Shine: A Step Towards Distributed Multi-Hop Visible Light Communication

Lennart Klaver and Marco Zuniga
Embedded Software Group
Delft University of Technology, The Netherlands
Email: L.P.Klaver@student.tudelft.nl, m.a.zunigazamalloa@tudelft.nl

***Abstract*—Visible light communication (VLC), a novel technology that enables standard Light-Emitting-Diodes (LEDs) to transmit data, is gaining significant attention. In the near future, this technology could enable devices containing LEDs –such as car lights, city lights, screens and home appliances– to form their own networks. VLC, however, is currently limited to point-to-point communication. To unleash VLC's full potential, we need to provide it with more sophisticated networking capabilities. In this paper, we present the design and implementation of a novel platform aimed at distributed multi-hop visible light communication. Compared to the state-of-the-art, our platform provides similar data rates and coverage, but adds two unique characteristics: (i) 360° coverage, which is necessary to investigate an important property of LED communication: directionality, and (ii) a flexible design, which allows our platform to be connected to many experimental boards such as Arduino, Beaglebone, Raspberry Pi and sensor nodes. To quantify the communication capabilities of our board, we evaluate three key components: link quality, neighbor discovery and packet forwarding. Overall, we hope that our work will lower the entry barrier for members of the pervasive and networking communities to investigate and exploit future LED-based networks.**

I. Introduction

For the last century, radio frequency (RF) has dominated the world of wireless communication. But RF has been a victim of its own success. The ever growing popularity of wireless embedded devices is rapidly consuming the available radio bandwidth. To alleviate the bandwidth saturation problem, researchers are exploring other parts of the electromagnetic spectrum (not to replace radio but to complement it). Among the options available, the visible light spectrum is becoming an attractive alternative because it is free, safe, unused and 10,000 times larger than radio. Furthermore, due to recent advancements in the area of visible light communication (VLC), standard light emitting diodes (LEDs) can now be used to transmit data at rates ranging from Kbps to Gbps.

Visible light communication is an important step towards exploiting a new pervasive device (the LED), but it is currently limited to point-to-point communication. Broadly, the work in VLC can be classified into two groups: advanced hardware and modulation methods to maximize the throughput of a link, which can allow desk lamps to transmit high-definition video at Gbps [1], [2], or simpler methods to transmit lower data rates, which can be used to achieve accurate indoor localization using the lighting infrastructure [3], [4]. Most of these studies have a fundamental assumption and a fundamental restriction –which fit the point-to-point communication paradigm. The assumption is that the transmitters and receivers are guaranteed to be within each other's coverage, either by placing them in the right direction (desk lamps) or by making sure that the mobile entity remains within coverage (indoor localization). The restriction is that nodes can not communicate beyond their 1-hop neighborhood. However, *due to the increasingly pervasive presence of LEDs in our environments, limiting VLC to point-to-point communication could hinder its potential.*

The mounting global pressure to improve the use of our energy resources[1] is leading to a rapid increase in the number of LED-based devices. And it is not only residential and commercial lighting that is being replaced with LEDs, a number of other objects such as car lights, city lights, billboards, smartphone and laptop screens, price tags, toys and home appliances, are also using LEDs to reduce their energy consumption. *Thus, considering that VLC can potentially transform any LED device into a wireless transmitter; in the near future, there may be a new generation of objects waiting to be networked in a distributed and multi-hop manner*. The *aim* of our work is to raise awareness about the research opportunities that (future) LED networks can open, and to present a platform that allows investigating those research opportunities. More concretely, the specific contributions of our work are:

- We delve into what has been mainly Physical Layer work in visible light communication, and take a ubiquitous computing perspective to identify the key elements and methods to design a board that is amenable for the pervasive and wireless networking community. (Section II).
- We introduce Shine, a platform that hides the low level complexities of visible light communication while exposing two unique features compared to the state-of-the-art: the ability to investigate the directional coverage of LEDs, and a flexible interface to connect to various platforms. The hardware schematics and accompanying software is freely available to the research community at itpweb.nl. (Section III).
- We evaluate three basic communication characteristics of Shine at the Data Link and Network Layers: link quality, neighbor discovery process and packet forwarding. (Section IV).
- Being aware that distributed multi-hop VLC is a nascent research area, we present open problems and opportunities where Shine can be used. (Section V).

