IEEE Access
The journal for rapid open access publishing

IEEE Xplore Digital Library
http://ieeexplore.ieee.org/Xplore

# Visible Light Communication Module: An Open Source Extension to the ns3 Network Simulator with Real System Validation

Adel Aldalbahi, *Student Member, IEEE,* Michael Rahaim, *Member, IEEE,* Abdallah Khreishah, *Member, IEEE,* Moussa Ayyash, *Senior Member, IEEE,* and Thomas D.C. Little, *Senior Member, IEEE,*

***Abstract*—The emergence of new physical media such as optical wireless, and the ability to aggregate these new media with legacy networks motivate the study of heterogeneous network performance, especially with respect to the design of protocols to best exploit the characteristics of each medium.**

**We consider Visible Light Communications (VLC), which is expected to coexist with legacy and future radio frequency (RF) media. While most of the research on VLC has been done on optimizing the physical medium, research on higher network layers is only beginning to gain attention, requiring new analyses and tools for performance analysis. To meet this need, we have developed a new ns3-based VLC module that can be used to study VLC-RF heterogeneous networks via simulation.**

**The proposed ns3 module has been developed based on existing models for intensity modulated LED signals operating as lighting units transmitting to optical receivers at indoor scales (meters). These models and the corresponding simulation model are validated using a testbed implemented with a software-defined radio (SDR) system, photodetector, phosphor-converted "white" LEDs, and under PSK and QAM modulation.**

**Two scenarios are used in the validation of the VLC module: (i) using a receiver placed normal to the transmitter with varying range, and (ii) using a receiver with a fixed range with varying angle of acceptance. Results indicate good correspondence between the simulated and actual testbed performance. Subsequently, we demonstrate how the VLC module can be used to predict the performance of a hybrid WiFi/VLC network simulated using the ns3 environment with UDP, TCP, and combined network traffic.**

***Index Terms*—visible light communications (VLC), Network layer, ns3, network simulation.**

## I. Introduction

Optical wireless communication (OWC) has gained a great deal of recent attention in the research literature especially as it relates to Visible Light Communications (VLC) [1], [2]. OWC refers to different types of optical communications including VLC, Light Fidelity (LiFi), infrared (IR), ultraviolet (UV), free space optics (FSO) and can be used for short-range and long-range communication. Moreover, OWC can be used to supplement other communication technologies like WiFi, cellular and bluetooth to provide higher data rates at crowded areas or in vehicular communication networks to solve road congestion and reduce accidents. Other applications of OWC include underwater communications and server connectivity within data centers [3]. This recent attention has also elevated interest in standardization for different use cases spanning camera-based communication, WPANs, and LANS.

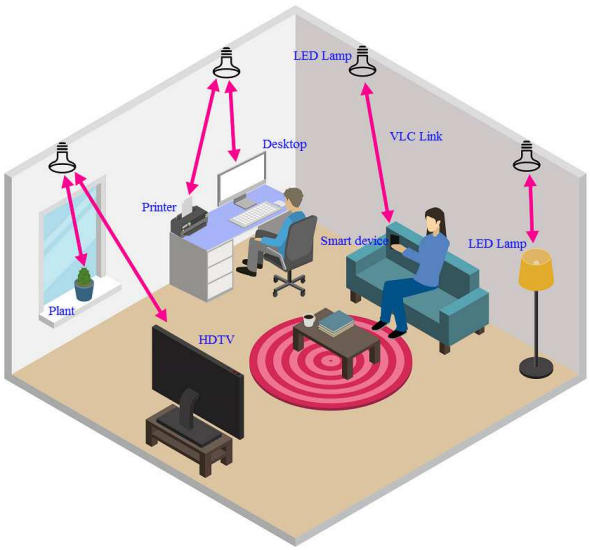


Fig. 1. Visible light as a communication medium between different wireless devices where the conventional lighting is expected to be replaced with LEDs.

The release of IEEE 802.15.7 standard for WPANs has led to increased interest in short range optical wireless communication using VLC. VLC uses visible light as a communication medium which is considered harmless to human eyes. It also has the potential for deployment of systems operating in a dual-use paradigm to provide both illumination and data communication. According to many researchers, VLC, due to its massive capacity, security, efficiency, and availability, is the ideal candidate to supplement existing wireless RF with more capacity. As a result, there is a growing interest in VLC at both the theory and applications levels. Fig. 1, depicts an environment where various wireless devices are connected using VLC.

When RF and OW media are combined to realize new capacity, it is essential to investigate the impacts of the physical media on the network performance [4]. Towards, this goal, we

Adel Aldalbahi and Abdallah Khreishah are with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, email: asa52@njit.edu, abdallah@njit.edu

Michael Rahaim is with the Department of Electrical and Computer Engineering, University of Massachusetts Boston, email:michael.rahaim@umb.edu

Thomas D.C. Little is with the Department of Electrical and Computer Engineering, Boston University, email: tdcl@bu.edu

Moussa Ayyash is with the Department of Information Studies, Chicago State University, email: mayyash@csu.edu

Digital Object Identifier: 10.1109/ACCESS.2017.2759779

discovered that there is a lack of robust tools to perform system analysis and evaluation of large hybrid networks involving VLC or VLC-RF components. With this gap in mind, and the potential to investigate large-scale hybrid systems, we embarked on an effort to implement an open source network simulator that is capable of performing simulation of VLC components and to facilitate network design and optimization.

A network simulator not only helps in meticulous and perspicuous understanding of the network model, but also avoids the heavy cost of implementing the physical system. Current network simulators, such as OPNET, ns2, ns3, OMNeT++ and NetSim have the structure to support evaluation of large scale networks, but they do not currently support VLC evaluation and environs. This has motivated our work to develop a VLC module within ns3 in order to offer an open source network-level VLC simulator. To the best of our knowledge, there are no freely available system modules included in popular network simulators (particularly ns3), and this is the first attempt to create an open source module for simulating communication systems through VLC channels.

ns3 is a platform for simulating Internet Protocols and real life network modeling. Being written in C++, it can be interfaced with external libraries and tools from the open source community. Moreover, Python codes can be written to interact with the ns3 simulator.

There are many open-source network simulators available for general purpose study of network models and protocols, but some are only available for commercial use [5]. Open source network simulators are more viable for the research community as the source code is openly provided and it is much easier to extend the network simulator to serve a particular requirement. Our criteria for selecting a simulation environment is based on the following attributes:

- Based on open source software and preferably multi-platform/platform independent
- Based on Object Oriented Programming (OOP) model
- Support for organizing system components in modular and hierarchical fashion
- Supports parallel execution environments
- Supports simulation of discrete event and dynamic communication system
- Supports a wide range of communications systems, modulation schemes, and protocols
- Supports large simulation scenario to make the simulation results more realistic and scalable
- Has robust debugging features
- Has an open interface for integrating external component libraries
- Has a tool for user interface (UI) and system visualization

The ns3 simulator meets these criteria, we decided to use ns3 as the platform supporting our proposed VLC module.

Our contribution in this paper is the use of existing VLC models in the ns3 module implementation and the subsequent validation of the module using a real VLC testbed. We also demonstrate how the ns3 module can be used to study hybrid WiFi/VLC systems.

The remainder of the paper is organized as follows: In section II, we review the related work on simulating VLC. In section III, we explain the theoretical model of VLC channel and the use of different performance metrics. VLC module as part of ns3 is explained in details in section IV. A testbed experiment using GNURadio and SDVLC to create unidirectional VLC communication link is presented in section V. Validation, results, and discussions are covered in section VI. We conclude the paper in section VII.

## II. Related Work

Simulation and analysis of VLC physical layer properties are covered in references [6]–[8]. The authors in reference [6] provide a simulation program based on Matlab and Simulink for indoor VLC. The named program can be used to calculate the illumination distribution, received signal waveform, and RMS delay spread under a single modulation scheme (i.e., NRZ-OOK). The authors in reference [7] present a channel model for VLC to obtain channel response under different indoor settings in which the simulation environment is created using Zemax. VLC physical layer (PHY) design is implemented for JIST, which is a high-performance discrete event simulator based on Java [8]. Reference [9] provides an implementation to simulate IEEE802.15.7 PHY based on OMNET++ simulation tool. The authors in reference [10] present comparative error rate performance evaluation of different IEEE 802.15.7 PHY types through Monte Carlo simulations. Considering references [6]–[10], we notice that none of the above simulators is based on open-source tools and thus, the need for common open source platform to study VLC as a network component has captured our attention.

VLC can be combined with other network technologies, such as WiFi to increase data rate and alleviate the load on RF spectrum. Initial work on the coexistence of hybrid system integrating VLC and RF based on simulation and analysis were introduced in references [11]–[14]. Our work described in reference [15] developed practical system implementations consisting of hybrid WiFi/VLC. In addition, we observed system throughput and user experience for web browsing. Our proposed module in this paper is integrated into the ns3 core library and can be used with the existing ns3 modules such as WiFi to study the performance of heterogeneous networks (HetNets).

There is other recent work on network performance using VLC. Reference [16] considers system throughput, latency, and other parameters in a system comprised of free space optical networking combined with VLC to provide multiple access. Reference [17]–[19] considers performance of vehicular networks that use VLC. Similarly, reference [20] deals with vehicular ad hoc networks (VANETs) based on simulation of urban mobility (SUMO) and ns3.

The above-mentioned related-works focus on simulation of VLC within a single layer and without practical validation in a real system. Our work focuses on integrating VLC physical models, spanning multiple layers, incorporating hybrid models, and a validation in a testbed.

## III. Theoretical Design of VLC Channel

The development of the VLC module requires sound foundational models of the physical system. In this section we

describe the physical layer that we adopt and implement in our VLC module.

