“VISIBLE LIGHT COMMUNICATION”

SUMMER TRAINING REPORT

submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
ELECTRONICS & COMMUNICATION ENGINEERING

Submitted By
BHAWANA CHHAGLANI (02011502816)

Submitted To
MS. MONICA BHUTANI

DEPARTMENT OF ELECTRONICS COMMUNICATION ENGINEERING
BHARATI VIDYAPEETH’S COLLEGE OF ENGINEERING
(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI)
NEW DELHI - 110063

AUGUST 2018

Acknowledgements

I have taken efforts in this training. However, it would not have been possible without the kind support and help of many individuals and the organization (IIT Delhi, E.E. Dept and Bharti School Of Telecommunications). I would like to extend my sincere thanks to all of them.

I would like to express my heartfelt gratitude towards **Ms. Monica Bhutani**, Department of Electronics and Communication Engineering, BVCOE, for organizing and mentoring this summer training session on Visible Light Communication.

I wish to express my profound and sincere gratitude to **Dr. Brejesh Lall**, Associate Professor, E.E. Dept., IIT Delhi, who guided me into the intricacies of this project non-chalantly with matchless magnanimity.

I sincerely thank our instructors **Mr. Shivam Shekhar** for their profound guidance and advice during the course of this training. I would also like to appreciate the help and cooperation of my fellow teammates.

Words of gratitude are not enough to describe the accommodation and fortitude which they have shown throughout my endeavor.

BHAWANA CHHAGLANI

Contents

List of Figures

Chapter 1

Literature Review

Visible light communication systems (VLC) is a subset of optical wireless communications. It offers a replacement to the existing standards of wireless communication, through light from light-emitting diodes (LEDs) as the mean of communication. As LEDs twinkle repeatedly at a high speed such that human eye cannot perceive changes in light intensity, but a photodiode can. Thus, an illumination source can now send information using the same light source.

Optical Wireless communication (OWC) is a general term which refers to all types of optical communications where cables (optical fibres) are not used. VLC, FSO, Li-Fi and infra-red remote controls are all examples of OWC.

Free Space Optical (FSO) communication is similar to VLC but is not constrained to visible light, so ultraviolet (UV) and infrared (IR) also fall into the FSO category. Additionally, there is no illumination requirement for FSO and so this tends to be used in narrow beams of focussed light for applications such as communication links between buildings. FSO often uses laser diodes rather than LEDs for the transmission.

Lifi is a term often used to describe high speed VLC in application scenarios where Wi-Fi might also be used. The term Li-Fi is similar to Wi-Fi with the exception that light rather than radio is used for transmission.

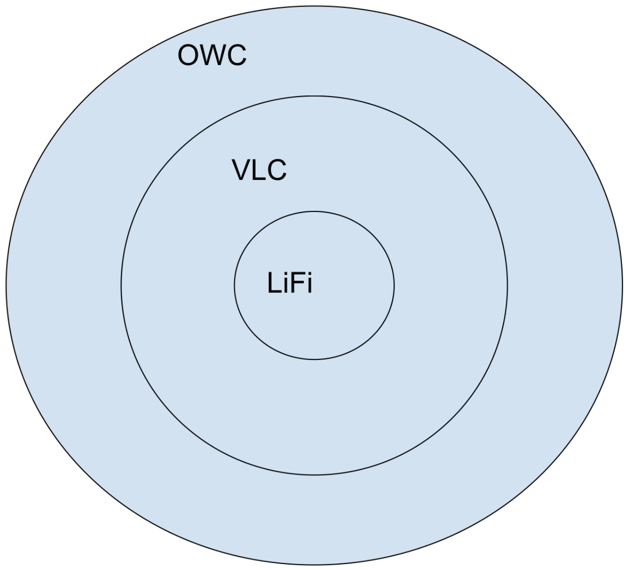

Figure 1.1: Lifi

1.1 History and Motivation

The concept of LiFi(Light Fidelity) was first seen in Photophone invented by Alexander Graham Bell. Photophone was a a wireless telephone of sorts that enabled sounds (including speech) to be transmitted via light. ItâĂŹs said that Bell actually valued the photophone more than the telephone from his lifetime achievements, even though it never quite took off and is now consigned to the footnotes of history. However, more than 130 years after the photophone first came to light, Harald Haas coined the term LiFi in TED talk in 2011.

Use of VLC for data transmission is having many benefits. VLC eliminates drawbacks of transmission via electromagnetic waves outside the visible spectrum. VLC may causes health problems because of exposure to moderation is assumed to be safe on the human body. Moreover, there is no interference with electromagnetic radiation occurs so that visible light can be used in hospitals and other institutions without problem.

1.2 Architecture

Basic architecture of LiFi includes led and photodiode.

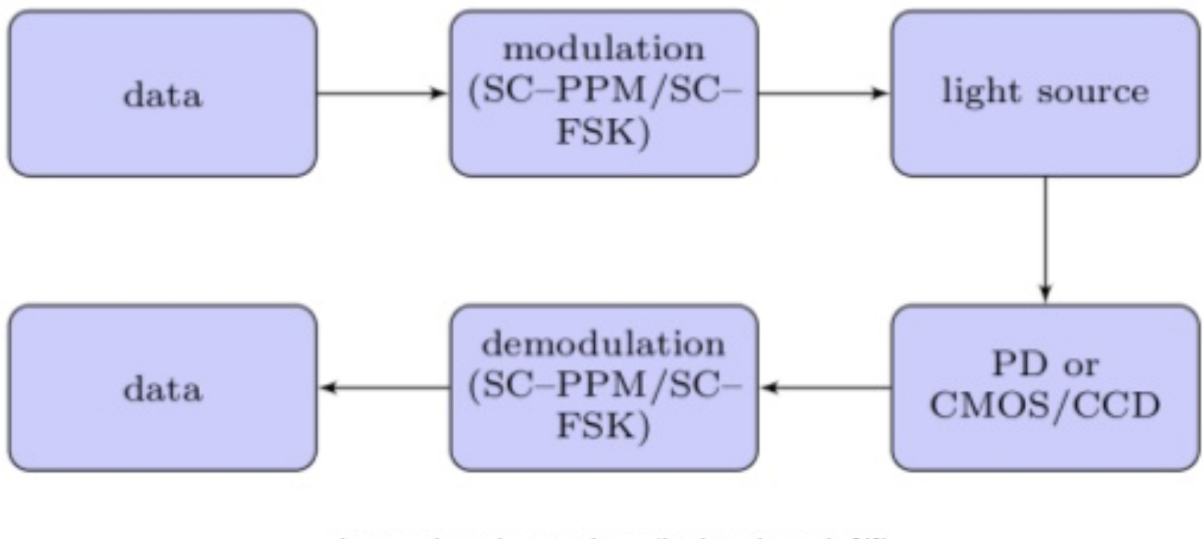

Figure 1.2: Basic Lifi Architecture

There are many other receivers for LiFi apart from photodiode:
• LDR(Light Dependent Resistor)
• Camera
• Image Sensor
• Solar Panel
• LED(Light Emitting Diode)

1.2.1 Modulation

The modulation signals is used to switch LED at desired frequencies that contains information to be transmitted. Modulation techniques is needed in order the communication is still available even the illumination is not required. Because of that, a modulation technique may support a dimmable illumination. The variation in intensity of light corresponding to the information in the message signal. There are many typical of modulation in LiFi i.e. Single Carrier Modulation (SCM), Multiple Carrier Modulation (MCM), and Colour Modulation.

1. **Single Carrier Modulation (SCM) -** SCM is a modulation multiplexed any number of signal, then modulated individually with different frequency. SCM using a microwave as a subcarrier and an optical carrier. Modulation technique in SCM are on-off keying (OOK), pulse-position modulation (PPM), and pulse-amplitude modulation (PAM). SCM is suitable for low-to- moderate data rates applications.

2. **Multi- Carrier Modulation (MCM) -** Because of the performance of SCM degrades as their spectral efficiency increase. SCM also require complex

equalization process when employed at high data rates. MCM is developed to replace the disadvantage of SCM. There are several kinds of MCM i.e. OFDM. **OFDM -** Orthogonal Frequency Division Multiplexing, is a form of signal modulation that divides a high data rate modulating stream placing them onto many slowly modulated narrowband close-spaced subcarriers, and in this way is less sensitive to frequency selective fading. OFDM is a form of multicarrier modulation. An OFDM signal consists of a number of closely spaced modulated carriers. When modulation of any form - voice, data, etc. is applied to a carrier, then sidebands spread out either side. It is necessary for a receiver to be able to receive the whole signal to be able to successfully demodulate the data. As a result when signals are transmitted close to one another they must be spaced so that the receiver can separate them using a filter and there must be a guard band between them. This is not the case with OFDM. Although the sidebands from each carrier overlap, they can still be received without the interference that might be expected because they are orthogonal to each another. This is achieved by having the carrier spacing equal to the reciprocal of the symbol period.

The OFDM scheme differs from traditional FDM in the following inter-related ways:

(a) Multiple carriers (called subcarriers) carry the information stream,

(b) The subcarriers are orthogonal to each other, and

(c) A guard interval is added to each symbol to minimize the channel delay spread and intersymbol interference.

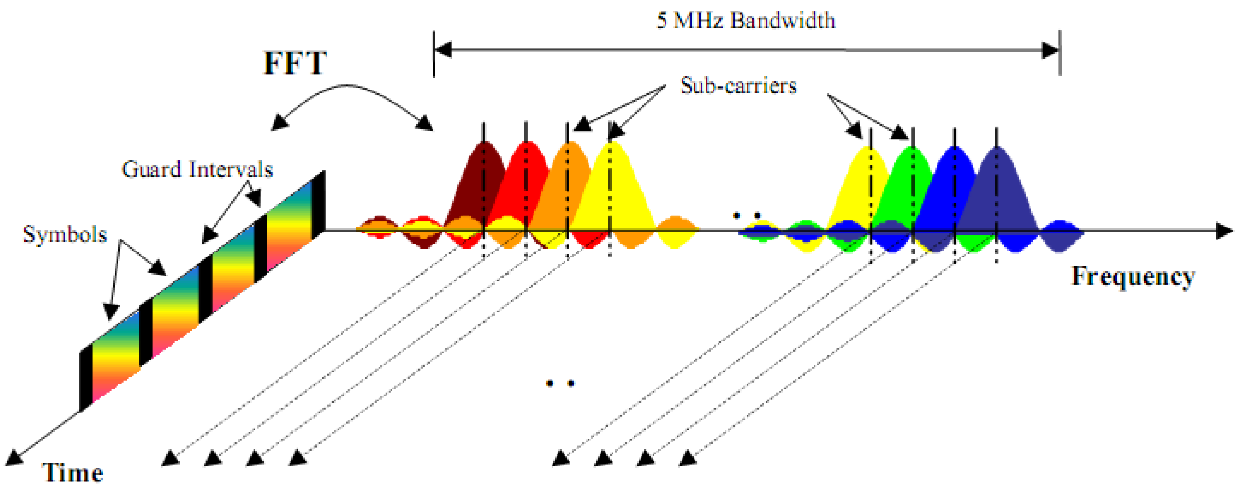

Figure 1.3: Orthogonal Frequency Multiple Division

OFDM Advantages

OFDM has been used in many high data rate wireless systems because of the many advantages it provides.

•***Immunity to selective fading:*** One of the main advantages of OFDM is that is more resistant to frequency selective fading than single carrier systems because it divides the overall channel into multiple narrowband signals that are affected individually as flat fading sub-channels.