[1] In 2012, residential and commercial lighting accounted for 12% of the total electricity consumption in the U.S. [5]

II. Background, Related Work and Design Considerations

To design a platform for pervasive computing applications, we had to start by looking into the many different ways in which VLC can be achieved, so the most appropriate hardware and methods can be selected. In this section, we first summarize the reasons why LEDs are becoming an attractive alternative for wireless communication, then we describe the options available for VLC transceivers (and the reasons behind the choices made in our design), after that, we position our work within current efforts aimed at providing VLC with networking capabilities, and finally, we describe a motivating application and the limitations of low-end VLC boards.

A. The relevance of LEDs for wireless communication

Communication with light is an old idea, dating back to the 19^{th} century[2] and used in various projects including the famed Smart Dust in 2001 [6], where millimeter-sized devices communicate via laser using a complex array of mirrors. In spite of its potential, the idea of widespread communication using the visible spectrum never really took off, but it is making a comeback due to three important reasons. First, LEDs are becoming a pervasive infrastructure. While other forms of free space optical communication have been used for decades, namely infrared and lasers, LED is the only technology currently permeating our daily environments (and at a very fast pace). Second, modern LEDs have the ability to modulate light intensity at very high rates. Contrary to the traditional incandescent bulb –which involves the slow process of heating a wire before releasing light– the semiconductor materials used in LEDs give *direct* access to the release of photons (light intensity). This phenomenon speeds up dramatically the transmission of data. Third, visible light does not pose any health hazard to human beings, which is not the case for other competing alternatives such as laser. Overall, LEDs are well positioned to be the best way to exploit the idle spectrum around visible light, and hence, we chose them over infrared and laser technologies for our board.

B. The transmitter: choosing the modulation method

Conveying information within a signal requires modulation. Broadly, the modulation methods available for LEDs can be classified into: amplitude modulation and frequency modulation. Overall, in visible light communication, amplitude and frequency modulation have opposite advantages and disadvantages. The main advantage of amplitude modulation techniques –namely On-Off-Keying (OOK), Pulse Time Modulation (PTM) and Pulse Amplitude Modulation (PAM)– is simplicity. On the other hand, frequency modulation techniques –such as Frequency Shift Keying (FSK), Phase Shift Keying (PSK) and Orthogonal Frequency Division Multiplexing (OFDM)– require more advanced hardware and methods. The disadvantages of amplitude modulation are more susceptibility to noise and lower data rates as compared to frequency modulation.

Among the various modulation methods available, we select On-Off-Keying (amplitude modulation) for two reasons: (i) it is a popular method with a good average performance [7], [3], [8] and (ii) its simplicity keeps the number of components low, which is central to fit multiple LED transmitters in a board. As we will describe in the next section, having multiple LEDs is key to investigate the issue of directionality, which is yet an open problem in VLC.

[2]The first wireless telephone, circa 1880, used modulated light beams.

C. The receiver: choosing the photodetector

While on the transmitter side the most widely used element is the LED, on the receiver side we have three popular alternatives: cameras, photodiodes, and LEDs themselves.

Cameras have many pixels, which allows them to focus on a specific source of light [9]. Their disadvantages are the heavy processing required to retrieve data from the images, and more importantly, their low frame rate which limits the data rate. High speed cameras can overcome this problem, but currently they are too big and power hungry.

Photodiodes have the advantage of being very sensitive to light, which allows them to obtain high data rates. To achieve these rates a dedicated circuitry for amplification, filtering and sampling is required.

LEDs can also be used as receivers, because when an LED is *off*, impinging light generates a small current, similar to the effect seen in photodiodes. With the appropriate circuitry, a single LED can operate as both transmitter and receiver (though not simultaneously).

While using a single LED for transmission and reception is an elegant solution, we decided to use photodiodes as receivers due to their higher sensitivity, the less complex circuitry required, and the ability to provide full duplex communication.