VLC is realized with intensity modulation and direct detection (IM/DD) such that the signal is represented by variations in the instantaneous optical power and the received optical signal is directly converted to an electrical current. In OWC, the average optical power is constrained due to the lighting requirement of the dual-use VLC purpose or to the eye safety regulations for infrared (IR).

In our module, we assume that the source is an optical emitter that produces an instantaneous optical power, $X(t)$, in watts. The intensity modulation of VLC guarantees that the source has non-negative output and a maximum optical power constrained, $C$. The previous assumption restricts the instantaneous optical power to stay within this range: $0 \leq X(t) \leq C$. In order for the emitter to meet specific average optical power, it might incorporate a DC bias. We define $min(X(t))$ and $max(X(t))$ as the minimum and maximum optical signal, respectively. We also define the instantaneous optical signal power as $x(t) = X(t) - min(X(t))$. In addition, we define the average optical power and average optical signal power as $E[X(t)]$ and $E[x(t)]$, respectively; where $E[\cdot]$ is the expected value operation.

### A. Channel Model

In VLC, the channel can be modeled as a linear additive white Gaussian noise (AWGN) channel [21], [22]:

$$Y(t) = \gamma X(t) * h(t) + N(t). \tag{1}$$

where $\gamma$ is photodetector responsivity $(A/W)$, $*$ indicates the convolution operator, $h(t)$ is the optical channel impulse response and $N(t)$ represents the AWGN. The channel we adopt considers line of sight (LOS) links. The received optical signal power for a single source is given by:

$$P_r = h \cdot E[x(t)] \tag{2}$$

where, $h$ is the LOS link gain for a transmitting LED with Lambertian radiation pattern and is given by [21]:

$$h = \frac{(m+1)A}{2\pi d^2} \cos^m(\phi) T_s(\psi) g(\psi) \cos(\psi) \tag{3}$$

where the above parameters are defined as follows: $\phi$ and $\psi$ are the emission and acceptance angles, respectively. $m$ is the order of Lambertian emission, $A$ is the photodetector area, $T_s(\psi)$ and $g(\psi)$ are the gains of the optical filter and concentrator, respectively. The Lambertian order of the transmitter is given by $m = -\ln(2)/\ln\big(cos(\phi_{1/2})\big)$. Here, $\phi_{1/2}$ indicates the transmitters' semi-angle at half illuminance [22]. The gain of the optical concentrator can be determined by [21], [22]:

$$g(\psi) = \begin{cases} \frac{n^2}{\sin^2(\psi_c)} & 0 \leq \psi \leq \psi_c \\ 0 & \psi \geq \psi_c. \end{cases} \tag{4}$$

Here, the above model is for a compact parabolic concentrator (CPC) with an acceptance angle of $\psi$. $n$ represents the refractive index of the lens and $\psi_c$ is the receiver field-of-view (FOV).

### B. Performance Analysis of VLC

We evaluate the system performance using the following metrics: SNR, SER, PER and goodput under different values of the communication range $d$.

*1) SNR Modeling:* In OWC, SNR can be used to report the quality of the communication link. We consider SNR for measuring the capability of VLC link. The optical SNR can be defined as the ratio of the received average signal power to the ambient noise. In this paper, the optical SNR is defined as [22]:

$$SNR_o = \frac{P_r^2 \gamma^2}{\sigma_{total}^2} = \frac{\overline{I}_s^2}{\sigma_{total}^2} \tag{5}$$

where $P_r$ is the average received optical signal power, $\sigma_{total}^2$ is the total noise variance, and $\overline{I}_s^2$ is the squared average current. The noise can consist of different types of noise sources. For example, interference from fluorescent or incandescent light bulbs, photon-generated shot noise and thermal noise are all types of noise sources. In OWC, stemming noise from ambient light, is considered to be a major noise source. In our module, new noise models can be added in addition to the use of the existing ns3 noise models. For the default noise, we assumed the use of p-i-n/fieled-effect transistor (FET) transimpedance preamplifier in the receiver to evaluate total noise in a manner consistent with that of the testbed; hence, the total noise variance due to shot and thermal is given by [22]:

$$\sigma_{total}^2 = \sigma_{shot}^2 + \sigma_{thermal}^2 \tag{6}$$

where $\sigma_{shot}^2$ is the shot noise variance expressed by

$$\sigma_{shot}^2 = 2q\gamma P_r B + 2qI_{bg} I_2 B, \tag{7}$$

and thermal noise variance $\sigma_{thermal}^2$ is given by

$$\sigma_{thermal}^2 = \frac{8\pi k T_k}{G} \eta A I_2 B^2 + \frac{16\pi^2 k T_k \Gamma}{g_m} \eta^2 A^2 I_3 B^3, \tag{8}$$

Table I represents the parameters used in (7) and (8). In addition, the noise model in the simulator allows for simplified analysis if $\sigma_{total}^2$ is known or defined.

*2) Symbol Error Rate:* SER is a commonly used performance metric to measure the probability of error within a transmitted symbol.

In this paper, we utilize M-ary modulation (i.e., PSK and QAM) for the purpose of comparing the testbed results with that of the simulator. The testbed we utilized was made originally for RF and modified to support VLC [23].

- Phase Shift Keying (PSK) is a digital modulation scheme that encodes the symbols by varying the phase of a reference signal (i.e., the carrier wave) depending on the number of constellation points. We adopt PSK with 4 and 16 constellation points. The motivation for this decision is to increase the bandwidth efficiency of the PSK scheme. For M-PSK assuming high SNR, the SER can be approximated by [24]:

$$SER_{MPSK} \approx \frac{2}{\log_2 M} Q\Big(\sqrt{(2SNR_e}\sin\Big(\frac{\pi}{M}\Big)\Big) \tag{9}$$

  where $M$ is the modulation order, $Q(x)$ is the tail probability of the standard normal distribution and $SNR_e$ is the electrical SNR.

TABLE I
TOTAL NOISE PARAMETERS

| Parameter | Variable |
|---|---|
| Electronic charge | $q$ |
| Equivalent noise bandwidth | $B$ |
| Measured background current | $I_{bg}$ |
| Noise bandwidth factor | $I_2$ |
| Boltzmannbs constant | $k$ |
| Absolute temperature | $T_k$ |
| Fixed capacitance | $\eta$ |
| Open-loop voltage gain | $G$ |
| FET channel noise factor | $\Gamma$ |
| FET transconductance | $g_m$ |
| Noise bandwidth factor | $I_3$ |
| FET transconductance | $g_m$ |

- Quadrature amplitude modulation (QAM) is a non-constant envelope scheme that conveys data by varying the phase and the amplitude of the carrier wave. Using the same average signal power, QAM can achieve higher bandwidth efficiency than M-PSK. The SER for M-QAM is expressed as [24]:

$$SER_{MQAM} \approx \frac{\sqrt{M}-1}{\sqrt{M}} \frac{4}{log_2 M} Q(\sqrt{\frac{3}{M-1} SNR_e}). \quad (10)$$

In (9) and (10), the conversion from $SNR_e$ to $SNR_o$ is given by [25]:

$$SNR_e = \frac{\sigma_{I_s}^2}{\bar{I}_s^2} SNR_o. \quad (11)$$

Here, $\sigma_{I_s}^2$ is the signal variance. As a result, the SER calculations in (9) and (10) can be defined in terms of optical SNR in order to account for the average optical power constraints of the OWC signal.

*3) Packet Error Rate:* PER is another important metric we present in this paper to validate the receiver sensitivity. In ns3, it is convenient to group a number of $n$ bits to form a packet and depending on the error rate, the packet is discarded. The PER depends on the number of bits per symbol within a packet's payload. Assuming that the bit errors are independent of each other, the relationship between PER and SER is defined by

$$PER = 1 - (1 - SER_M)^{(p+h)}. \quad (12)$$

where $SER_M$ is handled by (9) and (10) for PSK and QAM, respectively. $p$ is the payload size in bits and $h$ is the number of bits in header. In the above definition, we dispense with error coding but this can be added to our simulator. Furthermore, the simulator provides an additional way to simulate BER/SER using the signal processing block given by Fig. 2 where for each given modulation schemes PSK or QAM, random bits can be generated, mapped to symbols then biased. A conversion factor is used to convert from electrical to optical SNR as in (11). At the receiving end, we introduce error to the signal and count the number of erroneous bits which are used to compute the PER.

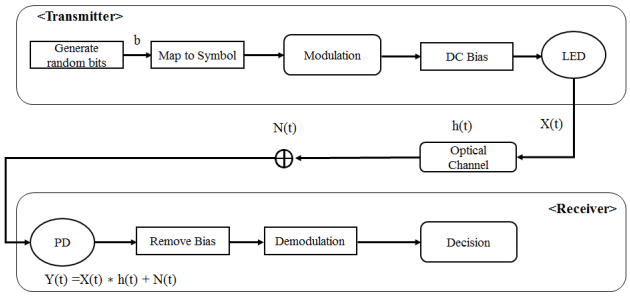


Fig. 2. signal processing block.

*4) Goodput:* The goodput is defined as the amount of payload bytes received by the receiver excluding any retransmitted bytes and protocol overhead per unit time. We define goodput as

$$goodput = \frac{8 \cdot p \cdot \zeta}{T}. \quad (13)$$

where $\zeta$ is the number of good packets received at the application layer and $T$ is the transmission period in seconds.

## IV. NS3-VLC MODULE

In this section, we provide in details the structure of our VLC simulator. The detailed information on how to use our model and to add it to the ns3 core library can be found in https://github.com/Aldalbahias. The ns3 simulator is a popular discrete-event network simulator targeted primarily for research and educational use. It is licensed under the GNU GPLv2 license and is available for research and development purposes. ns3 provides models of how packet data networks work and perform and provides a simulation engine for users to conduct simulation experiments. Our module is summarized in Fig. 3. Our implementation of the VLC simulator primarily consists of the following modules or classes:

**(1)** VLC net device, **(2)** VLC TX net device, **(3)** VLC RX net device, **(4)** VLC Mobility Model, **(5)** VLC Error Model, **(6)** VLC SNR, **(7)** VLC Channel Model, **(8)** VLC Propagation Loss Model, **(9)** VLC Modulation Scheme, **(10)** VLC Channel Helper, and **(11)** VLC Device Helper.