•***Resilience to interference:*** Interference appearing on a channel may be bandwidth limited and in this way will not affect all the sub-channels. This means that not all the data is lost.

•***Spectrum efficiency:*** Using close-spaced overlapping sub-carriers, a significant OFDM advantage is that it makes efficient use of the available spectrum.

•***Resilient to ISI:*** Another advantage of OFDM is that it is very resilient to inter-symbol and inter-frame interference. This results from the low data rate on each of the sub-channels.

•***Resilient to narrow-band effects:*** Using adequate channel coding and interleaving it is possible to recover symbols lost due to the frequency selectivity of the channel and narrow band interference. Not all the data is lost.

•***Simpler channel equalisation:*** One of the issues with CDMA systems was the complexity of the channel equalisation which had to be applied across the whole channel. An advantage of OFDM is that using multiple sub-channels, the channel equalization becomes much simpler.

DIFFERENCE BETWEEN SCM and MCM

Modulation			Characteristic
SCM	OOK		Transmit data by sequentially turning on and off the LED, the reliable communication range would decrease at low dimming levels, and increasing and decreasing the brighteness of the LED would cause the data rate to decrease
	PPM		Provide efficient and dimming support in variable pulse position modulation (VPPM)
	PAM		Sensitive to signal distortion[8], combination with other modulation technique to get a better performance
MCM	OFDM		Excellent for situations where multiple transmitters are used simultaneously, avoid shadowing effects, the interference can be mitigated by shifting the system bandwidth to higher frequency[9]
		DCO-OFDM	The substantial energy dissipation due to the biasing [10]
		ACO-OFDM	Efficient in term of optical power for lower SNR value for IM/DD channel [11]
		PAM-DMT	Better optical power efficiency compared to DCO-OFDM [12]
		AHO-OFDM	Support various dimming targets to achieve system performance [12]
		Flip-OFDM	Equivalent with the ACO-OFDM in term of spectral efficiency and error performance[13]
		U-OFDM	Equivalent with the ACO-OFDM in term of spectral efficiency and error performance[13]

Figure 1.4: Different techniques for Lifi Modulation

According to the standard of VLC IEEE 802.15.7, the architecture of VLC consists of mainly two layers :

1. Physical Layer
2. MAC layer (Medium Access Control)

Physical layer consists of :

1. PHY-I (11.67-266.6 Kbps) - PHY I has been optimised for low rate, long distance, outdoor applications such as vehicles and traffic lights.
2. PHY-II (1.25-96 Mbps) - PHY II has been optimised for high rate, indoor infrastructure and point-to-point applications.
3. PHY-III(12-96 Mbps) - PHY III has been optimised for indoor point-to-point applications, which support multiple LEDs to produce white light.

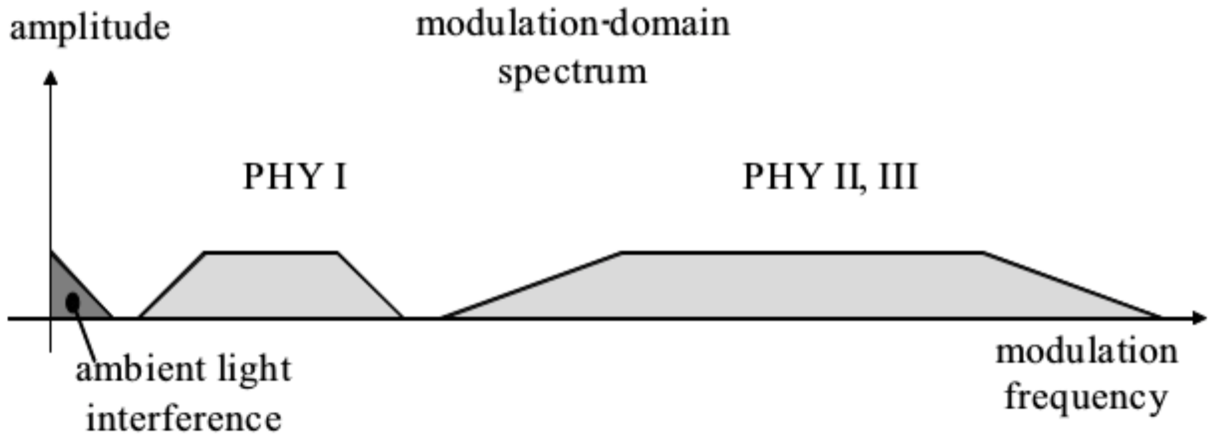

Figure 1.5: FDM separation of the PHY types in the modulation domain

THREE PHYSICAL LAYERS IN IEEE 802.15.7 STANDARD

PHY Types	Modulation	Clock Rates	Data Rates
PHY-I	OOK & VPPM	200/400 kHz	11.67–266.6 kb/s
PHY-II	OOK & VPPM	≤120 MHz	1.25–96 Mb/s
PHY-III	CSK	12/24 MHz	12–96 Mb/s

Figure 1.6: Physical layers in VLC

MAC layer is responsible for performing the following functions:

- Generating network beacons if the device is a coordinator
- Synchronizing to network beacons
- Supporting VPAN association and disassociation
- Supporting color function
- Supporting visibility
- Supporting dimming
- Flicker-mitigation scheme
- Supporting visual indication of device status and channel quality
- Supporting device security
- Providing a reliable link between two peer MAC entities
- Supporting mobility

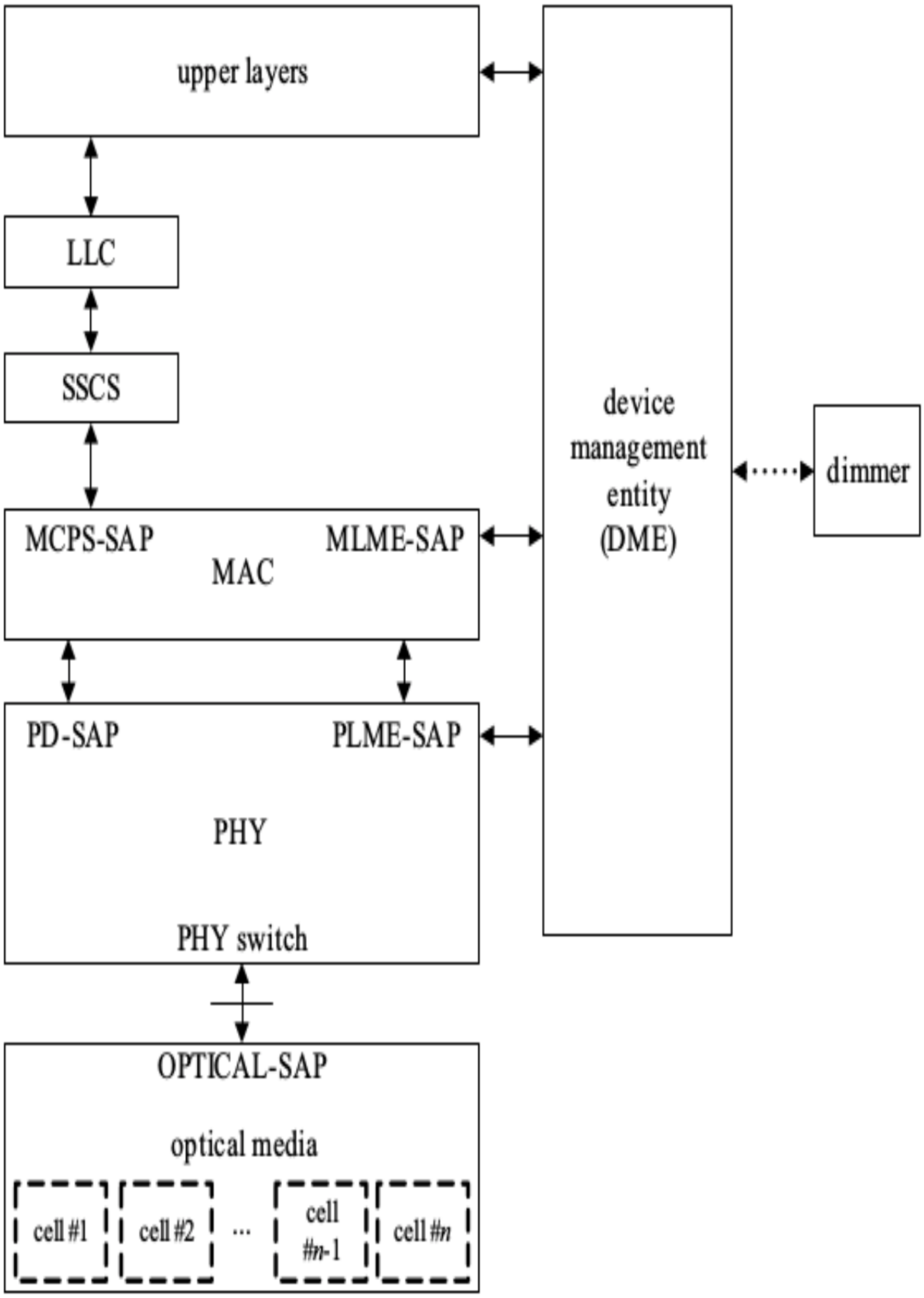

Figure 1.7: VPAN device architecture

MAC Topologies supported :

- Peer to Peer
- Broadcast
- Star

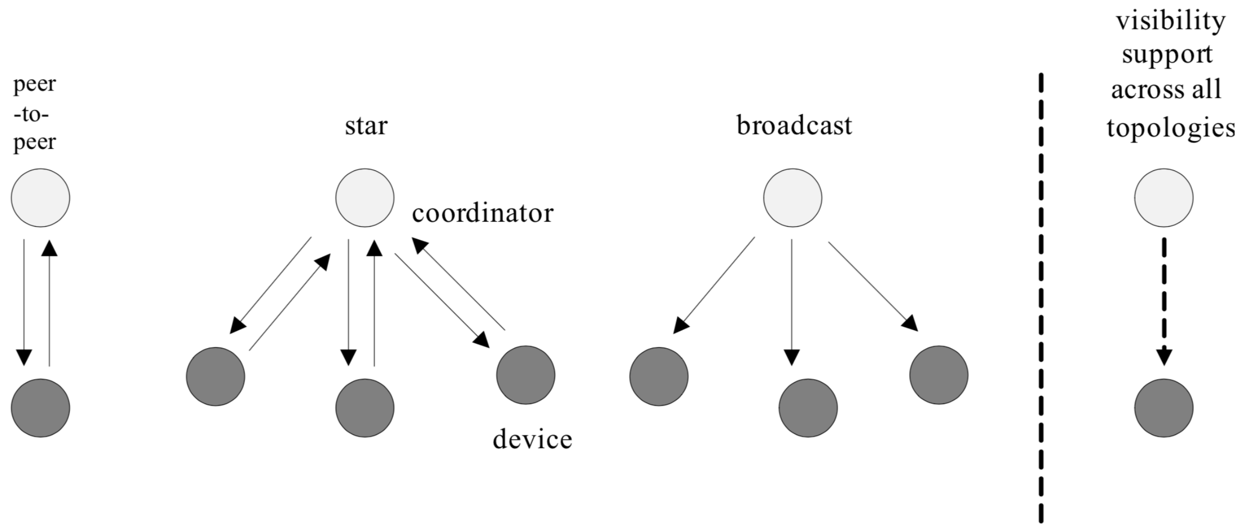

Figure 1.8: Supported MAC topologies

Various modulation schemes available in LiFi are:
• OOK(On-Off Keying)
• VPPM(Variable Pulse Position Modulation)
• CSK(Color Shift Keying)
Amplitude and Phase modulation cannot be used in LiFi.