D. State of the art on networked VLC

Until the last couple years, VLC research was mostly focused on getting faster links at the Physical Layer. There were no studies trying to integrate VLC nodes into larger networks. Recently, the community has spotted the need to connect VLC devices to other networks and to each other. Next, we describe the state-of-the-art on networked VLC and highlight the novelty of our work. Table I summarizes the characteristics of the SoA platforms including our own.

Clique topologies. A series of studies have looked into the design of Physical and MAC Layers for clique (single-hop fully connected) topologies in VLC [11], [12], [7]. These studies use LEDs as receivers, and propose a novel time synchronization method to avoid light flickering. The MAC is based on the well known CSMA/CA paradigm and it is tested using up to six devices in a clique.

Connecting to the TCP/IP stack. The studies related to "clique" topologies look into connecting VLC devices among themselves. Recently, Wang *et. al.* [8] used a similar LED-to-LED platform to connect VLC devices to the TCP/IP stack. Their work allows standard networking commands such as *ping* and *iperf* to be used seamlessly with VLC nodes. This advancement allows individual VLC devices to be part of the larger Internet.

Our work builds on top of the above mentioned studies in the following way: we use the same modulation method (OOK) and obtain comparable data rates (1kbps) and transmission

Platform	Schmid *et al.* [7] Jul. 2013	Wang *et al.* [8] Sep. 2014	Yang *et al.* [10] May 2014	***Shine***	Tsonev *et al.* [2] Apr. 2014
Data Rate	800bps	2.2Kbps	Unknown[a]	1kbps	3.7Gbps
Transmission Range	∼2m	∼1m	∼6m	∼1m	∼5cm
Modulation	OOK	OOK+RLL	Manchester	OOK	OFDM
Transmitter	Single LED	Single LED	Luminaires	Multiple LEDs	Single LED
Receiver	same as tx.	same as tx.	Photodiode	Multiple photodiodes	Photodiode
Directional Routing	No	No	No	Yes	No
Network capabilities	Star topology	TCP/IP	Linear relay	Distributed Multi-hop	None
Host Platform	Arduino	Beaglebone	Custom	Generic/(Arduino)	Custom

TABLE I: Comparison with the state-of-the-art. Shine is aimed at distributed multi-hop routing and it is not tied to a single host platform. The last column represents the high-end spectrum of VLC Physical Layer research, which uses sophisticated hardware.

[a] The authors do not claim a data rate explicitly, but it seems to be in the Kbps range.

ranges (1m). We differ however in three important ways: Shine (i) is generic and not tied to a specific platform, (ii) consists of many LEDs and Photodiodes to explore the issue of directionality, and (iii) is aimed at multi-hop communication, instead of the single-hop communication targeted thus far.

Chain topologies. Yang and Pandharipande use luminaries to form a chain topology [10]. When a luminary can not reach another one directly, it uses intermediate luminaries to relay the information by leveraging the reflective nature of the floor. Our work is closer in spirit to this study. We strengthen this type of work by releasing a *general* platform with the underlying *"directional"* support to study *any* type of multi-hop topology, static and dynamic.

E. Limitations of low-end VLC platforms

Due to the resource constrained nature of most embedded systems, the work tackling the low-end part of VLC devices (including our own), use simple methods and hardware that only achieve a few kbps and a short range. For comparison purposes, Table I shows what sophisticated VLC platforms can achieve [2]: six orders of magnitude higher data rates, albeit at shorter distances.

F. A motivating application for distributed multi-hop VLC

Shine is a first step towards investigating the role of LEDs in future SmartCities. Some initial studies have shown vehicle-to-vehicle communication using a camera on one car and LEDs on the license plate at another car [9]. We want to investigate a more complex communication network (in a miniature set up) where the LEDs of cars and street lights form networks that exchange information about speed, mass, size and driving patterns to achieve tasks such as collision warning, adaptive cruise control and trajectory determination. Shine is a basic building block that will allow us to investigate the necessary network stack to form this new kind of LED networks.