While modeling the classes, emphasis has been on minimal class creation and maximal use of the existing ns3 code base. Moreover, we have conformed to the ns3 naming conventions and style. In the following sections we describe how these classes are interconnected in more details.

### A. VLC Net Device

In ns3, a network device (net device) class captures the application programming interface (API) which the IP and the address resolution protocol (ARP) layers need to access to manage an instance of a network device layer. The network layer does not require any adjustment to handle new formats of address because the net device class captures the specific format of MAC addresses used by the device. However, the net device is an abstract class and is extended by many other

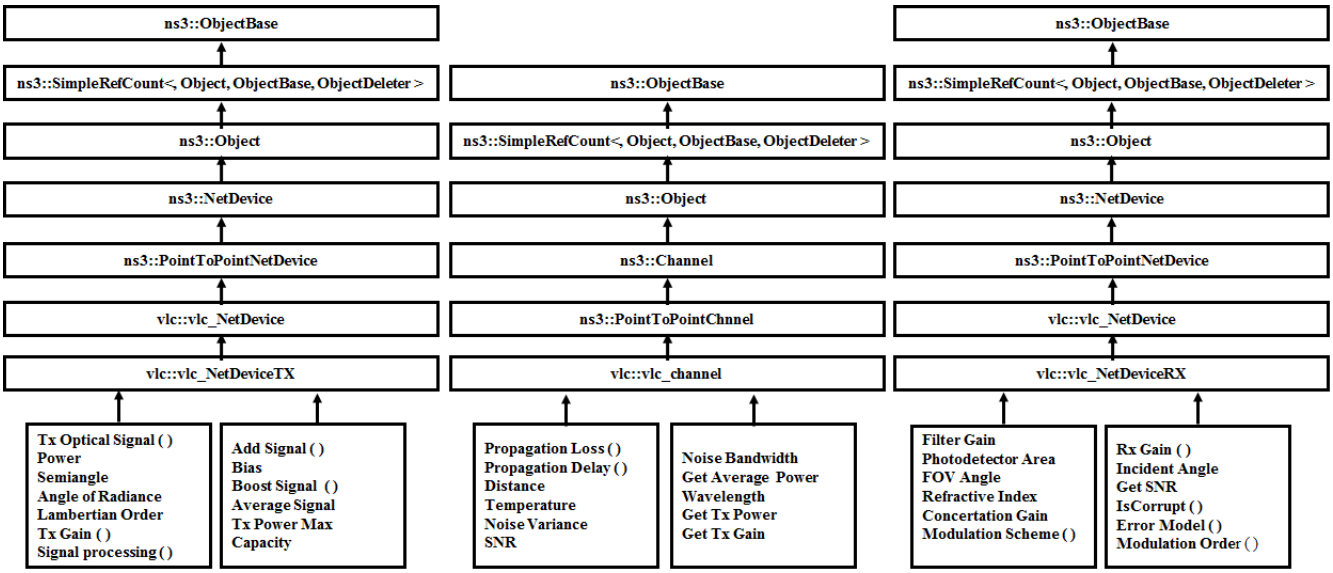


Fig. 3. ns3 Simulator-VLC Module.

classes in which the one that is needed for our purpose is p2p net device. In ns3, the p2p network modules implement a very simple p2p link connecting exactly two p2p net devices over p2p Channel [26]. The key point here is that the point-to-point protocol (PPP) link is assumed to be established and authenticated at all times.

In VLC, the main idea is to setup a communication network over two devices over a channel which follows certain communication protocol with the ability to be addressed in a network domain. Intuitively, it is made imperative to implement the VLC net device class from the ns3 p2p device class. In essence, the VLC net device class communicates over a p2p link and can be addressed over a network by network layer protocols. Additionally, the VLC net device is attributed with a mobility model which configures the device's physical state such as azimuth, elevation and coordinate vector. These parameters allow one to model the network behavior such as SNR, BER, PER, goodput, power loss, etc., in a dynamic scenario where the device configuration keeps on changing.

### *B. VLC TX Net Device*

The VLC TX net device class is created to capture the behavior of a device transmitter in a VLC network. It is derived from ns3 net device class and hence inherits all the attributes of a typical net device as well as the mobility configuration needed in a VLC ecosystem. Additionally, it has attributes to store transmitter optical signal power, maximum transmitted power, angle of radiance, semi-angle of the transmitter, Lambertian order, transmitter gain, and biasing power.

The VLC simulator is implemented with IM/DD such that the signal is represented by variations in the instantaneous optical power and the received optical signal is directly converted to an electrical current. We also use $P_{max}$ to represent the transmitter's maximum instantaneous optical. The source produces an instantaneous optical power $X(t)$, in watts, constrained by $0 \leq X(t) \leq P_{max}$. The transmitted optical signal power $x(t)$ is represented as a vector of instantaneous real values and associated with appropriate accessor methods. Since, optical intensity is non-negative, a DC bias is needed to boost the signal. Hence, a bias value is incorporated as a data member of this class. Initially, the signal can be populated with instantaneous signal values and after setting the bias the signal can be boosted in the simulator. In a dual-use lighting and communication VLC system, we have an illumination constraint that specifies average optical power. In order to achieve a specified average transmitted optical power, the signal may not necessarily utilize the entire range of source. In the simulator, optical power signal $x(t)$ can be recovered by subtracting the bias value from the biased signal.

### *C. VLC RX Net Device*

The VLC RX net device class is created to capture the behavior of a device receiver in a VLC network. Like VLC TX net device, it is also derived from ns3 net device and inherits all the attributes of a typical net device as well as the mobility configuration needed in a VLC ecosystem. Since the receiver class is designed by keeping in mind the components that would reside on an actual receiver, we have added filter gain, photodetector area, field-of-view angle, refractive index, angle of incidence, concentration gain, RX gain and receiver error model as private data members of the class. All these data members are associated with their respective setter and getter methods. Also, the receiver optical power signal and receiver optical intensity signals are added as vectors of values. These instantaneous values are calculated using the propagation loss model from the transmitter during signal transmission via the VLC channel. The received optical signal is used to determine the optical SNR where the BER can be set according to a specified modulation scheme. Before any processing of packets, the VLC error model can be used to determine the error rate of the packet, if any. In our work, we utilize a Boolean value to represent the model corruption status. A

bias value is subtracted from the received signal at the receiver since it does not add anything to the signal. Furthermore, these vectors can be used to determine the average optical signal power and intensity signals.

### D. VLC Mobility Model

The VLC mobility model is derived from the ns3 mobility model class and is used to capture the physical attributes of the device such as position, velocity, azimuth, and elevation. The ns3 mobility model base class is an abstract class defining the virtual functions for position, velocity, and other parameters. Also, it contains trace callback functions which can be used to capture the state values in case of change in object state.

In the VLC mobility model, azimuth represents the left and right rotation of the device. Elevation represents the up and down rotation of the device, and position represents the device location. These attributes are used within the transmitter, receiver, and channel model to determine the angle of incidence and angle of radiance.

### E. VLC Error Model

The VLC Error model class is derived from the ns3 Rate Error Model class where the packets are dropped according to the underlying distribution that depend on the BER value. The VLC error model follows a similar strategy by first calculating the error rate based upon a SNR value and then determining the appropriate BER/SER values (for example, by using (9) and (10) in case of PSK and QAM). The user can choose from a set of modulation schemes (i.e., OOK, MPAM, VPPM, MQAM, MPSK). Based on the BER/SER value, the Corrupt Packet function can be utilized to set the Boolean flag of packet corruption. Additionally, Q function has been implemented in the VLC error model which is needed for BER computation. Internally, it is implemented with *std library erfc* function.

### F. VLC SNR Model

The VLC SNR model class is an ns3 object that computes the SNR of the signal given by (5) and is dependent on optical signal power values, distance, responsivity, and other static constants. The physical variables such as power, responsivity, noise variance, temperature, and electric noise bandwidth have been made private data members with appropriate accessor methods. Other constants such as background current, noise bandwidth factor, open loop voltage gain, FET transconductance, FET channel noise factor, etc., are made static constants to conserve memory space. Noise variances, like shot variance and thermal variance values, are computed on the fly in CalculateNoiseVar function and set as a data member of the VLC SNR. Finally, the SNR value is computed in the CalculateSNR function using noise variance, optical signal power, and responsivity.

### G. VLC Channel Model

The VLC Channel given by (3) is derived from the ns3 p2p channel class. It inherits all the virtual functions to receive signals from the VLC TX net device and to transmit it to corresponding VLC RX net device. The main idea here is to provide a model that acts as a channel between two nodes while also keeping in mind the signal corrupting factors present in real world scenario. The signal corruption is modeled by VLC SNR which corrupts the transmitted packets according to the optical signal power and distance between the two devices. Afterwards, the corrupted signal is forwarded to the VLC RX net device. These computations are done each time a packet is passed across the channel; hence providing an on the fly packet corruption model. The private data members of the class are propagation loss, propagation delay, SNR, and average power. At runtime, the ns3 propagation loss model is instantiated with VLC propagation loss model to compute the optical signal power at the receiver end. The propagation delay model is used to estimate delays between the communicating nodes.

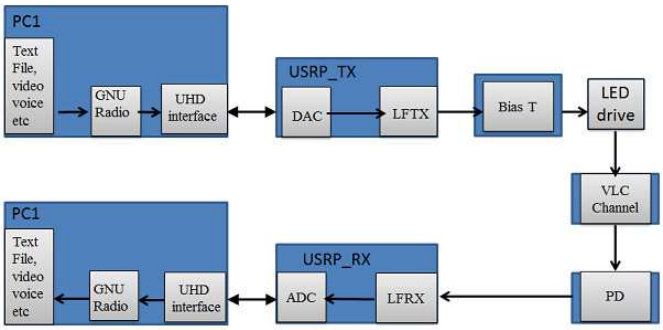


Fig. 4. VLC Signal Chain.