1. **OOK Modulation** - In this modulation, data is represented by on and off of the LED. In simple terms, light 'ON state' represents logic '1' and light 'OFF state' represents logic '0'. In LiFi standard, OOK uses manchester codes to represent digital information in the form of 1's and 0's. Here encoding is done using edge transition concept where in low to high represent logic '1' and high to low transition represent logic '0'. OOK modulation technique under dimming helps to achieve variable data rate and constant range. This is achieved using insertion of compensation time.

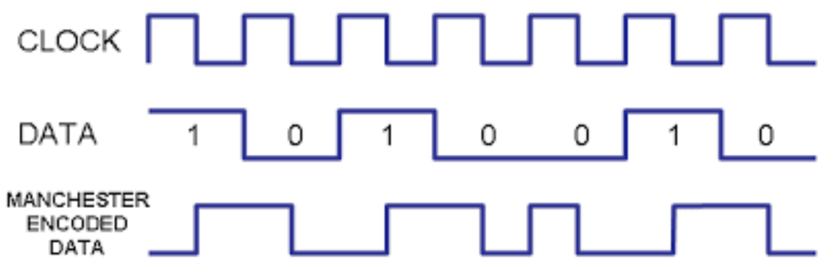

Figure 1.9: OOK Modulation

OOK Dimming - During dimming, the OOK modulated light intensity must decrease. This can be achieved by either redefining the "on" and "off" levels of the OOK symbol to have a lower intensity, or the levels can remain the same and the average duty cycle of the waveform can be changed by the insertion of "compensation" time into the modulation waveform.

Data Transmission Time	Compensation Time	Data Transmission Time	Compensation Time

Figure 1.10: OOK Diming

2. **VPPM Modulation** - This modulation technique is similar to PPM technique but here pulse width is variable for light dimming control application. VPPM modulation technique under dimming helps to achieve variable range and constant data rate. This is achieved by adjusting pulse width. The advantage of this modulation type is that it protects from intra frame flicker. This is due to the fact that pulse amplitude is held constant and dimming is controlled by variable is pulse width.

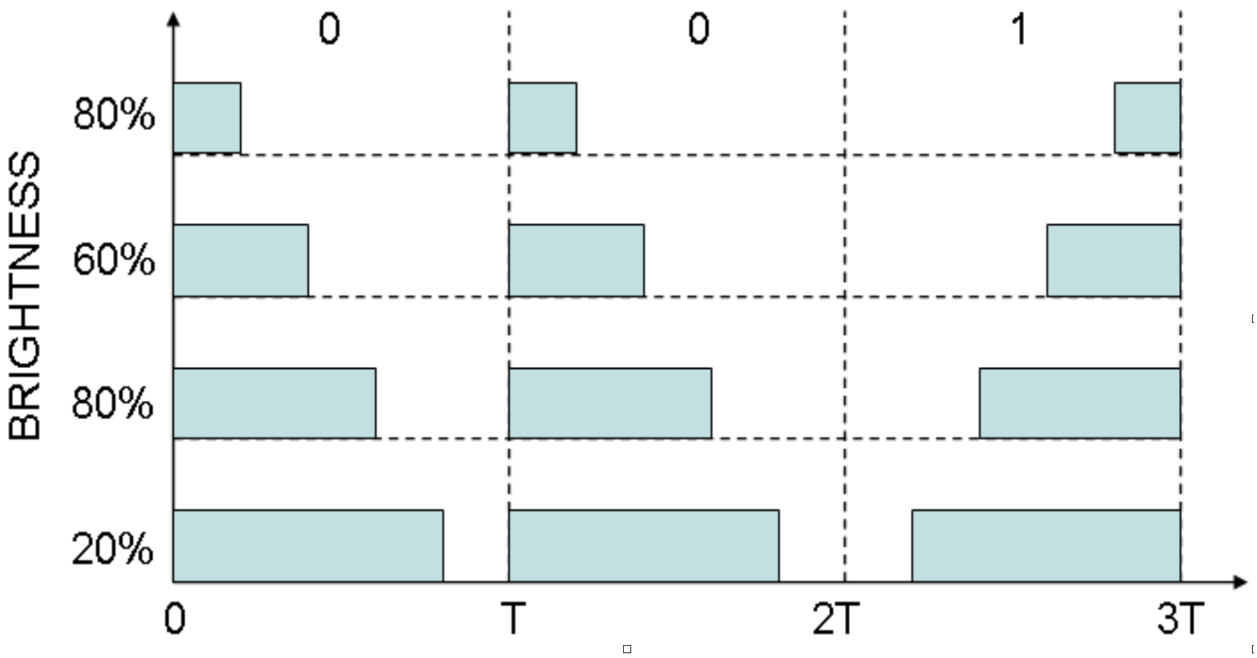

Figure 1.11: VPPM Modulation

VPPM Dimming - Dimming and full brightness in VPPM is achieved by controlling the âĂIJONâĂİ time pulse width.

3. **CSK Modulation** - The CSK modulation scheme is used to represent information bits in the form of different color wavelengths. Red LED, Green LED and Blue LED are used at the transmit end to produce different colors of different

wavelengths to code the information bits.

Characteristics of CSK:-

(a) Connectivity is guaranteed by the color coordinates: CSK channels are decided by mixed colors that are allocated in the color coordinates plane; therefore, the connectivity is guaranteed by the color constellation on the xy color coordinates.

(b) Total power is constant: the total power of all CSK light sources is constant because the envelope of the sum of all light signals is constant.

(c) Variable bit rate: CSK enables variable bit rate due to higher order modulation support; that is, it supports multiple bits per CSK symbol.

CSK Dimming - CSK dimming employs amplitude dimming and controls the brightness by changing the current driving the light source.

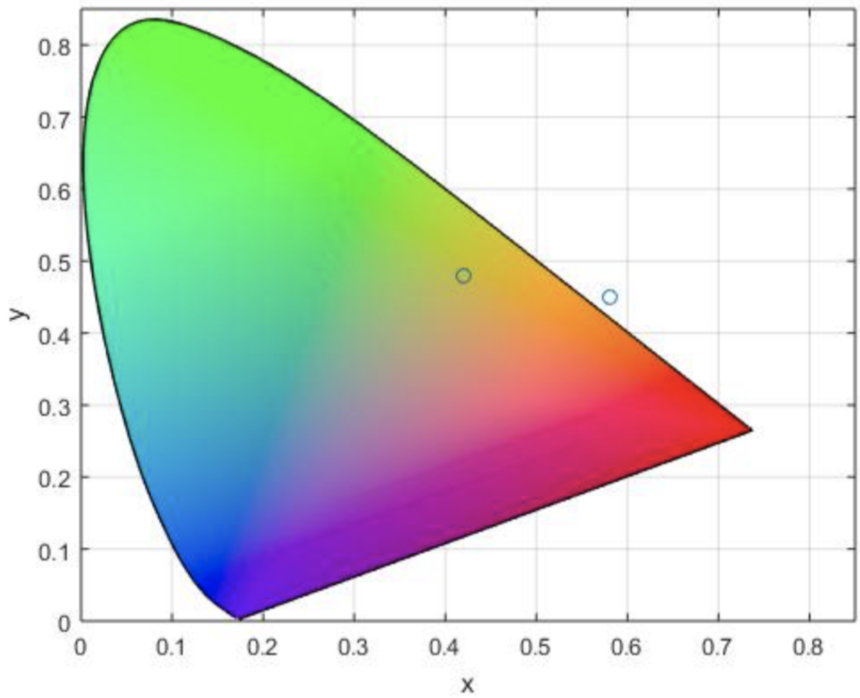

Figure 1.12: CSK Modulation

FLICKER MITIGATION SUPPORT

The flicker in visible light communications is classified into two categories according to its generation mechanism: intra-frame flicker and inter-frame flicker. Intra-frame flicker is defined as the perceivable brightness fluctuation within a frame. Inter-frame flicker is defined as the perceivable brightness fluctuation between adjacent frame transmissions.

1. **Intra-frame flicker mitigation -** Intra-frame flicker mitigation is accomplished by either the use of run length limiting coding, modulation scheme, or both.
2. **Inter-frame flicker mitigation -** The scheme used for interframe flicker mitigation is the transmission of an idle pattern between data frames whose average brightness is equal to that of the data frames.

Flicker mitigation	Data transmission (Intra-frame flicker)	Idle or RX periods (Inter-frame flicker)
OOK modulation	Dimmed OOK mode, RLL code	Idle/visibility patterns
VPPM modulation	VPPM guarantees no intra-frame flicker, RLL code	
CSK modulation	Constant average power across multiple light sources, scrambler, high optical clock rates (MHz)	

Figure 1.13: Flicker mitigation support

1.3 Advantages

•The infrastructure required for VLC is already available. LiFi can be implemented into existing lighting infrastructure with the addition of a few relatively simple and low-cost front-end components operating in baseband.
•Visible light spectrum is unlicensed and currently largely unused for communications. Moreover, potential bandwidth of visible light (âĹij 400 THz to âĹij 780 THz) is thousand times wider than the conventional RF bandwidth (3 kHz to 300 GHz).
•LED is having advanced properties such as high brightness, reliability, lower power consumption and long lifetime.
•LiFi offers more security than WiFi. RF waves pass through walls and are susceptible to eavesdropping whereas light waves do not pass through walls.
•LiFi is faster than WiFi.

Comparison between WiFi and LiFi:-

Property	WiFi	LiFi
Speed	150Mbps	>1Gbps
Carrier	Radio Spectrum	Light
Spectrum Range	30Hz- 300 GHz	430-770 THz
Operating Frequency	2.4GHz	Hundreds of TeraHertz
License	License Required	It is free.
Security	Less Secure	More secure

Figure 1.14: Wifi Vs. Lifi

1.4 Applications

The rapid advancements in the field of VLC are opening doors for various applications in many areas. Some of them are briefly listed below:-

•IoT(Internet of things)
•Indoor Positioning
•Vehicle to Vehicle communication
•Vehicle to Infrastructure communication
•Underwater Communication
•Hospitals
•Aircrafts
•Petrochemical Plants

1.5 Challenges

•Providing an efficient uplink scheme for LiFi (from photo diode to LED luminaire) has been challenging, as LiFi with illumination has predominantly broadcast characteristics. Therefore, IR(Infrared) is used for uplink.
•The transmission distance of visible light sources is limited and requires LOS and best SNR conditions to achieve high data rates. With an object or human blocking the LOS, the observed optical power degrades resulting in severe data rate reductions.
•Other artificial and natural light sources create interference and act as unmodulated sources at the receiver. In presence of ambient light, SNR decreases dramatically and thus data accuracy decreases.
•Mobility introduces various issues in LiFi.

Chapter 2

Related Work

There has been a lot of research going on in developing MAC layer for visible light communication on Python, Matlab and NS3.

2.1 Python

A successful demonstration of MAC layer developed on Python for peer to peer network has been shown[1]. The designed MAC Layer was adapted from IEEE 802.15.7 standard, specifically for delivering and receiving the MAC frames from the coordinator to the device or vice versa, wrapping the IP packets and unwrapping the MAC Frames with MAC headers and trailers, and performing the error detection in which the cyclic redundant check 16 (CRC-16) was selected. It was developed to support the multimedia contents (video and image) as well as data packet for internet accessing purpose like a wireless fidelity (Wi-Fi) connectivity. The python code consisted of two functions namely MACwrap and MACunwrap. The MAC layer has been realized in Python programming and running above the Zynq-7000 processor.