III. THE SHINE PLATFORM

From a networking perspective, we set two overarching goals for Shine. First, to expose the unique properties of the VLC Physical Layer to the upper layers of the network stack, and second, to make the platform as generic as possible. To describe how Shine achieves these goals, we divide this section in three parts: first, we explain how we tackle the issue of directionality (related to the first goal); then, we describe the steps taken to make our platform generic; and finally, we present the software components that bundle the different pieces to provide a simple API for the Physical Layer.

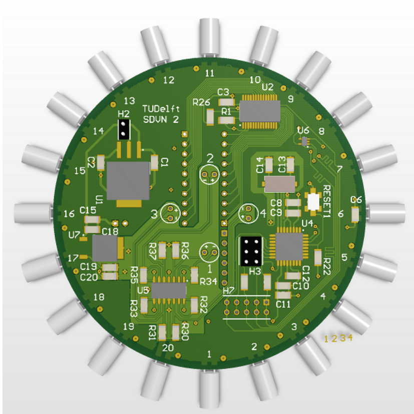
A. Directional communication

Transmitter coverage. One of the most important differences between the coverage capabilities of LEDs and most radios is directionality: LEDs have a narrow main lobe. To analyze the complexity of directional VLC sources, we decided to instrument our platform with as many LEDs as necessary to provide a 360° coverage. In this way, a user can later decide what and how many LEDs to use depending on the application.

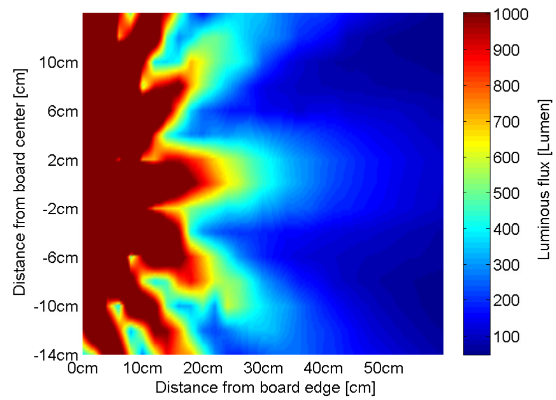
Based on the electrical and electronic components required in our design, the radius of the board came up to be 4cm. This radius allows 20 LEDs to be comfortable placed at the circumference. To provide full coverage we use an LED with an 18° beam and a high-intensity-to-current ratio to extend the transmission range (Avago HLMP-CM1A-450DD). The resulting coverage is depicted in Figure 1. Shine provides a contiguous coverage up to 0.5m and non contiguous coverage up to 1m. Under some settings, for instance a dark room, Shine can reach 2m coverage. The driving circuitry, which modulates the light intensity of each LED, is the same as in [8] and consists of two resistors and one transistor[3]. The circuitry is depicted in Figure 2a. The control of the 20 pins (one for each LED) is done via an onboard microcontroller with UART interface and is explained later.

Receiver coverage. For the receiver end, we selected the Osram SFH203P photodiode. This diode has (i) a 90° angle of incidence, which implies that only four photodiodes are required to get a good 360° coverage, and (ii) a high sensitivity for visible light and a reduced sensitivity for infrared light, which reduces the amount of noise. The driving circuitry consists of an operational amplifier and an ADC. The ADC reports the received information (light intensity) to an onboard processor via a Serial Peripherial Interface (SPI) bus. The schematic of the receiver is shown in Figure 2b. The hardware is capable of simultaneous data decoding on each of the four receiver channels.

[3] To reduce the amount of space and wiring required by each of the 20 transistors, we use transistor arrays.