### H. VLC Helpers

The purpose of the helper API in ns3 is to make the code development and upgrade more easier. In our module, we created channel and NetDevice helpers to make the implementation of large-scale networks simpler. The VLC channel helper consists of methods and functions to link the VLC channel to two NetDevices where the VLC NetDevice helper is used to set different attributes on top of these connected devices. These helpers are not only limited to NetDevice and channel parameters but also can be used to assign different methods depending on the type of network used.

## V. Testbed Implementation

We utilize GNURadio which is an open source toolkit to provide signal processing and software defined radios. Along with GNURadio, we utilize universal software radio peripheral (USRP-N210) with low frequency transmitter and receiver (LFTX/LFRX) as an interface to the VLC front end where the conventional transmitter and receiver RF antennas are replaced by optical transmitter and photodetector, respectively. This software-defined visible light communication (SDVLC) [23], allows us to realize the VLC PHY layer and implement a single-way UDP link. The testbed signal chain is presented in Fig. 4.

A text file, video, voice, etc., can be transmitted and passed to GNURadio software that are used to handle both the signal

processing and MAC layer protocols. For example, the text data is packetized then digitally sampled using GNURadio signal processing blocks. Different parameters such as sampling rate, data rate, modulation scheme, etc., can be adjusted by modifying the signal processing block inside GNURadio. The digital samples are then forwarded to the USRP using USRP hardware driver (UHD) interface over 1G Ethernet connection. The signal is then passed to LFTX daughterboard for passband modulation. The driver signal output is transmitted over the optical channel. The transmitted light intensity is then detected at the receiver and converted into electrical current based on direct detection using a commercial photodetector. The LFRX daughterboard at the receiver is used to demodulate the received carrier signal while the receiver USRP performs the conversion from analog to digital. The digital sampling from the USRP is forwarded to the receiving PC over 1G Ethernet connection using (UHD) interface. At the receiving side, the GNURadio application is used to process the received signals through signal processing blocks. The output is then sent out to a text file, media player, etc., then to a preferred software such as Matlab and Simulink for more analysis.

The front-end of the testbed consists of bias-T and identical Osram semiconductor LEDs in series with MOSFET. The bias is required to shift the bipolar signal generated by the USRP such that the input to the LED driver is within the linear range of the conversion. At the receiver, we use a commercial transimpedance photodetector (Thorlabs-PDA36A) with an aspheric condensing lens (ThorlabsbALC2520-A) to enhance detection of the incoming light intensity. The detector utilizes a PIN silicon photodiode with varying responsivity depending on the detected visible wavelength. The detector is equipped with eight positions rotary switch to control the gain. The testbed parameters are given in Table II.

The testbed we implement consists of a single Dell PC running Linux (Ubuntu 14.04 LTS) operating system, single USRP-N210 equipped with two daughterboards (LFTX/LFRX), ethernet connection, an analog LED driver board, a DC bias-T, a triple output power supply (Hewlett Packard hp-6235A), a Si transimpedance photodetector (Thorlabs-PDA36A) with an optical lens, a measuring tape, and a mixed domain Oscilloscope (Tektronix-MDO4034-3) as shown in Fig. 5. The PC acts as both transmitter and receiver, equipped with Gigabit Ethernet controller (Intel corporation Ethernet connection I217-LM). The static routing network configuration for the testbed is given in Table III. From Fig. 5, the PC with a source IP address of 192.168.10.1 is connected to the USRP-N210 with an IP address of 192.168.10.2 through a Gigabit Ethernet cable. The single USRP is equipped with transmitting and receiving daughterboards. Therefore, the received data is processed through the same USRP then back to the same PC using the same Gigabit Ethernet cable. The bias-T and the input of the power supply are set to the values shown in Table II.

TABLE II
TESTBED PARAMETERS

| Parameter | Value |
|---|---|
| number of LEDs (LUW CN5M) | 16 |
| Area of photodetector | 13.0 $mm^2$ |
| Responsivity | $0.2 - 0.4$ $A/W$ |
| input of power supply | 6.0 V |
| Bias-T (mini-circuit, $0.1 - 4200$ $MHz$) | 2.0 V |

TABLE III
TESTBED ROUTING TABLE

| Source | Next hope | Destination | Mask | interface |
|---|---|---|---|---|
| 192.168.10.1 | 192.168.10.2 | 192.168.10.1 | 255.255.255.0 | eth0 |

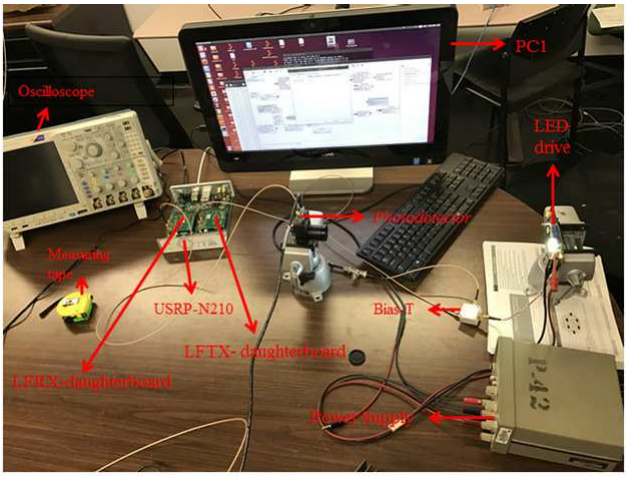


Fig. 5. System testbed.

## VI. VALIDATION, RESULTS, AND DISCUSSION

To express the testbed system using our simulator, we demonstrate a p2p VLC network. The configuration consists of two nodes as shown in Fig. 6, where node $A$ acts as an optical source and node $B$ as a photodetector. Node $A$ utilizes the transmitter's parameters provided by NetDeviceTx class as mentioned earlier where node $B$ utilizes the receiver's parameters set by NetDeviceRx. The VLC channel is used to make a connection between node $A$ and $B$. Also, ns3 static routing is used to force the data flow in one direction from $A$ to $B$ through UDP connection. In addition, we used Internet Protocol version 4 (IPV4) to assign IP addresses for both the transmitter and the receiver. The simulator parameters used in this experiment are provided by Tables IV and V where the collected data from the simulator was passed to Matlab and Simulink for processing.

For comparison, we show the performance of our simulator to that of the testbed in terms of SNR, SER, PER and goodput using two scenarios as shown in Fig. 7. In scenario (i) the photodiode's normal is parallel to the normal of the transmitter ($\psi = \phi$) with varying distance while in (ii) the angle of acceptance is rotated with fixed distance between transmitter and receiver. Because our simulation is based on ns3, it inherits all of the associated features of ns3, giving it excellent potential to explore a wide range of configurations involving VLC components. Next, we explain the testbed and

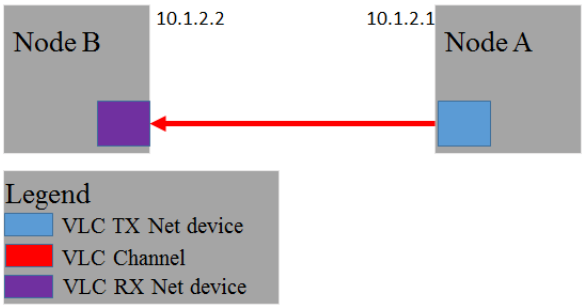


Fig. 6. Simulated Example.

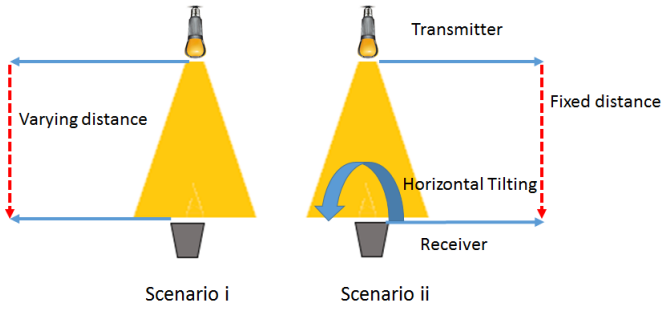


Fig. 7. Experimental Scenarios: *Scenario* i, receiver is located directly under transmitter where distance is varying. *Scenario* ii, receiver is tilted horizontally.

TABLE IV
SIMULATOR PARAMETERS FOR SCENARIO $i$ AND $ii$

| Parameter | Variable | scenario $i$ value | scenario $ii$ value |
|---|---|---|---|
| Angle of emission | $\phi$ | $0°$ | $0°$ |
| Angle of Acceptance | $\psi$ | 0 | $0° \leq \psi \leq 45°$ |
| Distance | d | $0.4m \leq D \leq 2m$ | $0.75cm$ |
| Total Transmitted bytes | $p_t$ | $1 \cdot 10^6$ $bytes$ | 16000 $bytes$ |
| Payload size in bytes | $p$ | 1472 | 1024 |
| data rate (4PSK, 4QAM) | $R_4$ | $400Kbps$ | $41Kbps$ |
| data rat (16PSK, 16QAM) | $R_6$ | $600Kbps$ | $90Kbps$ |

simulator results using the two scenarios mentioned above.

