simulated turbulent terrestrial free-space optics channel model," *International Journal of Antennas and Propagation*, vol. 2013, 2013.

[56] J. A. Anguita, M. A. Neifeld, B. Hildner, and B. Vasic, "Rateless coding on experimental temporally correlated FSO channels," *Journal of Lightwave Technology*, vol. 28, no. 7, pp. 990–1002, 2010.

[57] L. Liu, M. Safari, and S. Hranilovic, "Systematic Raptor codes for atmospheric optical channels," in *2013 13th Canadian Workshop on Information Theory (CWIT)*. IEEE, 2013, pp. 190–194.

[58] R. Pernice, A. Ando, D. Musso, F. Lionti, A. Parisi, S. Mangione, G. Adamo, L. Curcio, S. Stivala, and A. C. Busacca, "Rateless codes mitigation technique in a turbulent indoor Free Space Optics link," in *2014 Third Mediterranean Photonics Conference*. IEEE, 2014, pp. 1–3.

[59] A. Andò, R. Pernice, L. Curcio, S. Stivala, S. Mangione, P. Gallo, M. Campanella, G. Garbo, and A. Busacca, "Fading mitigation coding techniques for space to ground free-space optical communications," in *Euro Med Telco Conference (EMTC), 2014*. IEEE, 2014, pp. 1–5.

[60] W. Zhang, "Short-length Raptor codes for Free-space optical communications,"

2009, Master Thesis. [Available: https://macsphere.mcmaster.ca/bitstream/11375/8913/1/fulltext.pdf].

[61] F. Nadeem, B. Geiger, E. Leitgeb, M. S. Awan, and G. Kandus, "Evaluation of switch-over algorithms for hybrid FSO-WLAN systems," in *1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009.* IEEE, 2009, pp. 565–570.

[62] F. Nadeem, V. Kvicera, M. S. Awan, E. Leitgeb, S. Muhammad, and G. Kandus, "Weather effects on hybrid FSO/RF communication link," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 9, pp. 1687–1697, 2009.

[63] A. Akbulut, H. G. Ilk, and F. Ari, "Design, availability and reliability analysis on an experimental outdoor FSO/RF communication system," in *Transparent Optical Networks, 2005, Proceedings of 2005 7th International Conference*, vol. 1. IEEE, 2005, pp. 403–406.

[64] H. Wu and M. Kavehrad, "Availability evaluation of ground-to-air hybrid FSO/RF links," *International Journal of Wireless Information Networks*, vol. 14, no. 1, pp. 33–45, 2007.

[65] M. Usman, H. Yang, and M. Alouini, "Practical switching-based hybrid FSO/RF transmission and its performance analysis," *IEEE Photonics Journal*, vol. 6, no. 5, pp. 1–13, 2014.

[66] W. Zhang, S. Hranilovic, and C. Shi, "Soft-switching hybrid FSO/RF links using

short-length Raptor codes: design and implementation," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 9, pp. 1698–1708, 2009.

[67] A. AbdulHussein, T. T. Nguyen, and L. Lampe, "Rateless coding for hybrid free-space optical and radio-frequency communication," *IEEE Transactions on Wireless Communications*, vol. 9, no. 3, pp. 907–913, 2010.

[68] T. Kamalakis, I. Neokosmidis, A. Tsipouras, S. Pantazis, and I. Andrikopoulos, "Hybrid free space optical/millimeter wave outdoor links for broadband wireless access networks," in *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007.* IEEE, 2007, pp. 1–5.

[69] TEK Microsystems Inc, "Quixtream Gigabit Ethernet UDP/IP User Manual version 1.20.2," [Available: http://www.tekmicro.com].

[70] L. Liu, "Rate-adaptive techniques for free-space optical channels," 2013, Master Thesis. [Available: https://macsphere.mcmaster.ca/bitstream/11375/13344/1/fulltext.pdf].

[71] A. Bogdanov, M. Mertens, C. Paar, J. Pelzl, and A. Rupp, "SMITH-A parallel hardware architecture for fast Gaussian elimination over GF(2)," in *Workshop on Special-purpose Hardware for Attacking Cryptographic Systems (SHARCS 2006), Conference Records*, 2006.

[72] R. P. Ricardo, V. A. Pedroni, A. Gortan, and W. Godoy, "An Improved GF(2) matrix inverter with linear time complexity," in *2010 International Conference on Reconfigurable Computing and FPGAs (ReConFig).* IEEE, 2010, pp. 322–327.

[73] M. Lu and L. Liu and S. Hranilovic, "Raptor-coded free-space optical communicaiton experiment," submitted to Journal of Optical Communications and Networking.

[74] fSONA Networks Corp., "SONAbeam1250-M datasheet," [Available: http://www.fsona.com/technology.php?sec=laser_guide].

[75] fSONA Networks Corp., "Sonabeam M series," [Available: http://www.fsona.com/prod/SONAbeam_M.pdf].

[76] fSONA Networks Corp., "Engineering Specification - Custom 1250-M for McMaster University, ENGSPEC-90-0272."

[77] TP-LINK, "Gigabit SFP Media Converter MC220L," [Available: http://www.tp-link.com/en/products/details/cat-4792_MC220L.html].

[78] TEK Microsystems Inc , "Triton-V5 datasheet," [Available: http://http://www.tekmicro.com/PDFs/Tekmicro_Triton_2channel_ADC_DAC.2010.07.pdf].

[79] Xilinx, Inc. , "Virtex-5 Family Overview," Available:http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf .

[80] T. Mladenov, S. Nooshabadi, J. A. Montiel-Nelson, and K. Kim, "Decoding of Raptor codes on embedded systems," *Microprocessors and Microsystems*, vol. 36, no. 5, pp. 375–382, 2012.

[81] T. Mladenov, S. Nooshabadi, and K. Kim, "Efficient GF(256) Raptor code decoding for multimedia broadcast/multicast services and consumer terminals," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 356–363, 2012.

[82] A. Shokrollahi and M. Luby, "Raptor Codes," *Foundations and trends in communications and information theory. Now Publishers Inc*, vol. 6, no. 3-4, 2011.

[83] Oracle Corporation, "Internet Protocol Suite," [Available: https://docs.oracle.com/cd/E19455-01/806-0916/6ja85398m/index.html].

[84] Cisco Systems, Inc., "Introduction to IP version 6," [Available: http://www.cisco.com/c/en/us/tech/ip/ip-version-6-ipv6/index.html].

[85] A. A. Farid and S. Hranilovic, "Outage capacity optimization for free-space optical links with pointing errors," *Journal of lightwave technology*, vol. 25, no. 7, pp. 1702–1710, 2007.

[86] Oracle Corporation, "All about sockets," [Available: https://docs.oracle.com/javase/tutorial/networking/sockets/].

[87] Microsoft Corporation , "Windows XP," [Available: http://windows.microsoft.com/en-ca/windows/xp-downloads#2TC=featured].

[88] I. Eclipse Foundation, "Eclipse IDE for Java EE Developers," [Available: https://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/lunasr2].

[89] Oracle Corporation, "Java SE Version 6 Update 26," [Available: http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html#jdk-6u26-oth-JPR].

[90] T. Mladenov, S. Nooshabadi, and K. Kim, "Strategies for the design of Raptor decoding in broadcast/multicast delivery systems," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 423–428, 2010.

[91] Microsoft Corporation, "Networking overview," [Available: https://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/taskman_networking_overview.mspx?mfr=true].

[92] J. Postel, "User datagram protocol," August 1980, RFC 768. [Available: ftp://ftp6.de.freebsd.org/pub/doc/rfc/pdfrfc/rfc768.txt.pdf].