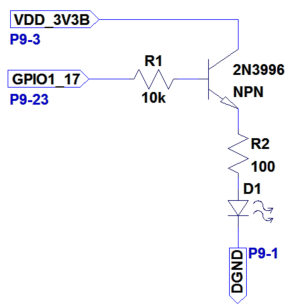


(a) Shine board

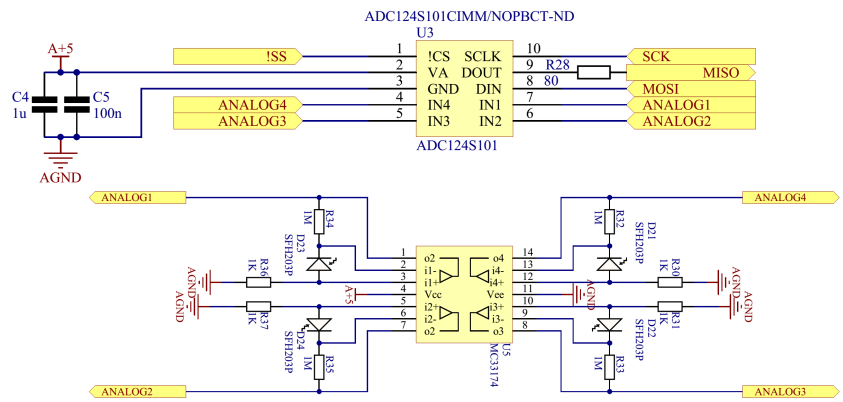


(b) Empirical coverage measured in Lumen

Fig. 1: Transmission Coverage. Shine has 20 LEDs placed in a circular board of 4cm radius (a), and provides contiguous coverage until 50cm (b). Beyond this point, it provides coverage up to 2m if the receiver falls within the main beam of one of the LEDs.



(a) transmitter (leds)



(b) receiver (diodes)

Fig. 2: Electric and electronic circuitry. The transmitter circuit designed for a single LED transmitter in Spice (a), this circuit is replicated 20 times on the board, one for each LED. The receiver circuit presented in Altium (b), is for the four photodiodes.

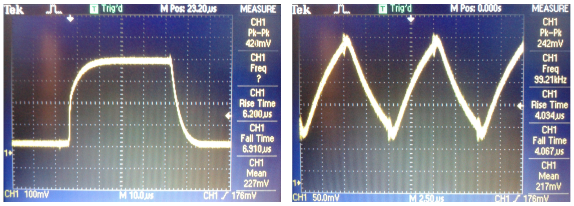
Achievable Rate. We now analyze the achievable rate of our design. This analysis not only elucidates Shine's current data rate, but also exposes opportunities for improvement. There are three macro components that determine Shine's data rate: the sampling rate of the receiver, the communication speed with the processor, and the modulation speed of the transmitter.

The receiver's ADC has a sampling rate of 1MHz. Considering that we have four photodiodes, the sampling rate is reduced to 250KHz. Furthermore, to satisfy Nyquist sampling rate, we require at least two samples per bit, which indicates that the maximum data rate at the receiver is 125 Kbps.

The communication with the processor poses a more stringent constraint than the ADC. The ADC reports 16 bits for every sample. Our processor has a maximum SPI clock speed of 8MHz, and each bit requires one clock tick for transmission, which results in a sampling rate of 500KHz (8MHz/16). Considering that we have to sample four photodiodes with at least two samples per bit, the maximum data rate is limited to 62.5 Kbps (500/4x2).

The main bottleneck of our design is the transmitter. A simple yet correct demodulation of On-Off-Keying requires square waves. We tested our driving circuitry with a function generator (input) and an oscilloscope (output); and Figure 3 shows that when we encode data at speeds faster than 10Kbps, the square waveform starts deforming, which makes demodulation fairly complicated.

Overall, while in theory Shine could achieve 10kbps, in practice, clock drifts and external noise affect the efficiency of the system. We found that a robust performance could be guaranteed for a data rate of 1Kbps, which validates the data rates achieved by comparable platforms (Table I). The main opportunities for improvement are hence to use more sophisticated transistor arrays to drive the LED at faster speeds (hardware) and to improve the timing accuracy of the system (hardware and software).



(a) 10 Kbps (b) 100 Kbps

Fig. 3: Transmitter waveform.

B. A generic platform

Due to the proliferation of embedded systems, researchers are currently using many different platforms to investigate pervasive computing applications: sensor nodes, smartphones, Raspberry Pi, Beaglebone, Arduino, to name a few. Each one of these platforms has unique characteristics in terms of hardware (e.g. processor, ADC, GPIO pins) and software (operating system and real-time capabilities). To avoid developing a platform that is tied to a unique host, we abstracted the necessary hardware and software components into a self-contained platform that provides Physical Layer services to *any* host via a simple serial connection.