### A. Signal-to-Noise-Ratio (SNR) Analysis

This section compares the performance of the simulator to the implemented testbed in terms of SNR. The results we obtained from the testbed were measured in a dark room with lights-off to alleviate the effects from ambient light. The experiment was repeated multiple times to ensure better estimation of SNR. It should be noted that the LOS is considered in this experiment. The testbed SNR ($SNR_t$) is computed for the two scenarios in Fig.7 using the following

$$SNR_t = \frac{E^2[V_s]}{\sigma_n^2}. \quad (14)$$

where $E^2[V_s]$ is the square of the average voltage measured at every 10 cm and is proportional to the average current used in (5). Therefore, observing $\sigma_n^2$ in terms of voltage makes the SNR definition in (14) equivalent to the definition in (5). In addition, $\sigma_n^2$ was measured by placing the receiver in front of the transmitter without any transmission. To observe $10k$ samples of voltage every one second, we used an oscilloscope and connected it directly to the photodetector. The simulator SNR ($SNR_s$) was estimated using (5) and the parameters given by Tables V and IV. The testbed and simulator SNR results for scenario $i$ and $ii$ are shown in Fig. 8 and Fig. 9, respectively. The results show that in both cases the $SNR_s$ closely matches the $SNR_t$. We can also see that the rotation of the receiver has an impact on the received optical power. Therefore, the SNR effectively vanishes beyond $40°$. In our system, the simulator can mimic the performance of real systems if the total noise is accurately estimated. At some

TABLE V
COMMON SIMULATOR PARAMETERS FOR SCENARIOS $i$ AND $ii$

| Parameter | Variable | Value |
|---|---|---|
| Semi-angle at half illumination | $\Phi_{1/2}$ | $35°$ |
| Filter Gain | $T_s$ | 1 |
| Boltzmann's constant | k | $1.3806e^{-23}$ $J/K$ |
| Noise bandwidth factors | $I_2$,$I_3$ | 0.562, 0.0868 $respt$ |
| Background current | $I_B$ | $1.13^{-6}$ $A$ |
| Open-loop voltage gain | $G$ | 10 |
| fixed capacitance of photodetector | $\eta$ | $112pF/cm^2$ |
| Responsivity | $\gamma$ | 0.2 $(A/W)$ |
| FET | gm | 30 $mS$ |
| electronic charge | $q$ | $1.60217e^{-19}$ $C$ |
| Area of PD | $A$ | 0.13 $cm^2$ |
| Refractive Index | $n$ | 1.5 |
| FOV at receiver | $\psi_{con}$ | $28°$ |
| Noise Bandwidth | B | 300000 $b/s$ |
| Absolute temperature | $T_k$ | 298 |
| FET channel noise factor | $\Gamma$ | 1.5 |
| Modulation order, PSK16, PSK4 | $M_p$ | $4, 16$ |
| Modulation order, QAM4, QAM16 | $M_q$ | $4, 16$ |
| number of bits | $N$ | $2, 4(i, e4, 16resp)$ |
| Elevation | elevation | $180.0°$ |
| Azimuth | azimuth | $0°$ |
| port number | $P$ | 5000 |

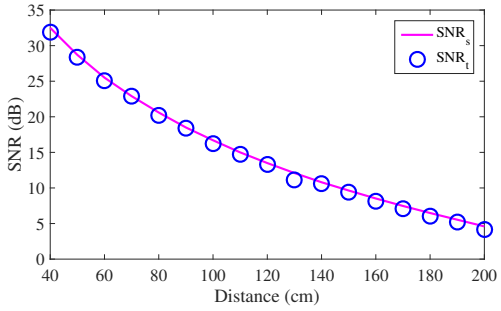


Fig. 8. Scenario $i$: Simulator and Testbed SNR with $\psi = \phi$ and varying distance between transmitter and receiver.

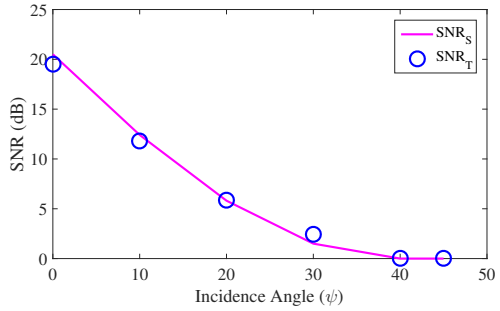


Fig. 9. Scenario $ii$: Simulator and Testbed SNR under different values of $\psi$.

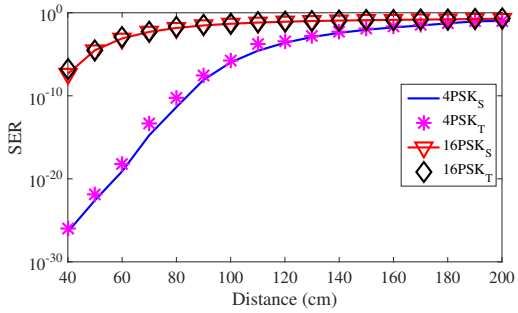


Fig. 10. 4 and 16 PSK SER results for simulator and testbed using Scenario $i$.

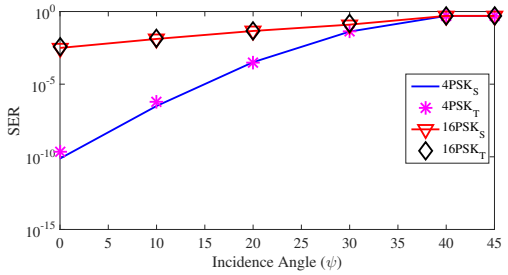


Fig. 11. 4 and 16 PSK SER results for simulator and testbed using Scenario $ii$.

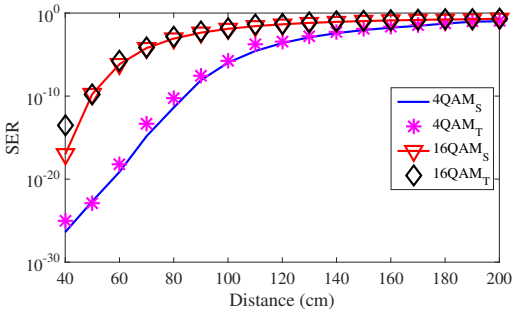


Fig. 12. 4 and 16 QAM SER results for simulator and testbed using Scenario $i$.

distances, in the same figures mentioned above there is a small variation in the result of the simulator to that of the testbed, for example at 130 cm or when $\psi = 30°$. We believe this is due to the used LED driver which consists of 16 LEDs. These LEDs can produce inter-symbol interference which affect the received signal strength.

### *B. Symbol Error Rate (SER) Analysis*

The SER equations given in (9) and (10) represent the simplest calculations of SER without taking into consideration other effects such as loss or signal enhancement (i.e., filtering). To compare the performance of the simulator to that of the testbed in terms of SER, we avoid the use of existing GNU-Radio modulation and demodulation blocks. These blocks contain a root raised cosine (RRC) filter used to enhance the shape of the transmitted signal and may cause inter-symbol interference (ISI), which is not considered in (9) and (10). To observe the SER of the testbed, we implement the flow graph given by Fig. 18 which represents the signal processing blocks. The SER was measured using built-in GNURadio Error Rate block. This block compares stream of reference data symbol/bits to the received input data and checks if they match. At this point, we highlight how the flow graph works. The flow graph consists of a simple transmitter and receiver to best represent the equations. We used a random source to generate bytes containing $four$ and 16 combinations of random data depending on the used modulation scheme ( i.e 4PSK-QAM or 16PSK-QAM). The generated bytes are mapped to symbols specified by the map block. The mapped symbols are differentially encoded using differential encoder block before they map to proper constellations points through chunks to symbols block. The transmitted signals are amplified using multiplier constant block then passed to the USRP transmitter. At the receiver side, we use the same USRP to receive the incoming signal. The received signal is then passed to an automatic gain controller (AGC), which is a closed loop-feedback to provide control over signal amplitude. After controlling the amplitude, the signal is passed to either a least mean square equalizer (LMS) or to a constant modulus algorithm (CMA) to lock on the signal phase, depending on the type of modulation scheme used. A Costas-loop [27] can be used here to correct for both phase and frequency offset. The constellation decoder block takes in the constellation object block as its parameter. This block has matching constellation points similar to the transmitter. The symbols are decoded using differential encoder block to get the original samples then passed to map block. The final symbols are passed to an unpack block to unpack the received bytes before passing them to the error rate block for comparison with the reference from the transmitter. The output from the flow graph is compared to (9) and (10) after SNR conversion.

The results of scenario $i$) and scenario $ii$) for simulator and testbed using PSK and QAM modulation schemes in terms of SER are shown in Figures 10-13. We utilize PSK and QAM schemes with 4 and 16 modulation order where $PSK_S$ and $PSK_T$ represent the simulator and testbed SER, respectively. The simulator produces results that closely match that of the testbed, but a noticeable mismatch can be seen in the previous figures. We conclude that this is due to an unpredictable delay caused by the GNURadio signal processing blocks. Also the delay in real systems cannot be deterministic and depends on

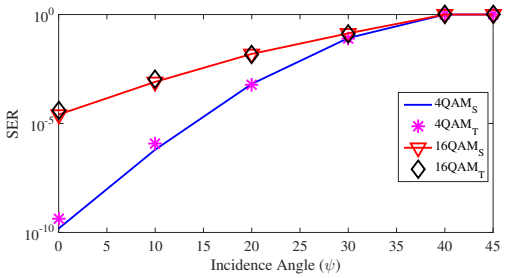


Fig. 13. 4 and 16 QAM SER results for simulator and testbed using Scenario $ii$.

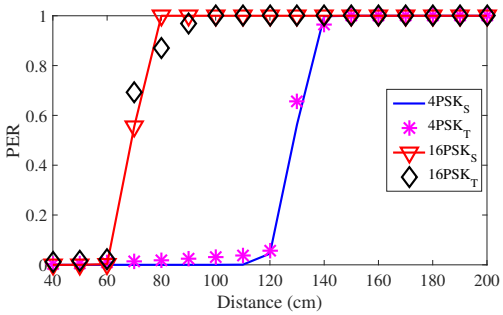


Fig. 14. Simulator and Testbed plot of PER using PSK scheme with 4 and 16 modulation order for scenario $i$.