2.2 Matlab

The implementation of MAC layer on MATLAB has been successfully done and various results related to throughput, delay and performance has been studied[2]. This paper analyzes the impact of superframe size on the network performance in terms of network throughput and delay. Because there is a trade-off between throughput and delay, a new measurement for evaluating the overall system performance was used i.e. performance.

$$Performance = e^{throughput} \div log(delay)$$

The aim of the paper was to find the optimum value of SO(Superframe Order) in order to maximize the performance. It was concluded that to achieve maximum performance the value of SO should be kept 7.

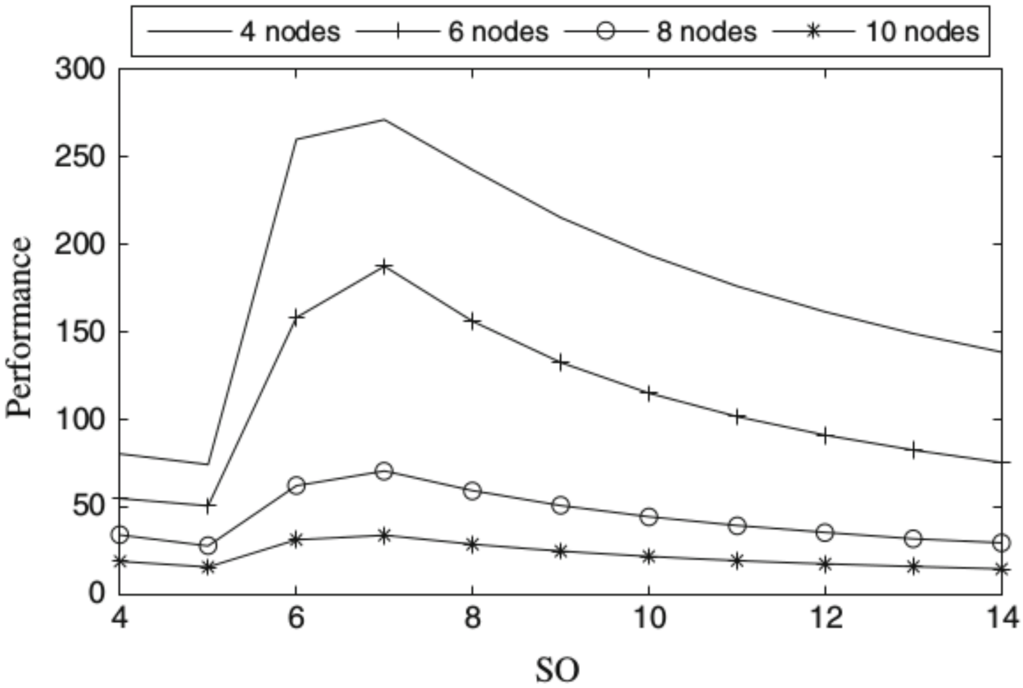

Figure 2.1: Performance VS Superframe Order

There is built-in library for zigbee in MATLAB according to IEEE802.15.4[3] standard[4]. It consists of several functions for physical, MAC and network layer. The MAC layer functions include MAC frame generator, decoder and CSMA/CA functions. These were studied and referred.

2.3 NS2

The paper[5] suggested improvements in the IEEE 802.15.4 NS-2 module to provide a better support for the emulation of networks with real-time requirements, through the incorporation of the contention free period (CFP) and of guaranteed time slot (GTS) defined within the IEEE 802.15.4 module present in the NS-2. The IEEE 802.15.4 module implemented in NS-2 is modified and extended to include the use of the GTS mechanisms based on the standard. So, the operations of the GTS allocation, use and deallocation are implemented. The addition of unimplemented MAC operations enhanced the simulation module so that is in accordance to the standard. This paper compared the latency of CAP Vs CFP:-

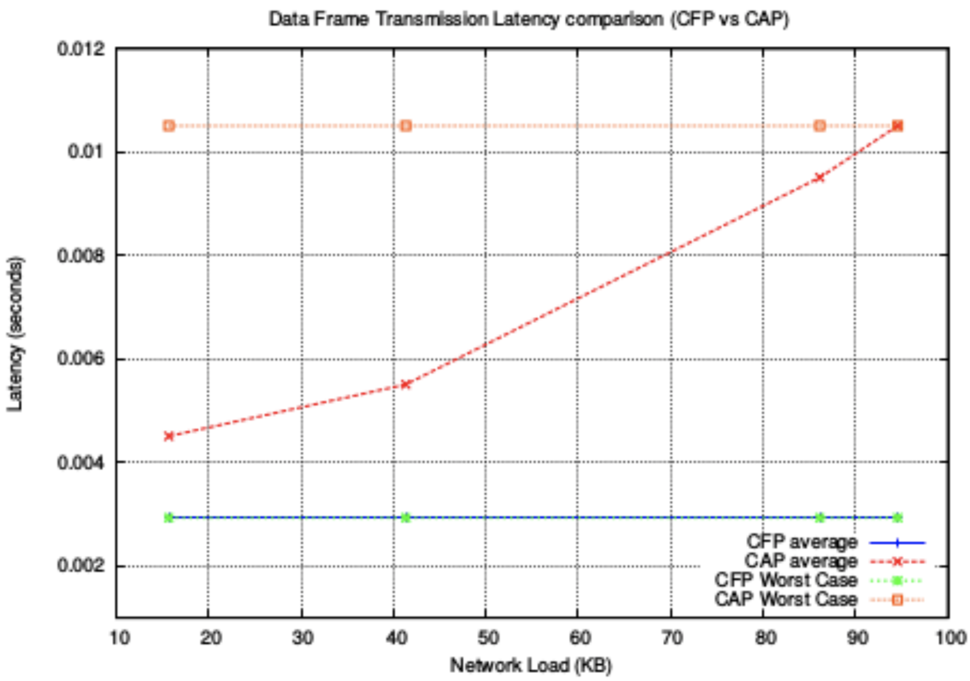

Figure 2.2: Data frame transmission latency comparision

Chapter 3

Documentation

This document will further explain the implementation techniques and results for two different type of simulators, namely **Matlab** and **NS3**.

3.1 MAC Superframe Structure

A coordinator on a VPAN can optionally bound its channel time using a superframe structure. A superframe is bounded by the transmission of a beacon frame. The structure of this superframe is described by the values of macBeaconOrder and macSuperframeOrder.
Beacon Order(BO) - It describes the interval at which the coordinator shall transmit its beacon frames.
0 <= BO <= 14
Superframe Order(SO) - It describes the length of the active portion of the superframe, which includes the beacon frame.
0<=SO<= BO<=14
Note:-
VPANs that do not wish to use the superframe structure (referred to as a non beacon-enabled VPAN) shall set both macBeaconOrder and macSuperframeOrder to 15.
MAC Superframe has been divided into active and inactive periods. The active period is further divided into CAP and CFP.

1. CAP - The CAP shall start immediately following the beacon and complete before the beginning of the CFP on a superframe slot boundary. If the CFP is zero length, the CAP shall complete at the end of the active portion of the superframe. A device transmitting within the CAP shall ensure that its transaction is complete (i.e. including the reception of any acknowledgment) one interframe space (IFS) period before the end of the CAP.

2. CFP - The CFP shall start on a slot boundary immediately following the CAP and it shall complete before the end of the active portion of the superframe. If any GTSs have been allocated by the coordinator, they shall be located within the CFP and occupy contiguous slots.

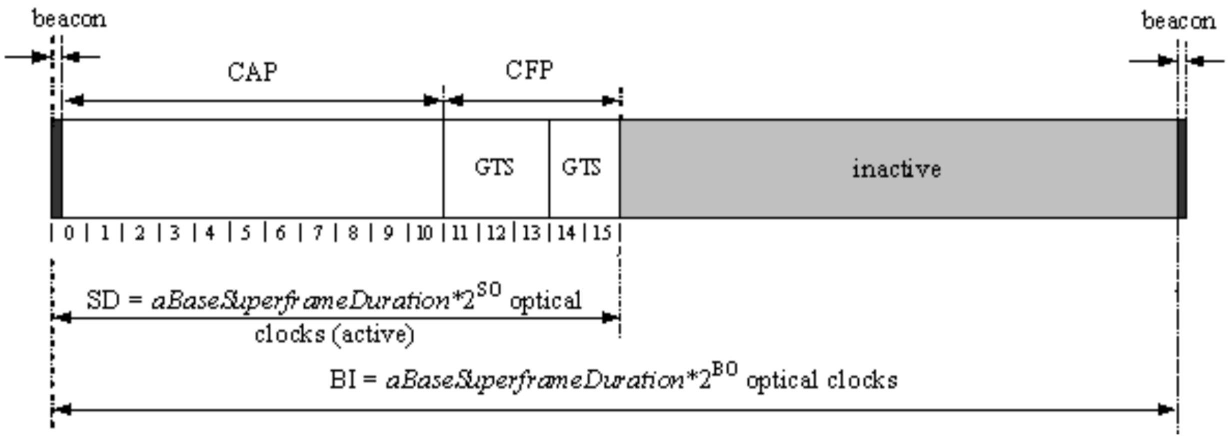

Figure 3.1: Superframe Structure

3.2 CONCEPT OF PRIMITIVES

The services of a layer are the capabilities it offers to the next higher layer or sublayer by building its functions on the services of the next lower layer. The services are specified by describing the information flow between the user and the layer. Each event consists of passing a service primitive from one layer to the other through a layer service access point (SAP) associated with an user. Service primitives convey the required information by providing a particular service.

A primitive can be one of the following four generic types:

• **Request:** The request primitive is used to request a service to be initiated.
• **Confirm:** The confirm primitive is used to convey the results of one or more associated previous service requests.
• **Indication:** The indication primitive is used to indicate the next higher layer of an internal event.
• **Response:** The response primitive is used to complete a procedure previously invoked by an indication primitive.
The following diagram shows the basic structure of primitives between service user and service provider layers:

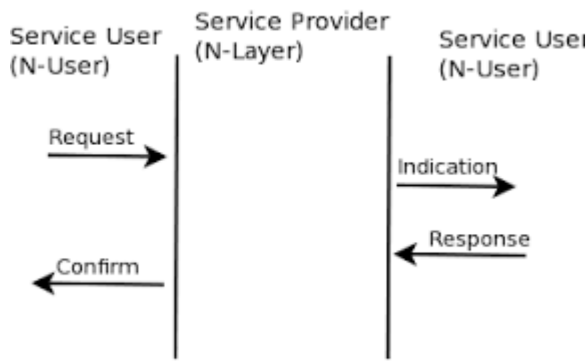

Figure 3.2: Types of primitives

The MAC sublayer provides the following two services, accessed through two SAPs: • **The MAC data service**, accessed through the MAC common part sublayer (MCPS) data SAP (MCPS-SAP), and •**The MAC management service**, accessed through the MLME-SAP.