Timing. Accurate and fast timing is arguably the most important aspect for modulation and demodulation. If cost and complexity would not be an issue, the best way to tackle this problem is certainly via hardware (FPGAs or ASICs). To reduce cost and complexity, we follow a software solution, as other low-end devices do [7], [8].

When an external platform is used to control LEDs and photodiodes, e.g. BeagleBone, timing needs to be tackled in kernel-space because in user-space VLC communication would need to compete with many other tasks, making timing very unstable. Kernel solutions however are usually tied to the specific platform and OS. To remove timing problems arising from a particular setup, we use an onboard processor exclusively dedicated to timing tasks related to VLC communication.

Serial Interface & Power Supply. The number of LEDs and photodiodes required to provide full coverage, pose two challenges for most microcontrollers: insufficient current level to power the LEDs and insufficient number of pins to control the LEDs and diodes. To solve the power problem, Shine has two onboard power supplies: one high current supply for the transmitters, and a noise reduced power supply for the receivers to limit signal distortion. To control the many number of input/output pins, Shine has the widely supported UART interface. In this manner, most host platforms only need to use two pins to control Shine's Physical Layer.

Arduino compatibility. Our platform is designed to work with any host. But considering that we use the same processor supported by Arduino (Atmel Mega 328p), we also expose the reset and power pins of the UART so the board can be reset to trigger the Arduino bootloader, and hence, use Arduino's libraries. Thus, for low overhead applications, Shine can be used as a fully independent platform running Arduino.

C. Physical Layer

The previous two subsections focused on the hardware required to achieve full directional coverage and to provide a flexible interface. We now describe the software components that enable Physical Layer communication.

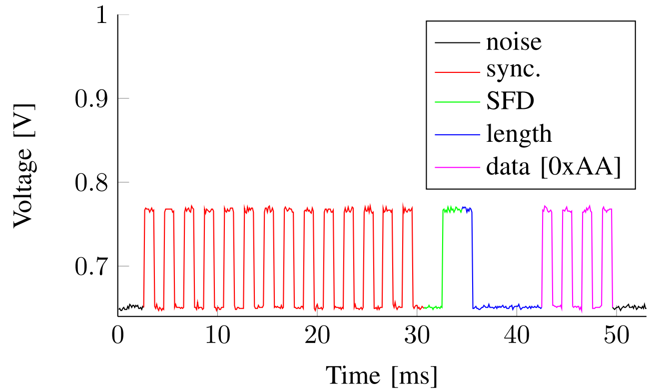


Fig. 4: Physical Layer Data Unit

Data Structure. Following standard practices, our Physical Layer Data Unit (PDU) contains *only* the information required to synchronize the timing of the sender and receiver, and to decode the information (issues such as addressing must be tackled at higher layers). Table II shows Shine's PDU and Figure 4 shows the PDU in a raw VLC signal (Shine uses unipolar encoding to map bits to symbols). The PDU consists of an initial sequence for time synchronization (Sync), a delimiter to denote the beginning of the frame (SFD), a parity check bit (check), the size of the payload (size) and the payload itself (data).

Preamble		**Length field**		**Payload**
Sync	SFD	Check	Size	Data
28 Bits	4 Bits	1 Bit	7 Bits	1..128 Bytes

TABLE II: Physical Layer Data Unit.

Carrier Sensing. The first step required for packet reception is to sense the channel for a valid transmission. Shine achieves this by looking for sharp transitions in the signal strength. Let us denote x_i and x_{i+1} as two consecutive samples captured by one of the four photodiodes (each photodiode is an independent reception channel). A *phase transition* is observed if $|x_i - x_{i+1}| > \Delta_{sense}$. This transition (could) denote the presence of data; for example, in Figure 4 a transition occurs at 3ms with a sudden jump of 100mV. The value of Δ_{sense} determines the sensitivity of the receiver, and it should be high enough to distinguish between noise and data. As an example, Figure 5 shows the changes in the noise floor observed during daylight when the noise floor is sampled at 10Khz. Based on the distributions observed in different scenarios and the resolution of the ADC (1.2mV), we found that Δ_{sense}=24mV is a good value to distinguish most valid signals from noise.