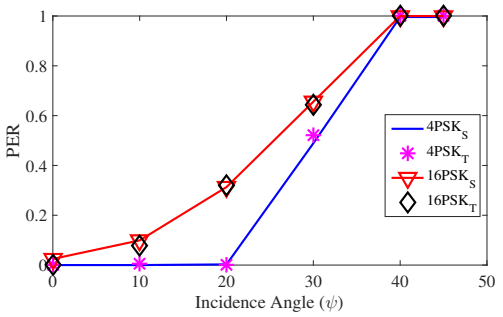


Fig. 15. Simulator and Testbed plot of PER using PSK scheme with 4 and 16 modulation order for scenario $ii$.

many factors such as the number of transmitted bits, memory, operating system (OS), etc.

### C. Packet Error Rate (PER) Analysis

Another important metric we present in this paper to validate receiver sensitivity is the PER. The same test configuration shown in Fig. 5 and the parameters in Table IV are used for this purpose. The PER of the testbed is determined by

$$PER_{testbed} = \frac{\text{number of packets with error}}{\text{number of received packets}}. \quad (15)$$

In our simulator, the PER is related to the BER as stated in (12). Therefore, the packet corruption was done according to the received BER value as explained earlier. The PER results for scenario $i$) and $ii$) using PSK and QAM modulation schemes with different modulation orders are presented in Figures 14-17. The testbed started with higher packet loss

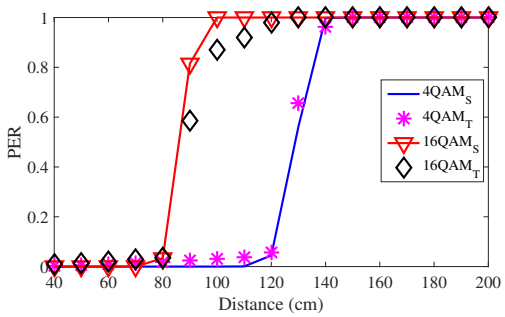


Fig. 16. Simulator and Testbed plot of PER using QAM scheme with 4 and 16 modulation order for Scenario $i$.

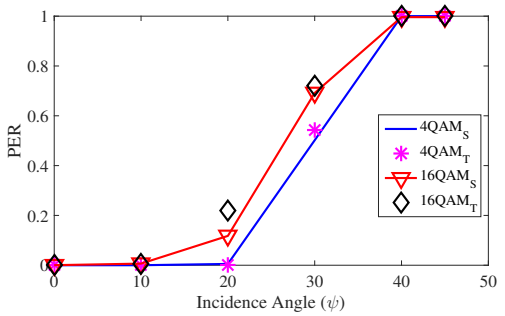


Fig. 17. Simulator and Testbed plot of PER using QAM scheme with 4 and 16 modulation order for scenario $i$.

compared to the simulator but as distance increases the simulation agrees with the testbed performance. Higher modulation schemes like $16PSK$ and $16QAM$ have more bits/symbol than $4PSK$ and $4QAM$ and as a result, have higher PER and BER as shown in the results. The overall results show agreements between VLC simulator and the testbed.

### D. Goodput Analysis

In this section, we compare the performance of the experiment with that of the simulator in terms of goodput. The parameters in Tables IV and V are used for scenario $i$ and $ii$. Here, the simulator goodput is defined by (13). In order to find the optimal goodput, we keep the same setting used previously and start with low data rate. The data rate is increased gradually until we observe the maximum number of received good packets at the receiver side. In addition, to know the data rate in GNURadio, we extract this from the baud rate. The baud rate or the symbol rate is defined as the number of changes per second of a signal or a symbol. The translation between baud and data rate is given by [28] as follows:

$$R = Baudrate \times log2(M). \quad (16)$$

Here, $R$ is the data rate and the sample rate can be set in GNURadio to achieve the required baud rate.

In the simulator, we define the data rate according to the testbed data rate. This corresponds to $400kbps$ for 4PSK and 4QAM, and $600kbps$ for 16PSK and 16QAM. We drop the packets according to the value of the PER given by (12). This

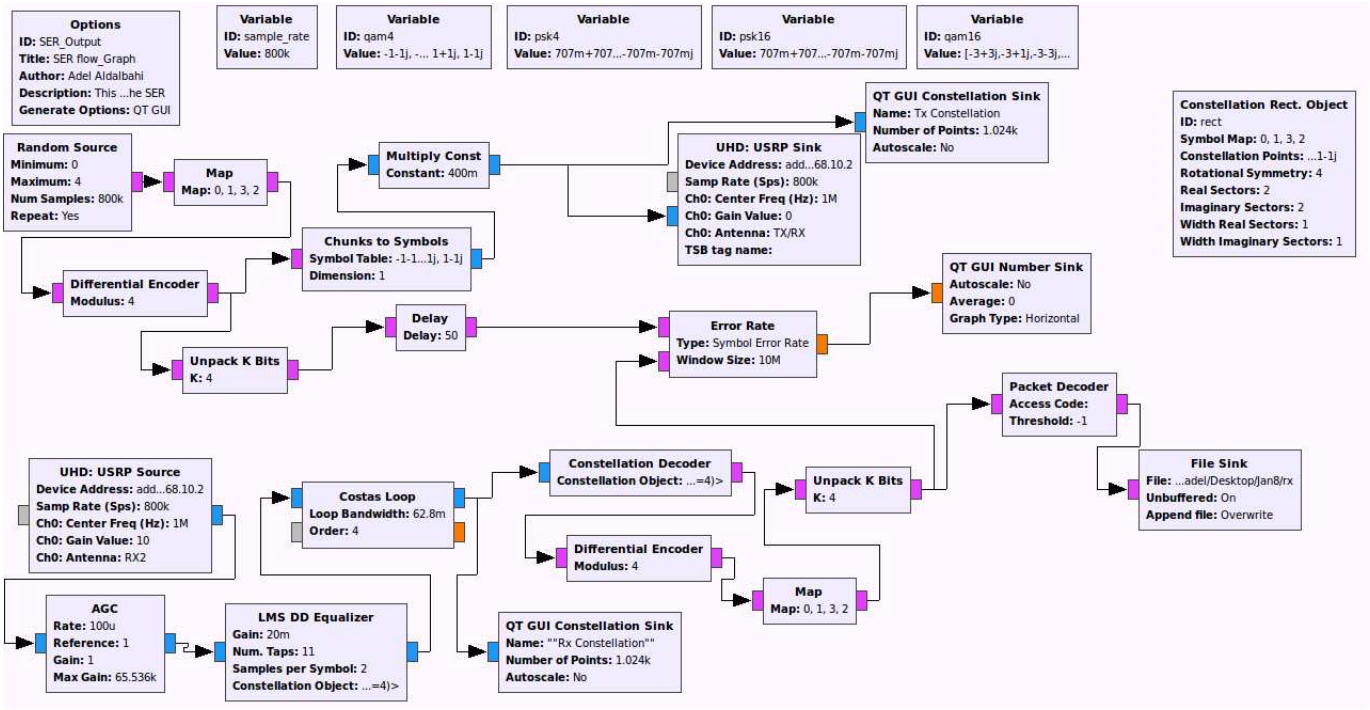


Fig. 18. GNURadio Flow Graph for Signal processing to Measure SER.

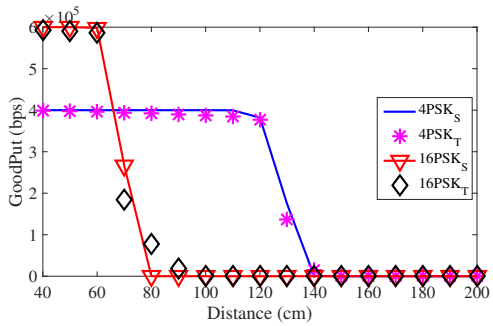


Fig. 19. Scenario $i$ Goodput results of Simulator and Testbed using PSK with 4 and 16 modulation order.

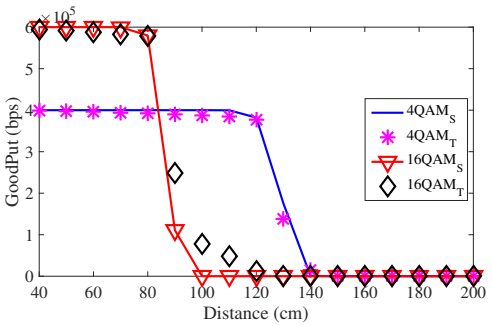


Fig. 21. Scenario $i$ Goodput results of Simulator and Testbed using QAM with 4 and 16 modulation order.

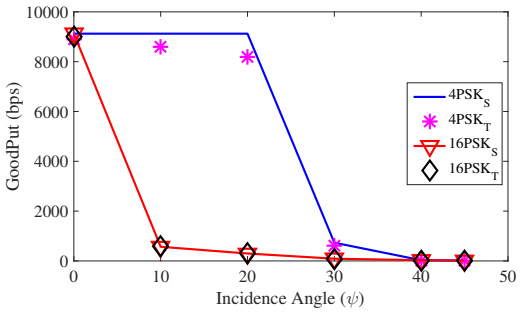


Fig. 20. Scenario $ii$ Goodput results of Simulator and Testbed using PSK with 4 and 16 modulation order.

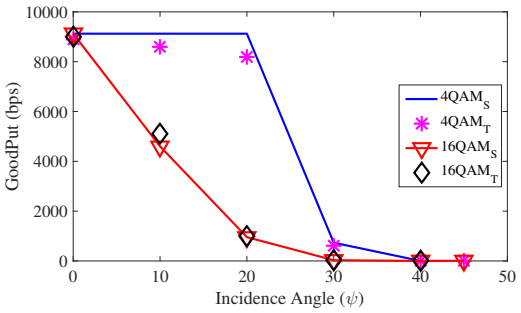


Fig. 22. Scenario $ii$ Goodput results of Simulator and Testbed using QAM with 4 and 16 modulation order.