3.3 CSMA/CA(Carrier Sense Medium Access/Contention Avoidance)

CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) is a protocol for carrier transmission in 802.11(https://ieeexplore.ieee.org/document/620533/) networks. Unlike CSMA/CD (Carrier Sense Multiple Access/Collision Detect) which deals with transmissions after a collision has occurred, CSMA/CA acts to prevent collisions before they happen. In CSMA/CA, as soon as a node receives a packet that is to be sent, it checks to be sure the channel is clear (no other node is transmitting at the time). If the channel is clear, then the packet is sent. If the channel is not clear, the node waits for a randomly chosen period of time, and then checks again to see if the channel is clear. This period of time is called the backoff factor, and is counted down by a backoff counter. If the channel is clear when the backoff counter reaches zero, the node transmits the packet. If the channel is not clear when the backoff counter reaches zero, the backoff factor is set again, and the process is repeated.
NOTE:-
CSMA/CA protocol is used in wireless networks because they cannot detect the collision so the only solution is collision avoidance.
CSMA/CA avoids the collisions using three basic techniques.

1. **Interframe spacing -** The MAC sublayer needs a finite amount of time to pro-

cess data received by the PHY. To allow for this, two successive frames transmitted from a device shall be separated by at least an IFS period; if the first transmission requires an acknowledgment, the separation between the acknowledgment frame and the second transmission shall be at least an IFS period. The length of the IFS period is dependent on the size of the frame that has just been transmitted.

2. **Contention Window -** Contention window is an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time. The number of slots in the window changes according to the binary exponential back-off strategy. It means that it is set of one slot the first time and then doubles each time the station cannot detect an idle channel after the IFS time.

3. **Acknowledgment -** The positive acknowledgment and the time-out timer can help guarantee that receiver has received the frame.

The IEEE802.15.7 specifies two-basic MAC modes:
(i) non-beacon-enabled, and (ii) beacon-enabled MAC.
The non-beacon enabled MAC is an asynchronous CSMA (Carrier-sense Multiple Access) MAC, which is very similar to the IEEE 802.11 MAC. The beacon-enabled MAC allows two different MAC periods:
(i) a synchronized-CSMA MAC period, and (ii) a time-slotted, contention-free MAC period.

3.4 SIMULATION

3.4.1 Matlab

In this section the implementation techniques applied for generating the Visible Light Communication (VLC) module in Matlab would be discussed. The Matlab model is created for two different topologies :

1. Peer to peer topology

2. Star topology

The models for both of these topologies would be individually discussed further in the document. Meanwhile the basics of these two topologies are explained below,

• Peer to Peer

In reference to the Visible Light Communication technology, peer to peer basically adheres to the communication networks where two nodes are taken into consideration. As the name suggests, the first peer would be the first node (either a device or a coordinator) and similarly the second peer would be the second node.

Note : *A peer-to-peer (P2P) network is created when two or more nodes are connected and share resources without going through a separate server. Here we are considering only two nodes.*

In Visible Light Communication (VLC), peer to peer is the most basic type of topology that could be implemented. In this, any one of the two nodes becomes the coordinator and other due to obvious reasons becomes the device. As there is only one device in picture, no case of collision exits.
The communication is two way, i.e., the coordinator as well as the device can transmit and receive information. An Additive White Gaussian Noise (AWGN) channel exists between the coordinator and the device.

• Star

The star topology is in a way an advancement of the peer to peer topology, where a single coordinator is connected with multiple devices. The communication established here is again a two way communication where the coordinator as well as all the devices can transmit as well as receive information.
An important thing to note here is that the devices are connected with only one coordinator and not with each other. So they cannot exchange information among themselves directly.
The channel would obviously be same as that in peer to peer topology.

3.4.1.1 PEER TO PEER

3.4.1.1.1 Architecture

The architecture of the model for peer to peer topology is based on the utilization of **comma-separated values** files as the communication channel between two Matlab windows. The Matlab windows act as the nodes for this communication model and are designated as the coordinator or the device depending on use.
To realize this communication model in a simulator such as Matlab, the execution

process needed to be divided into two major parts. Namely,
•Frame generation and decoding
•Transfer of primitives.
Note : *The concept of primitives has already been discussed earlier in this document.*
To generate and decode the frames, two separate functions have been created. The function *vlcMACFrameGenerator* is responsible for generating the frames based on the requirement of the model. The values of the parameters are depended on the *vlcConfig* file which allocates the default values for various frames whenever called.

For reference, the built-in zigbee library (IEEE 802.15.4 - MAC Frame - MATLAB) in Matlab 2017b was considered.

According to IEEE 802.15.7[6], a Visible Light Communiciation (VLC) frame is divided into three parts :
• **MHR (MAC Header)** - It comprises of frame control, sequence number, address information, and security-related information.

• **MSDU (MAC Service Data Unit)** - An MSDU is of variable length, which contains information specific to the frame type.

Note : *Acknowledgment frames do not contain a payload.*

• **MFR (MAC Footer)** - It contains FCS(Frame Check Sequence). The FCS field is 2 octets in length. The FCS is calculated over the MHR and MSDU parts of the frame. The FCS shall be only generated for payloads greater than zero bytes. The frames however could be of multiple types, each serving their own purpose. The following are the types as stated in the IEEE 802.15.7 standard :

1. Beacon frame
2. Data frame
3. Acknowledgment frame
4. Command frame
5. CVD frame

If a user wants to test how the frames are generated, the process is straightforward. All that needs to be done is that an object of the *vlcConfig* file needs to be passed into the *vlcMACFrameGenerator* function. The default type of frame is set to Beacon frame. So unless the user sets the frame type to a different value deliberately, a Beacon frame would be generated by default.

The following snippet shows an example to create a command frame with MAC command Beacon request,

```
commandConfig = vlcConfig;
commandConfig.FrameType = 'MAC command';
commandConfig.MACCommand = 'Beacon request';
Frame = vlcMACFrameGenerator(commandConfig);
```

The *vlcMACFrameDecoder* on the other hand does the exact inverse of *vlcMACFrameGenerator*. It takes a set of bits as input and decodes it into a type of frame with specific properties. The argument passed into the function must be a frame generated from the *vlcMACFrameGenerator* function, otherwise the output would be pointless.

The utilization of this function is same as that of the previous one, as it is required at the receiver end in the communication model. As the coordinator as well as the device both have the need for receiving frames, *vlcMACFrameDecoder* finds its application on both ends.

If a user wants to test the working of the decoder function, then the first step is to generate a frame as explained above (along with an example). The second step would be passing the generated frame into the *vlcMACFrameDecoder* function. The output would be a frame in an understandable format with its properties intact. Below is an example to test the working of the function,

```
[vlcFrame, dataPayload] = vlcMACFrameDecoder(Frame);
```

Note : *It is assumed here that the argument passed, i.e., 'Frame' is nothing but the output of the 'vlcMACFrameGenerator' function.*
Next is the transfer process of the primitives. This part is here the actual logic of the model comes into play. As discussed earlier, primitives are of four types, *request, confirm, indication and response*. The role of these primitives is to establish a common link between the MAC layer and the higher layer of either side (coordinator or

device). As the function of the higher layer depends on its MAC layer, it becomes necessary that the MAC communicates with it at every instance whenever a new process is initiated.

Taking in consideration the peer to peer topology here, the big picture that comes forward is that the process of transfer of primitives is a sequence of events. And this sequence depends on what frame the MAC layer receives or transmits from or to the other MAC layer. For the model that is implemented here, this sequencing of processes is achieved by initiating an infinite loop.

The working of the loop is in such a way that it generates the next primitive based on the previous primitive and the frame received. And the default values of the primitive parameters is set by passing an object of the class *vlcPrimitiveParameterConfig*. The functions *vlcMessageCoordSequencer* (for coordinator end) and *vlcMessageDeviceSequencer* (for device end) are called at every loop iteration. These are responsible for all the logical execution for the working of the model. The parameters passed into these functions are derived from the previous function call itself and the values generated after *vlcMACFrameDecoder* is executed.

As mentioned above, a **csv** file is used for establishing the link between the two nodes. In the case for peer to peer model, two files are used. One file *vlcProcess.csv* is used for writing the frames in form of bits by a node for the other node to read. And the other file *vlcHold.csv* is used for handing over the authority of execution to the device /coordinator based on the primitive logic. The *vlcHold.csv* file holds a single bit which could either be '1' or '0'. For '1', the authority is passed to the coordinator and for '0', the authority lies with the device.

Note : *It is mentioned here again that the bit value in 'vlcHold.csv' is entirely based on the logic of the sequencer. And as the processes are discrete, it becomes necessary that parallel execution doesn't takes place.*

3.4.1.1.2 Flowchart

The flowchart below explains the working of the peer to peer model.

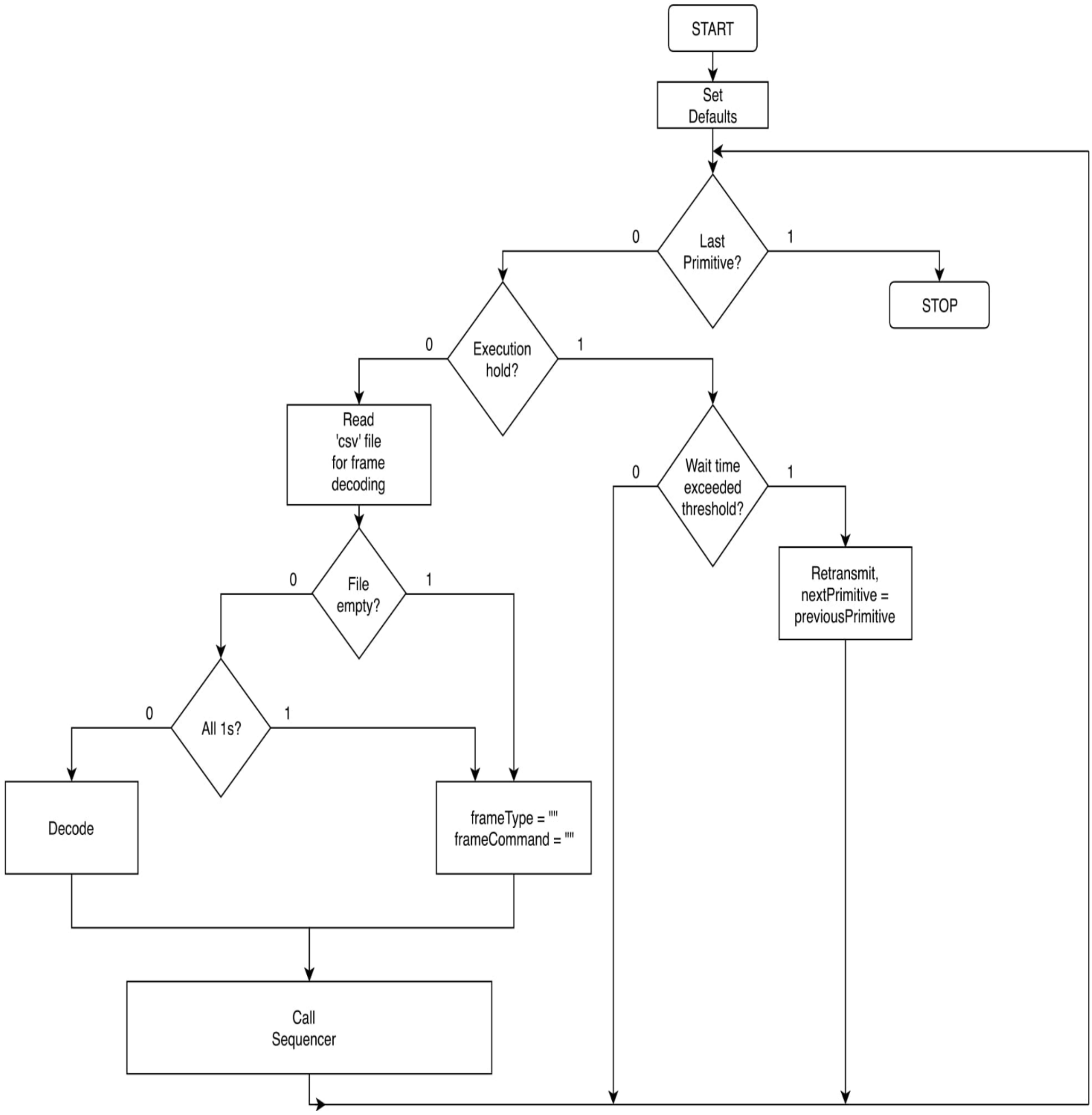

Figure 3.3: Flowchart : Peer to Peer

3.4.1.1.3 Assumptions

While this model is a substantial simulation model of the Visible Light Communication (VLC) Peer to peer topology, it does leave behind certain aspects of the communication model as defined in the IEEE 802.15.7 standard. Below is the list of assumptions made while generating this model in Matlab,