Synchronization. After detecting the existence of a (potential) encoded signal, the receiver needs to synchronize its timer with the sender's timer. This is done during the *sync* phase in the preamble (red phase in Figure 4). Denoting f_s as the

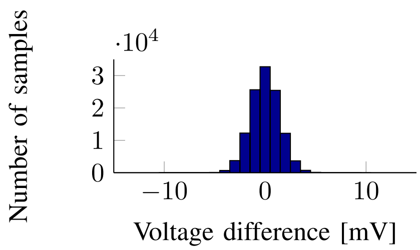


Fig. 5: Carrier sensing threshold (Δ_{sense})

sampling frequency of the receiver (10KHz in our case) and f_d as the data rate of the transmitter (1Kbps), the number of samples s between two consecutive *phase transitions* caused by a bit transmission is given by $s = f_s/f_d$. For our settings, the expected value of s is 10 samples. Upon detecting the presence of a valid PDU, a node should observe continuous phase transitions of length $s = 10$ (due to the alternating 1s and 0s in *sync*). Shine confirms that a valid signal is present if the standard deviation of the observed values s decreases below a threshold of $\Delta_{synch} = 1$. This synchronization process is depicted in Figure 6. Once a valid signal is detected, the mean and standard deviation of the intervals s start changing and reach a steady state of $\mu = 10$ and $\sigma = 0$.

Adaptive symbol thresholding. After the synchronization process is concluded, Shine needs to start decoding each incoming bit. To achieve this, we use an adaptive threshold Δ_{decode} to discern 1s and 0s. If the average signal strength of s continuous samples is above Δ_{decode}, the received symbol is deemed to be a 1, else it is deemed to be a 0. Considering that the intensity of the background light(s) can change while receiving a packet, we use a dynamic threshold Δ_{decode} to decode bits. Let us denote μ_0^i and μ_0^{i-1} as the average signal strength of the last two symbols deemed to be 0, similarly let us denote μ_1^i and μ_1^{i-1} for the last two symbols deemed to be 1. Then Δ_{decode} is defined to be $\frac{(\mu_0^{i-1}+\mu_1^{i-1})+(\mu_0^i+\mu_1^i)}{4}$. Figure 7 showcases the importance of having an *adaptive* threshold to cope with the dynamics in the environment. Around 50ms, the background light changes its intensity, increasing the perceived voltage at the photodiode, but the adaptive threshold is able to follow the trend.

API. Shine's API (developed in C++) provides three types of messages that the host platform can use to build upper layer protocols: transmit, receive and channel check. These messages are sent over the UART interface and use the Request/Response paradigm, where a *request* is always provided with a *response* confirming or denying the request.

For transmissions, we abstract the existence of LEDs and use angles instead, as shown in Table III.

Type	Starting Angle	Ending Angle	Length	Payload
1 Byte	2 Byte	2 Byte	1 Byte	1..128 Byte

TABLE III: Data transmission request message.

The 'angle abstraction' is performed as follows. Shine has a reference LED, named led_0, which determines the 0° angle. If the starting angle of a transmission is set to 20° and the

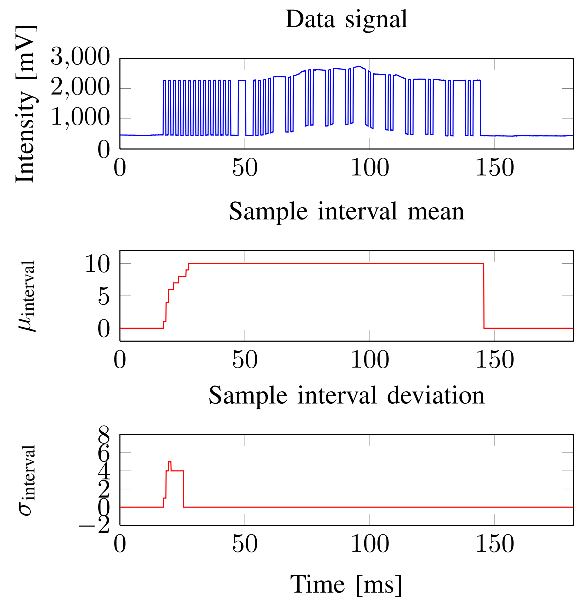


Fig. 6: Synchronization Threshold (Δ_{synch})