TABLE VI
TRANSMITTER AND RECEIVER COORDINATES

| Transmitter | Receiver |
|---|---|
| $LED_1(4.75, 1.25, 2.7)$ | $R_1(4.9, 1.25, 0.8)$, $R_2(4.5, 1.25, 0.8)$ |
| $LED_2(4.75, 4.75, 2.7)$ | $R_3(5.1, 4.75, 0.8)$, $R_4(4.2, 4.75, 0.8)$ |
| $LED_3(3.0, 1.25, 2.7)$ | $R_5(3.1, 1.25, 0.8)$, $R_6(2.84, 1.25, 0.8)$ |
| $LED_4(3, 4.75, 2.7)$ | $R_7(3.2, 4.75, 0.8)$, $R_8(2.6, 4.75, 0.8)$ |
| $LED_5(1.25, 1.25, 2.7)$ | $R_9(1.4, 1.25, 0.8)$, $R_{10}(1.0, 1.25, 0.8)$ |
| $LED_6(1.25, 4.75, 2.7)$ | $R_{11}(1.3, 4.75, 0.8)$, $R_{12}(1.1, 4.75, 0.8)$ |

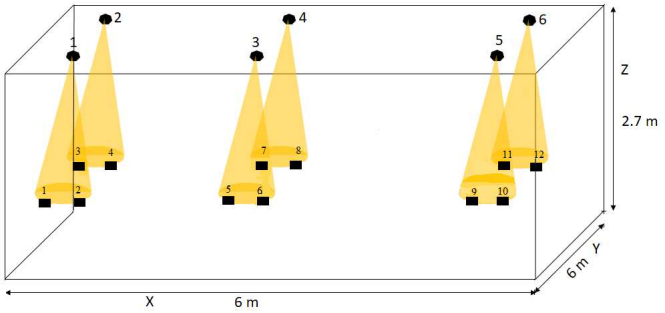


Fig. 23. VLC System model used to simulate distribution of illuminance and SNR.

is done by dropping a number of packets that is equal to the received PER amount which is related to BER.

The simulator and testbed goodput results for scenario $i$ and $ii$ using PSK and QAM schemes are presented in Figures 19-22. The results show that the testbed starts with lower goodput than the simulator. This is can be due to underestimation of the total noise while increasing the distances between the transmitter and receiver or when rotating receiver's incident angle. In addition, in the simulator, we account for the propagation delay but additional delay caused by USRP or GNURadio blocks are out the scope of this paper. The overall results show that the simulator is mostly agreeing with the testbed in both cases.

### *E. Evaluation of a Large WiFi/VLC System*

In this section, we first use our simulator to evaluate the performance of a large VLC system using the testbed parameters at the physical layer. In the second phase, we simulate the performance of the hybrid WiFi/VLC system and compare the results to that of the WiFi alone system at the higher layers.

Consider a room of length $6m$, width $6m$ and height $2.7m$ with $six$ LEDs installed on the ceiling and 12 receivers which can be mobile devices equipped with photodetectors, as illustrated in Fig. 23. In this scenario, we assume that every LED is serving two and only two receivers. The transmitters and receivers coordinates are given by Table VI-E.

The key point here is that each LED is assumed to consist of 16 LUW CN5M LEDs with total luminous flux of 720 lumens $(lm)$ and efficacy of 73 $lm/W$ to match that of the testbed. In addition, each LED is modulated with on-off keying $(OOK)$. The receivers are placed on the desk at hight of $0.8m$ from the receiver surface. The effective area and responsivity of

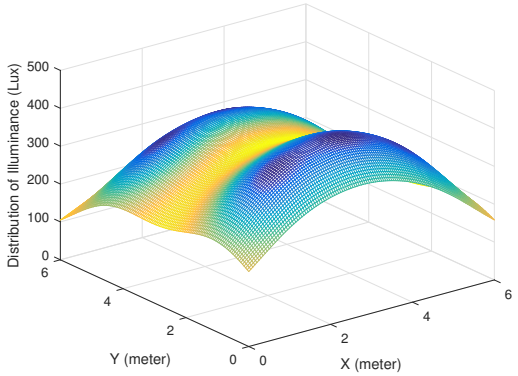


Fig. 24. Distribution of illuminance with respect to receiver locations.

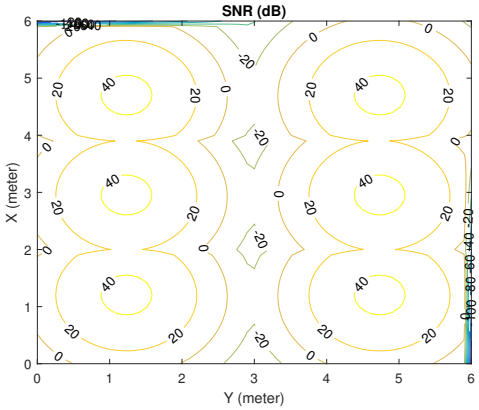


Fig. 25. SNR counterplot with respect to each LED and serving users location.

the photodiodes are shown in Table V, similar to that of the testbed.

Given the coordinates of the transmitter and the receiver as $(x_t, y_t, z_t)$ and $(x_r, y_r, z_r)$, the emission angle is given by

$$\phi = arccos\left(\frac{|z_t - z_r|}{\sqrt{((x_t - x_r)^2 + (y_t - y_r)^2 + (z_t - y_r)^2)}}\right). \quad (17)$$

The illuminance (in lux) at any point of $(x_r, y_r, z_r)$ is given by [21]

$$I = L\frac{(m+1)cos^m(\phi)cos(\psi)}{2\pi d^2}. \quad (18)$$

where L is the luminous flux of an LED (in lumen). Fig. 24, depicts the distribution of illuminance of the receivers in Fig. 23 within their respective locations as given by Table VI-E (assuming $\psi \neq \phi$). We can see that these LEDs have a maximum of more than 300 $lux$ which satisfies the minimum typical lightning requirement due to the use of LUW CN5M type LEDs. Furthermore, the SNR counterplot based on these LEDs is shown in Fig. 25 with respect to the users locations given by Table . Higher SNR values are located right below the center of the LEDs. A very low SNR appears in the middle of the room.

Next we compare WiFi versus a hybrid system utilizing VLC for the downlink and WiFi as an uplink. We use the

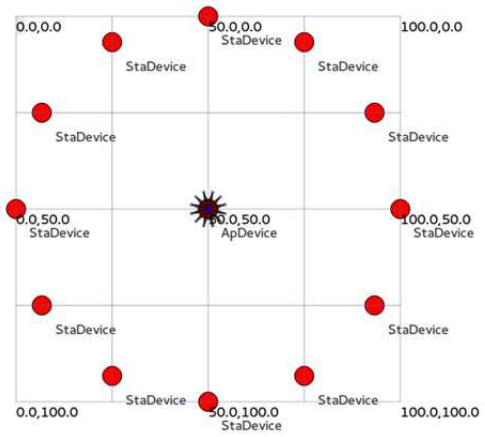


Fig. 26. Network model to simulate Average Throughput of users using different WiFi protocols (i.e., UDP, TCP and UDP+TCP) and Hybrid WiFi/VLC. Actual distance in simulation is set to 2 m.

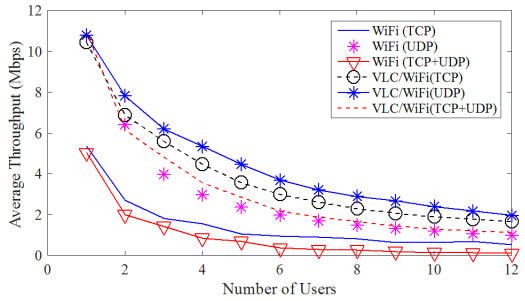


Fig. 27. Comparison of Average Throughput vs number of users for i) WiFi/VLC, ii) TCP, iii) UDP and iv) TCP-UDP systems.

network configuration shown in Fig. 26. The network consists of 12 nodes and a single access point. We run four simulations based on this configuration. In the first set, all receivers use TCP as their communication protocol, whereas we use UDP in the second run. In the third test, $six$ of the receivers use TCP while the other $six$ use UDP. In the last set, we change the downlink of the 12 receivers to VLC while WiFi remains for the uplink. This uplink is also tested using TCP, UDP and TCP-UDP for fair comparison with the previous sets. These systems are compared in term of average throughput versus number of users.

Fig. 27 shows the average throughput versus the number of users achieved by the four systems mentioned above assuming an 11 Mbps WiFi and 2 m link. Using the network illustrated in Fig. 26, we tested the average throughput as the number of WiFi users increases. To measure the average throughput. different sockets (i.e., TCP or UDP depending on the tested scenario) are attached to the nodes. ns3 $flowmonitor$ is used to compute the throughput at each node.

From the results shown in Fig. 27, as the number of users increases, the WiFi performance decreases sharply due to contention in the three systems studied (TCP, UDP, TCP+UDP). In the case of the hybrid WiFi/VLC system, contention is limited to the WiFi uplink. Therefore, the hybrid system performs better than the WiFi alone because the contention happens on the uplink while the downlink is freely assigned to VLC. This both demonstrates the utility of the ns3 VLC module and shows the potential for hybrid WiFi/VLC systems.

## VII. Conclusion

Visible light communications as a new physical medium shows good promise to help alleviate pressure on the use of RF spectrum. However, we have lacked good tools to understand and predict performance of VLC when integrated into an end-to-end networked system. This work establishes an ns3 module for integrating VLC links into larger simulations including other combinations involving heterogeneous WPAN, LAN, or mobile carrier technologies.