In this module, Contention Free Period (CFP) is ignored. No Guaranteed Time Slot

(GTS) are allocated.
• The VPAN is non-beacon enabled.
• Both Beacon Order (BO) and Superframe Order (SO) are set to 15, i.e. superframe structure is not used.
• CVD(Color Visibility Dimming) frame has been ignored.
• Physical layer is not involved. Only MAC to MAC communication is performed.
• Several MAC commands have been ignored viz. Information element, Color stabilization information, CVD disable and Color stabilization timer notification.
• A number of inessential primitives have been ignored viz. (naming the type of service here, considered all related primitives to be ignored) MLME-GET, MLME-GTS, MLME-RX-ENABLE, MLME-SET, MLME-SYNC and MLME-SYNC-LOSS.
Note : *'MLME-SET' is replaced by the method of simple object creation (of class vlcPrimtiveParameterConfig) by the end user for altering the values of the primitive parameters.*

3.4.1.2 STAR

3.4.1.2.1 Architecture

This section covers the working of the star model implemented in Matlab. The concept is quite the same as that of the peer to peer model. The medium used for establishing a communication between multiple Matlab links is the same i.e., **comma-separated values** files.

The algorithm is designed in such a way that the end user can set as many nodes (particularly devices) as he wants. The topology would support only one coordinator and multiple devices due to obvious reasons. Another important thing to mention here is that the process of generation and decoding of frames is also same as that in the peer to peer model.

Note : *To know more about the frame generation and decoding processes, refer section 3.4.1.1.1.*

There a number of *csv* files used in the process. Two among them serve the same purpose as discussed in the peer to peer model, namely, *vlcHold.csv* (gives the authority for execution to the device or the coordinator) and *vlcProcessi.csv* (used for writing frames).

Note : *'i' here denotes the device ID of any individual device, as every device has its own separate 'vlcProcess.csv' file.*

The other files in particular are *vlcAuthority.csv, vlcDissociation.csv and vlc-Terminate.csv*. The discussion about their purpose respectively is as follows, *vlcAuthority.csv* is used to assign the authority to the devices. The coordinator generated an authority ID and writes it in the file for the devices to read every time before execution. As each device has its own unique ID.So whenever the authority ID matches the device ID, that particular device is authorized for execution.

Now, about the *vlcDissociation.csv*, this file is required to maintain the count of the number of devices that are dissociated during the process (either abruptly or after successfully completing the whole execution). Another important utilization of this file occurs while deciding the authority ID for the devices, as it also holds the ID of the devices which are dissociated in the process. The random number generated is compared with the IDs present in this file and if any match is found then the generated number is immediately rejected. The execution is then looped back to the generation of the number again.

The final file remaining for discussion is *vlcTerminate.csv* and this file serves a very important purpose i.e., it sends an indication to the device for abrupt dissociation. Whenever BE exceeds the threshold value, a device is abruptly dissociated due to communication failure and this method is called.

3.4.1.2.2 Flowchart

The flowchart below explains the working of the device end of the star model.

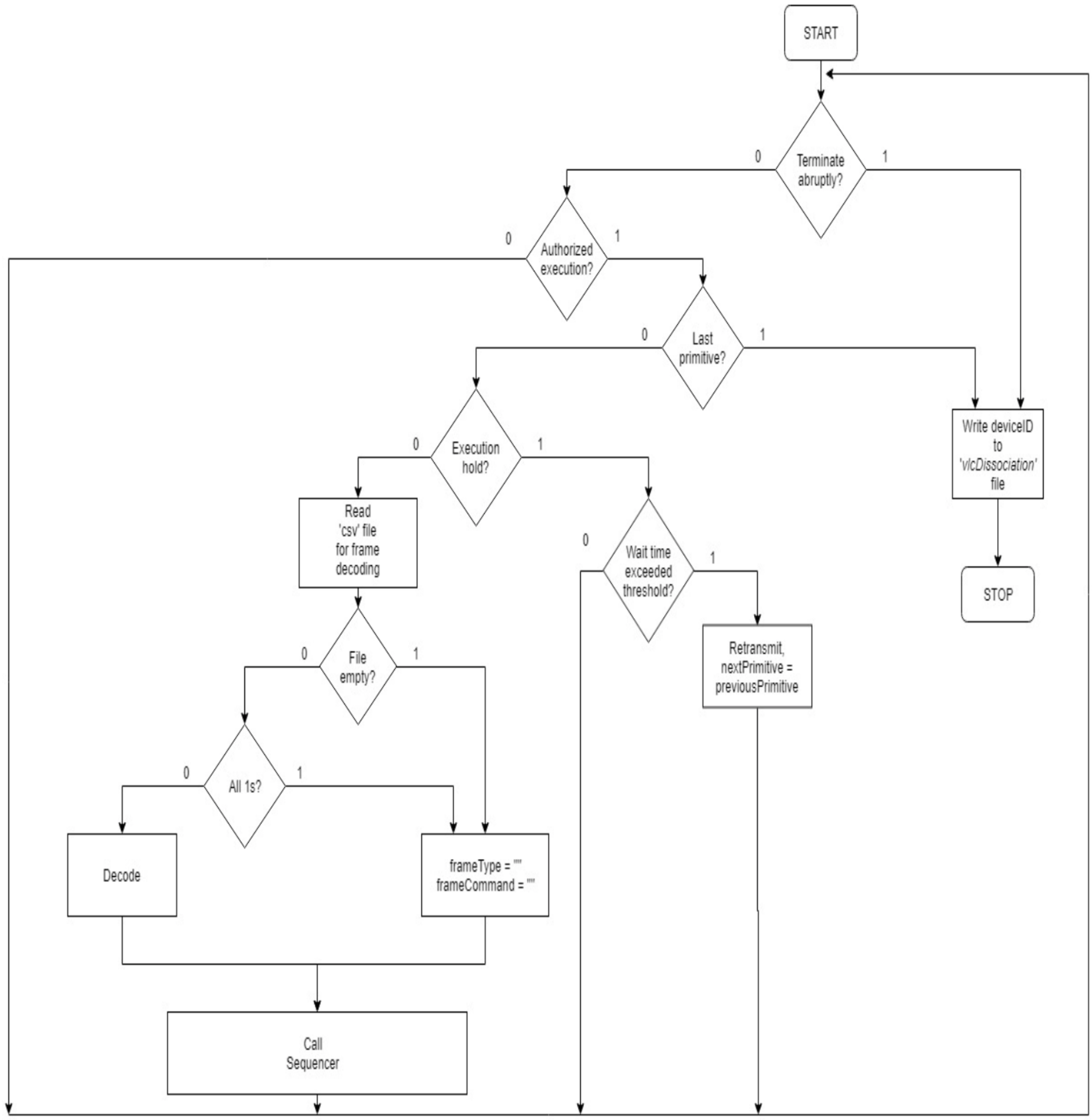

Figure 3.4: Flowchart : Device end for Star model

The following flowchart describes the working of the coordinator end of the star model.

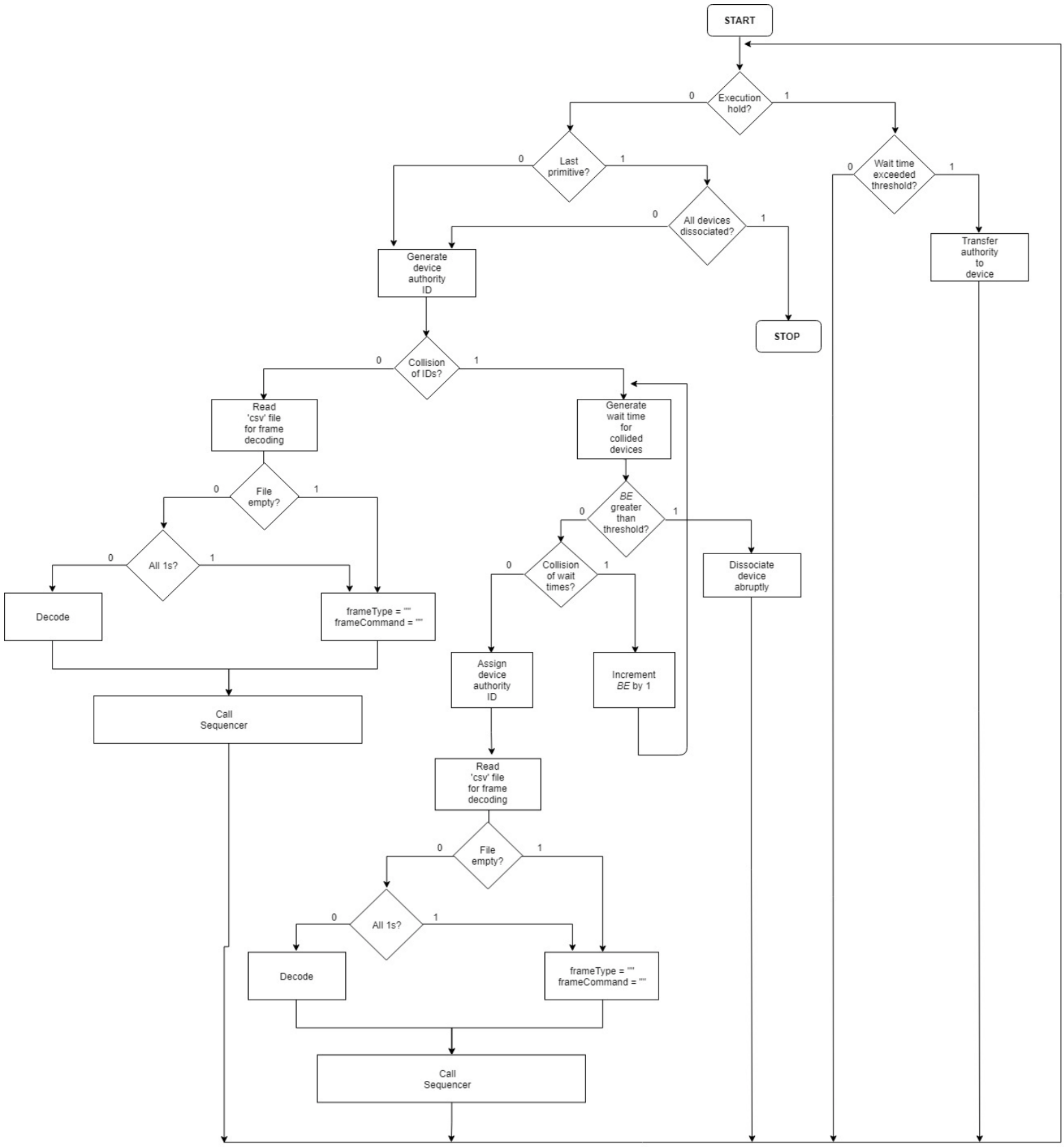

Figure 3.5: Flowchart : Coordinator for Star model

3.4.2 NS3

Ns-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns-3 is free software licensed under the GNU GPLv2 license, and is publicly available for research, development, and use.

The ns-3 project is committed to building a solid simulation core that is well documented, easy to use and debug, and that caters to the needs of the entire simulation workflow, from simulation configuration to trace collection and analysis. Furthermore, the ns-3 software infrastructure encourages the development of simulation models which are sufficiently realistic to allow ns-3 to be used as a realtime network emulator, interconnected with the real world and which allows many existing real-world protocol implementations to be reused within ns-3.

The ns-3 simulation core supports research on both IP and non-IP based networks. However, the large majority of its users focuses on wireless/IP simulations which involve models for Wi-Fi, WiMAX, or LTE for layers 1 and 2 and a variety of static or dynamic routing protocols such as OLSR and AODV for IP-based applications.
ns-3 also supports a real-time scheduler that facilitates a number of "simulation-in-the-loop" use cases for interacting with real systems. For instance, users can emit and receive ns-3-generated packets on real network devices, and ns-3 can serve as an interconnection framework to add link effects between virtual machines.

Another emphasis of the simulator is on the reuse of real application and kernel code. Frameworks for running unmodified applications or the entire Linux kernel networking stack within ns-3 are presently being tested and evaluated.

3.4.2.1 Introduction to NS3

ns-3 is a discrete-event network simulator in which the simulation core and models are implemented in C++. ns-3 is built as a library which may be statically or dynamically linked to a C++ main program that defines the simulation topology and starts the simulator. ns-3 also exports nearly all of its API to Python, allowing Python programs to import an âĂIJns3âĂİ module in much the same way as the ns-3 library is linked by executables in C++.

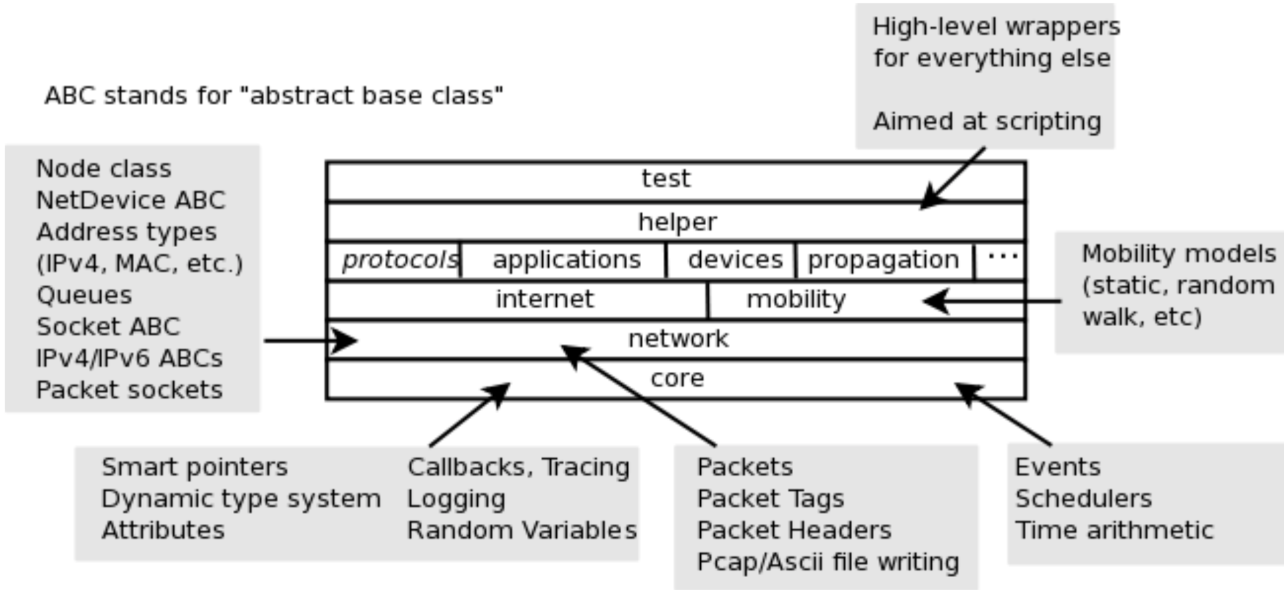

Figure 3.6: NS3 structure

The source code for ns-3 is mostly organized in the src directory and can be described by the diagram in Software organization of ns-3. We will work our way from the bottom up; in general, modules only have dependencies on modules beneath them in the figure.

We first describe the core of the simulator; those components that are common across all protocol, hardware, and environmental models. The simulation core is implemented in src/core. Packets are fundamental objects in a network simulator and are implemented in src/network. These two simulation modules by themselves are intended to comprise a generic simulation core that can be used by different kinds of networks, not just Internet-based networks. The above modules of ns-3 are independent of specific network an d device models, which are covered in subsequent parts of this manual.

In addition to the above ns-3 core, we introduce, also in the initial portion of the manual, two other modules that supplement the core C++-based API. ns-3 programs may access all of the API directly or may make use of a so-called helper API that provides convenient wrappers or encapsulation of low-level API calls. The fact that ns-3 programs can be written to two APIs (or a combination thereof) is a fundamental aspect of the simulator. We also describe how Python is supported in ns-3 before moving onto specific models of relevance to network simulation.

The remainder of the manual is focused on documenting the models and supporting capabilities. The next part focuses on two fundamental objects in ns-3: the Node and NetDevice. Two special NetDevice types are designed to support network emulation use cases, and emulation is described next. The following chapter is devoted to

Internet-related models, including the sockets API used by Internet applications. The next chapter covers applications, and the following chapter describes additional support for simulation, such as animators and statistics.

3.4.2.2 Visible Light Communication(VLC) Model

This chapter describes the implementation of ns-3 models for the visible light communication (VLC) as specified by IEEE standard 802.15.7 (2011).

The source code for the lr-wpan module lives in the directory **src/lr-wpan**.

3.4.2.2.1 Design

The grey areas in the figure (adapted from Fig 3. of IEEE Std. 802.15.7-2011) show the scope of the model.

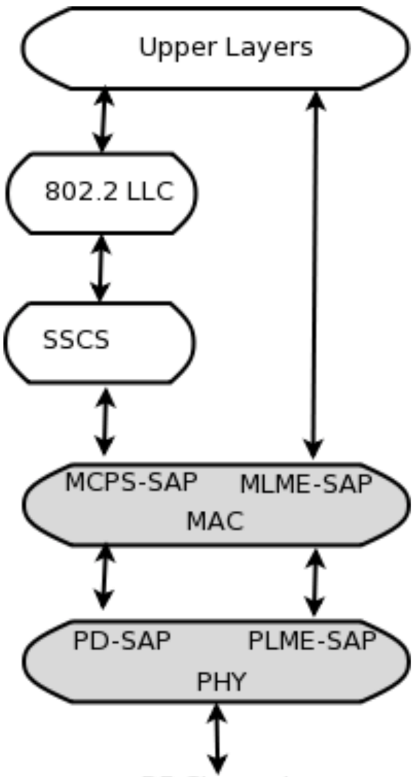

Figure 3.7: VLC module design

The Spectrum NetDevice from Nicola Baldo is the basis for the implementation.
The implementation also plans to borrow from the ns-2 models developed by Zheng and Lee in the future.

3.4.2.2.2 APIs

The APIs closely follow the standard, adapted for ns-3 naming conventions and idioms. The APIs are organized around the concept of service primitives as shown in the following figure adapted from IEEE Std. 802.15.7-2011.

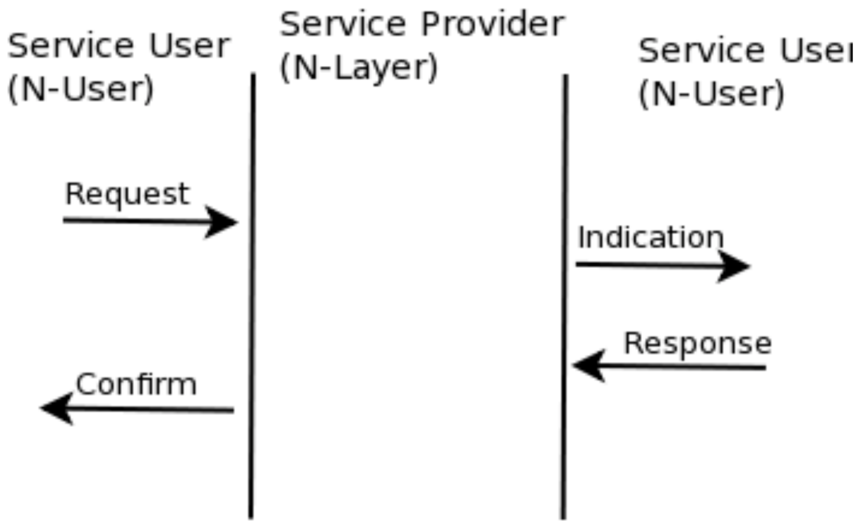

Figure 3.8: Primitives

The APIs are organized around four conceptual services and service access points (SAP):

- MAC data service (MCPS)
- MAC management service (MLME)
- PHY data service (PD)
- PHY management service (PLME)

In general, primitives are standardised as follows (e.g. Sec 6.2.1 of IEEE 802.15.7-2011)::

```
MCPS-DATA.request      (
                    SrcAddrMode,
                    DstAddrMode,
                    DstVPANId,
                    DstAddr,
                    MsduLength,
                    Msdu,
                    MsduHandle,
                    TxOptions,
                    SecurityLevel,
```

```
                        KeyIdMode,
                        KeySource,
                        KeyIndex,
                        DataRate,
                        BurstMode,
                        ColorReceived,
                        ColorNotreceived
                        )
```

This maps to ns-3 classes and methods such as::

```
struct McpsDataRequestParameters
{
  uint8_t m_srcAddrMode;
  uint8_t m_dstAddrMode;
  ...
};

void
VlcMac::McpsDataRequest (McpsDataRequestParameters params)
{
...
}
```

3.4.2.2.3 MAC

The MAC at present implements the unslotted CSMA/CA variant, without beaconing. Currently there is no support for coordinators and the relevant APIs.
The implemented MAC is similar to ContikiâĂŹs NullMAC, i.e., a MAC without sleep features. The radio is assumed to be always active (receiving or transmitting), of completely shut down. Frame reception is not disabled while performing the CCA.
The main API supported is the data transfer API (McpsDataRequest/Indication/-Confirm). CSMA/CA according to Stc 802.15.7-2011 is supported. Frame reception and rejection according to Std 802.15.7-2011 is supported, including acknowledgements. Only short addressing completely implemented. Various trace sources are supported, and trace sources can be hooked to sinks.

3.4.2.2.4 NetDevice

Although it is expected that other technology profiles (such as 6LoWPAN and ZigBee) will write their own NetDevice classes, a basic VlcNetDevice is provided, which encapsulates the common operations of creating a generic Vlc device and hooking things together.