This paper presents an ns3 module for VLC that is derived from existing optical wireless/VLC physical device, channel, and modulation models that can be applied to study stand-alone VLC systems, or used in concert with WiFi or other RF models to study complex and large-scale hybrid networks.

The model is validated through comparison with a real VLC testbed implementation using GNURadio software-defined radio and USRP hardware linked to LED emitters and PD receivers. Results indicate good agreement between the model simulations and the measured performance of the real system, with expected variations due to the characteristics of the experimental setup and nuances of practical devices.

Based on this new ns3 VLC module we expect to study new combinations of VLC physical media and modulation formats with existing and future RF technologies such as powerline communications and mmWave to better understand, predict, and optimize future hybrid wireless access networks.

## VIII. Acknowledgment

This work was supported in part by the Engineering Research Centers Program of the National Science Foundation under NSF Cooperative Agreement No. EEC-0812056 and by NSF CNS-1617645, CNS-1617924, and EEC-1560131.

## References

[1] Z. Ghassemlooy, W. O. Popoola, and S. Rajbhandari, "Optical wireless communications: system and channel modelling with matlab®," 2012.

[2] S. Arnon, J. Barry, G. Karagiannidis, R. Schober, and M. Uysal, *Advanced optical wireless communication systems*. Cambridge university press, 2012.

[3] S. Arnon, *Visible light communication*. Cambridge University Press, 2015.

[4] M. Ayyash, H. Elgala, A. Khreishah, V. Jungnickel, T. Little, S. Shao, M. Rahaim, D. Schulz, J. Hilt, and R. Freund, "Coexistence of wifi and lifi toward 5g: concepts, opportunities, and challenges," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 64–71, 2016.

[5] M. H. Kabir, S. Islam, M. J. Hossain, and S. Hossain, "Detail comparison of network simulators," *International Journal of Scientific & Engineering Research*, vol. 5, pp. 203–218, 2014.

[6] H. Nguyen, J.-H. Choi, M. Kang, Z. Ghassemlooy, D. Kim, S.-K. Lim, T.-G. Kang, and C. G. Lee, "A matlab-based simulation program for indoor visible light communication system," in *Communication Systems Networks and Digital Signal Processing (CSNDSP), 2010 7th International Symposium on*. IEEE, 2010, pp. 537–541.

[7] F. Miramirkhani, M. Uysal, and E. Panayirci, "Novel channel models for visible light communications," in *SPIE OPTO*. International Society for Optics and Photonics, 2015, pp. 93 870Q–93 870Q.

[8] B. Tomaš, "Visible light communication physical layer design for jist simulation," *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, vol. 22, no. 341, pp. 41–46, 2014.

[9] C. Ley-Bosch, R. Medina-Sosa, I. Alonso-González, and D. Sánchez-Rodríguez, "Implementing an ieee802. 15.7 physical layer simulation model with omnet++," in *Distributed Computing and Artificial Intelligence, 12th International Conference*. Springer, 2015, pp. 251–258.

[10] E. Sarbazi and M. Uysal, "Phy layer performance evaluation of the ieee 802.15. 7 visible light communication standard," in *Optical Wireless Communications (IWOW), 2013 2nd International Workshop on*. IEEE, 2013, pp. 35–39.

[11] M. B. Rahaim, A. M. Vegni, and T. D. Little, "A hybrid radio frequency and broadcast visible light communication system," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE, 2011, pp. 792–796.

[12] C. Lee, C. Tan, H. Wong, and M. Yahya, "Performance evaluation of hybrid vlc using device cost and power over data throughput criteria," in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2013, pp. 88 451A–88 451A.

[13] H. Chowdhury, I. Ashraf, and M. Katz, "Energy-efficient connectivity in hybrid radio-optical wireless systems," in *Wireless Communication Systems (ISWCS 2013), Proceedings of the Tenth International Symposium on*. VDE, 2013, pp. 1–5.

[14] Z. Huang and Y. Ji, "Design and demonstration of room division multiplexing-based hybrid vlc network," *Chinese Optics Letters*, vol. 11, no. 6, p. 060603, 2013.

[15] S. Shao, A. Khreishah, M. Ayyash, M. B. Rahaim, H. Elgala, V. Jungnickel, D. Schulz, T. D. Little, J. Hilt, and R. Freund, "Design and analysis of a visible-light-communication enhanced wifi system," *Journal of Optical Communications and Networking*, vol. 7, no. 10, pp. 960–973, 2015.

[16] Z. Wu, "Free space optical networking with visible light: a multi-hop multi-access solution," Ph.D. dissertation, BOSTON UNIVERSITY, 2012.

[17] A. Calisti, "Simulation of visible light communications in vehicular networks," Ph.D. dissertation, 2014.

[18] C. B. Liu, B. Sadeghi, and E. W. Knightly, "Enabling vehicular visible light communication (v2lc) networks," in *Proceedings of the Eighth ACM international workshop on Vehicular inter-networking*. ACM, 2011, pp. 41–50.

[19] B. Tomaš, H.-M. Tsai, and M. Boban, "Simulating vehicular visible light communication: Physical radio and mac modeling," in *Vehicular Networking Conference (VNC), 2014 IEEE*. IEEE, 2014, pp. 222–225.

[20] W. Liu, X. Wang, W. Zhang, L. Yang, and C. Peng, "Coordinative simulation with sumo and ns3 for vehicular ad hoc networks," in *Communications (APCC), 2016 22nd Asia-Pacific Conference on*. IEEE, 2016, pp. 337–341.

[21] T. Komine and M. Nakagawa, "Fundamental analysis for visible-light communication system using led lights," *IEEE transactions on Consumer Electronics*, vol. 50, no. 1, pp. 100–107, 2004.

[22] J. M. Kahn and J. R. Barry, "Wireless infrared communications," *Proceedings of the IEEE*, vol. 85, no. 2, pp. 265–298, 1997.

[23] M. Rahaim, A. Miravakili, S. Ray, V. Koomson, M. Hella, and T. Little, "Software defined visible light communication," in *Wireless Innovation Forum Conference on Communications Technologies and Software Defined Radio (WInnComm SDR)*, 2014.

[24] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[25] M. Rahaim and T. D. Little, "Reconciling approaches to snr analysis in optical wireless communications," in *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*. IEEE, 2017, pp. 1–6.

[26] "Point-To-Point solutions description," accessed: 2016-011-10. [Online]. Available: https://www.nsnam.org/doxygen/group/point-to-point.html

[27] "GNURadio-tutorial solutions description," accessed: 2016-011-1. [Online]. Available: http://gnuradio.org/redmine/projects/gnuradio/wiki/Tutorials.html

[28] L. Frenzel, "Whats the difference between bit rate and baud rate," *Electronic Design, April*, vol. 20, p. 12, 2012.


**Adel Aldalbahi** received the B.S degree in Electrical Engineering from Virginia Commonwealth University, Richmond, VA, in 2011 and M.S degree in Power and Energy System from New Jersey Institute of Technology in 2013, Newark, NJ. He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology, Newark, NJ. His current research interests include visible light communication, Wireless networks, modeling and analysis. He is a student member of the IEEE.

**Michael Rahaim** is an Assistant Professor in the Department of Electrical and Computer Engineering at University of Massachusetts Boston. His current research focuses on next generation wireless networks, software defined radio, and heterogeneous integration of wireless technologies including RF, OW, and VLC. Dr. Rahaim received his BS degree in electrical and computer engineering from RPI in 2007, and his MS degree and PhD degree in computer engineering from Boston University in 2011 and 2015. He is a Member of the IEEE and of the IEEE Communications Society.

**Abdallah Khreishah** received his Ph.D and M.S. degrees in Electrical and Computer Engineering from Purdue University in 2010 and 2006, respectively. Prior to that, he received his B.S. degree with honors from Jordan University of Science & Technology in 2004. During the last year of his Ph.D, he worked with NEESCOM. In Fall 2012, he joined the Electrical and Computer Engineering department of NJIT as an Assistant Professor and promoted to Associate Professor in 2017. His research spans the areas of wireless networks, visible-light communication, vehicular networks, congestion control, cloud & edge computing, and network security. His research projects are funded by the National Science Foundation, New Jersey Department of Transportation, and the UAE Research Foundation. He is currently serving as an associate editor for the International Journal of Wireless Information Networks. He served as the TPC chair for WASA 2017, IEEE SNAMS 2014, IEEE SDS -2014, BDSN-2015, BSDN 2015, IOTSMS-2105. He has also served on the TPC committee of IEEE Infocom 2017, IEEE Infocom 2016, IEEE PIMRC 2016, IEEE WCNC 2016, IEEE CCH 2016, IEEE PIMRC 2015, and ICCVE 2015. He is the chair of the IEEE EMBS North Jersey chapter.

**Moussa Ayyash** received his B.S., M.S. and Ph.D. degrees in Electrical Engineering. He is senior member of the IEEE. He is currently a professor at Chicago State University (CSU) in the Information Studies Department. He also serves as the Director of the Center for Information and National Security Education and Research at CSU. His research interests span digital and data communication areas, wireless networking, visible light communications, Internet of Things, and interference mitigation. He is currently on the technical program committee and reviewer for many IEEE conferences and journals.

**Thomas D.C. Little** is a professor in the Department of Electrical and Computer Engineering at Boston University. He is associate director of the National Science Foundation Smart Lighting Engineering Research Center, a collaboration of Rensselaer Polytechnic Institute, the University of New Mexico, and Boston University. His recent efforts address research in pervasive computing using wireless technologies. This includes video streaming, optical communications with the visible spectrum, and applications related to ecological sensing, vehicular networks, and wireless healthcare. He received his B.S. degree in biomedical engineering from RPI in 1983, and his M.S. degree in electrical engineering and Ph.D. degree in computer engineering from Syracuse University in 1989 and 1991, respectively. He is a senior member of the IEEE, a member of the IEEE Computer and Communications Societies, and a member of the Association for Computing Machinery.