3.4.2.2.5 Enabling Vlc

Add **vlc** to the list of modules built with ns-3.

3.4.2.2.6 Helper

The helper is patterned after other device helpers. In particular, tracing (ascii and pcap) is enabled similarly, and enabling of all vlc log components is performed similarly. Use of the helper is exemplified in **examples/vlc-data.cc**. For ascii tracing, the transmit and receive traces are hooked at the Mac layer.
The default propagation loss model added to the channel, when this helper is used, is the LogDistancePropagationLossModel with default parameters.

3.4.2.2.7 Examples

The following examples have been written, which can be found in **src/vlc/examples/**:

- **vlc-data.cc**: A simple example showing end-to-end data transfer.
- **vlc-error-distance-plot.cc**: An example to plot variations of the packet success ratio as a function of distance.
- **vlc-error-model-plot.cc**: An example to test the phy.
- **vlc-packet-print.cc**: An example to print out the MAC header fields.

In particular, the module enables a very simplified end-to-end data transfer scenario, implemented in vlc-data.cc. The figure shows a sequence of events that are triggered when the MAC receives a DataRequest from the higher layer. It invokes a Clear Channel Assessment (CCA) from the PHY, and if successful, sends the frame down to the PHY where it is transmitted over the channel and results in a

DataIndication on the peer node.

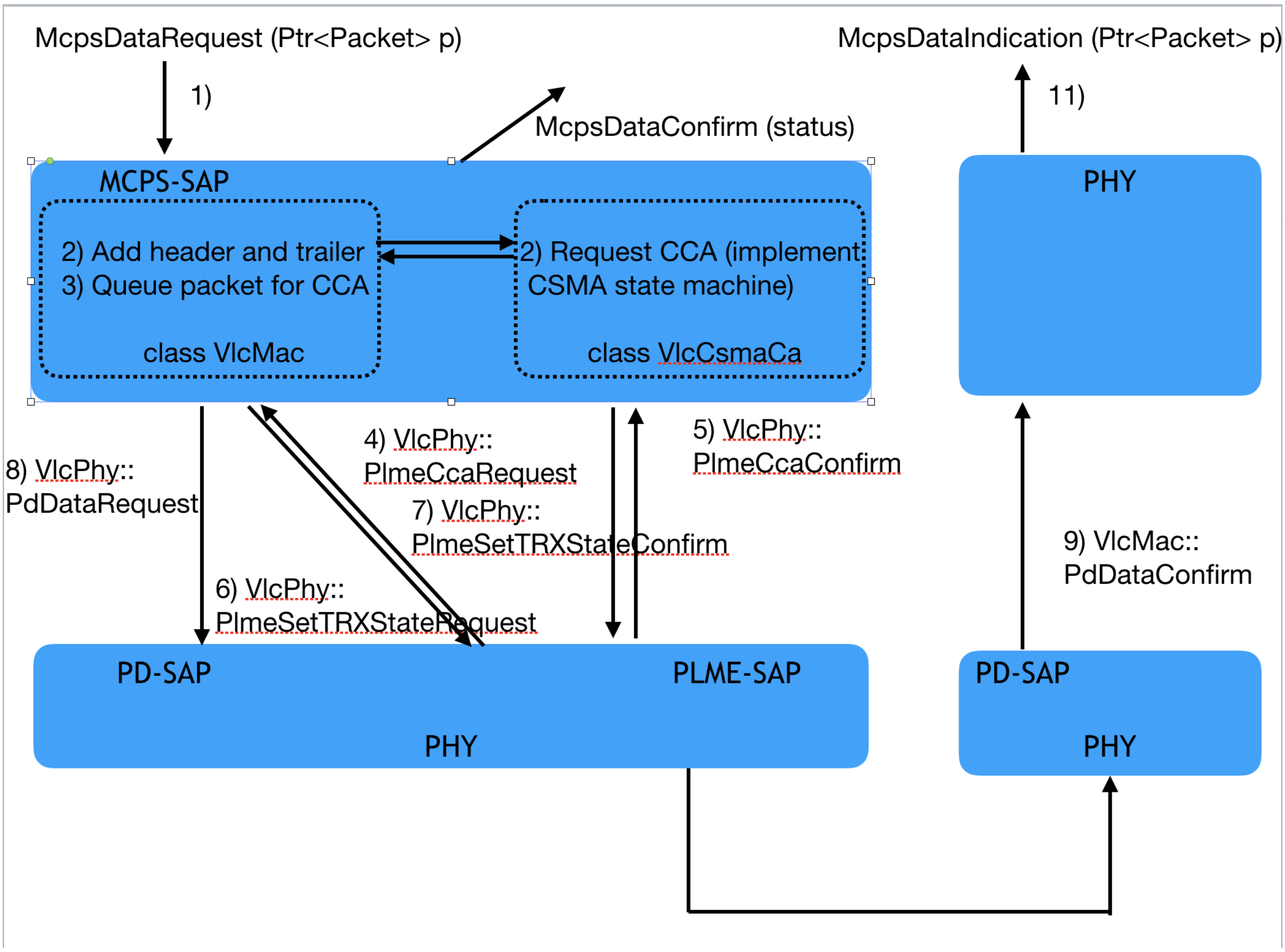

Figure 3.9: VLC Module Architecture

The example **vlc-error-distance-plot.cc** plots the packet success ratio (PSR) as a function of distance, using the default LogDistance propagation loss model and the 802.15.7 error model. The channel (default 11), packet size (default 20 bytes) and transmit power (default 0 dBm) can be varied by command line arguments. The program outputs a file named **802.15.7-psr-distance.plt**. Loading this file into gnuplot yields a file **802.15.7-psr-distance.eps**, which can be converted to pdf or other formats.

Chapter 4

Results

4.1 MATLAB

1. The models to demonstrate the working of Visible Light Communication for **peer to peer** topology and **star** topology has been successfully built in Matlab 2017b.

   The following functions have been successfully built and tested:

   • *vlcMACFrameGenerator* - To generate MAC frames according to 802.15.7 standard.
   • *vlcMACFrameDecoder* - To decode MAC frames.
   • *vlcCoordinatorstart* - To execute coordinator end logic.
   • *vlcDevicestart* - To execute device end logic.

   The following classes have been successfully built and tested:

   • *vlcConfig* - For defining default frame properties.
   • *vlcPrimitiveParamterConfig*- For defining default primitive properties.
   • *vlcMACPIBattributes* - For defining default PIB attributes properties.
   • *vlcMACsublayerconstants* - For defining default sublayer constants properties.

2. The **message sequence charts** for MAC to MAC communication have been successfully implemented.

Note : *Both the models ('peer to peer and 'star') have a manual (Read Me file) attached with their core directory for the end user. The files 'Test_CoordinatorStart' and 'Test_DeviceStart' are generated to test the working of the given models.*

4.2 NS3

The Visible Light Communication(Vlc) model has been build and tested succesfully on NS3. The following tests have been written, which can be found in **src/vlc/tests/**:

- **vlc-ack-test.cc**: Check that acknowledgments are being used and issued in the correct order.
- **vlc-collision-test.cc**: Test correct reception of packets with interference and collisions.
- **vlc-error-model-test.cc**: Check that the error model gives predictable values.
- **vlc-packet-test.cc**: Test the 802.15.4 MAC header/trailer classes
- **vlc-pd-plme-sap-test.cc**: Test the PLME and PD SAP per IEEE 802.15.7
- **vlc-spectrum-value-helper-test.cc**: Test that the conversion between power (expressed as a scalar quantity) and spectral power, and back again.

Validation of the result:

The model has not been validated against real hardware. The error model has been validated against the data in IEEE Std 802.15.7-2011. The MAC behavior (CSMA backoff) has been validated by hand against expected behavior. The plot is an example of the error model validation and can be reproduced by running **vlc-error-model-plot.cc**

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

In the models designed for **Matlab**, the designed MAC layer for network-enabled VLC has been successfully tested and met the functionality specifications referring the IEEE 802.15.7 standard such as support the network between the transmitter(Coordinator) to the receiver (Device) and error detection (we used the CRC-16). This greatly assists the IEEE 802.15.7 standard related research. The modules can represent the message sequence charts given in the standard. Using these modules, various properties such as packet *discard probability, packet delay, throughput, collision probability* and *bit-error rate* have been studied. In the module, inactive period and CFP(Contention Free Period) is ignored i.e. the concept of GTS(Guaranteed time slots) is not included.

The packet discard probability has been calculated by the CRC(Cyclic Redundant Check).

Note : *Matlab has built in function for CRC i.e. comm.CRCGenerator. It generates CRC code bits and append it to input data.*

In this module, CRC-16 has been used as specified by the standard. Other properties are calculated by modelling an AWGN(Additive White Gaussian Noise) channel with suitable SNR(Signal-to-Noise Ratio). The various BER and throughput values have been calculated at various SNR.

Talking about the model in **NS3**, Any type of network simulation regarding visible light communication could be carried out by the **vlc** model to obtain significant results.

In addition, Relevant mathematical equations or formulas could be embedded in the model for the favourable results.

5.2 Future Scope

We look forward to implement

- CSMA/CA(Carrier Sense Multiple Access / Contention Access) (**on Matlab*)
- Broadcast topology
- Physical layer (**on Matlab*)

for Visible Light Communication on Matlab.

Chapter 6

References

[1] MAC Layer Design for Network-Enabled Visible Light Communication Systems Compliant with IEEE 802.15.7

[2] Performance Analysis and Optimization of MAC Frame Size in IEEE 802.15.7

[3] IEEE 802.15.4 Std: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs) IEEE Computer Society Press, pp. 1–679,October 2003.

[4] https://in.mathworks.com/help/comm/examples/ieee-802-15-4-mac-frame-generation-and-decoding.html

[5] Improving NS-2 Network Simulator for IEEE 802.15.4 standard operation

[6] IEEE 802.15.7 Standard for Local and Metropolitan area networks.(2011). Part 15.7: Short-range Wireless Optical Communication Using Visible Light.

[7] Hass, H., Wireless data from every light bulb (2011). In TEDx forum. The video is available at: http://www.ted.com/talks/harald_haas_wireless_data_from_every_light_bulb.html.

[8] Jovicic, A., J. Li, and T. Richardson (2013) Visible light communication: opportunities, challenges and the path to market," IEEE Commun. Mag., 51(12): 26âĂŞ32.

[9] APPLICATIONS, LIMITATIONS, AND IMPROVEMENTS IN VISIBLE LIGHT COMMUNICATION SYSTEMS FAZLULLAH KHAN, FARMAN KHAN, QAMAR JABEEN, SYED ROOHULLAH JAN, SHAHZAD KHAN, Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan, Department of Computer Science, Bacha Khan University Charsadda, Pakistan

[10] VISIBLE LIGHT COMMUNICATION Shweta B. Suryawanshi, Prajakta Jadhav, Tanaka, Y., Komine, T., Haruyama, S., Nakagawa, M. (2003). Indoor visible light data transmission system utilizing white LED lights. IEICE Transactions on Communications, E86âĂŞ-B, 2440âĂŞ-2454.

[11] https://ieeexplore.ieee.org/document/6162558/

[12] https://ieeexplore.ieee.org/document/4699964/

[13] https://www.nsnam.org

[14] https://www.nsnam.org/docs/release/3.28/doxygen/index.html

[15] https://www.nsnam.org/docs/release/3.28/doxygen/group__lr-wpan.html

[16] https://ieeexplore.ieee.org/document/1700009/