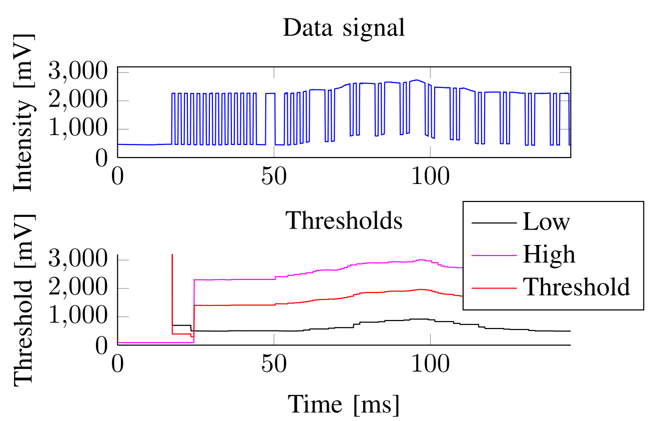


Fig. 7: Decoding Threshold (Δ_{decode})

ending angle to 80°, then leds 1 through 4 will be used for transmissions. Note that if a platform has less LEDs, the same angle will lead to less LEDs being turned on.

For the reception of data, the process is slightly different. Considering that each diode represents an individual channel, the host platform request data from a specific channel and the response includes the angles covered by the requested channel, as shown in Table IV.

Type	Channel
1 Byte	2 Byte

Type	Starting Angle	Ending Angle	Length	Payload	RSS
1 Byte	2 Byte	2 Byte	1 Byte	1..128 Byte	2 Byte

TABLE IV: Data reception messages: request (top) and response (bottom).

Usage. Overall, the use of Shine requires three simple steps: (1) connect Shine to a 12V power source, (2) connect the

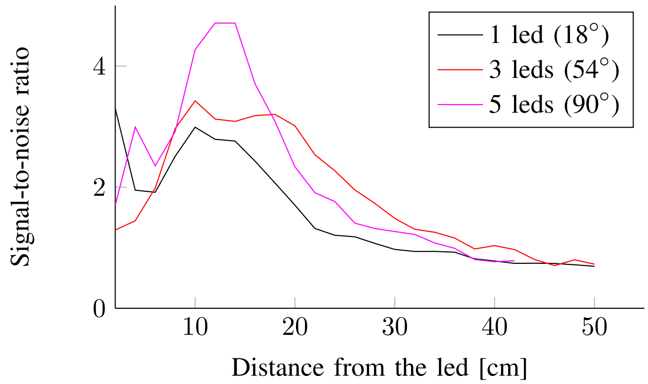


Fig. 8: Link Quality Analysis

two UART pins between Shine and the host platform, and (3) install the Physical Layer API in the host for the development of upper layer protocols.

IV. Evaluation

In this section we present a preliminary evaluation of Shine, where we focus on three aspects: link quality analysis, neighbor discovery and multi-hop communication.

A. Link Quality Analysis

Considering that the transmission request described in Table III does not address individual LEDs but angles, it is important to investigate how transmissions occurring from multiple LEDs affect the SNR (signal-to-noise ratio) of links. We use the standard definition of $\text{SNR} = \frac{V^2_{\text{signal,rms}}}{V^2_{\text{noise,rms}}}$ for our analysis. Assuming a balanced distribution of 1s and 0s in the data signal, the SNR can be simplified to $\text{SNR} = \frac{V_p^2}{2V_n^2}$, where V_p is the voltage given by the ADC when a 1 is received, and V_n is the ADC's voltage for the background light present in the environment.

Figure 8 depicts our results. The experiments were done in a dark room with a constant background light intensity of 5.0 ± 0.8 lumen. We transmitted packets between a sender and receiver for distances up to 50cm. This distance was chosen because it is the maximum *contiguous* coverage that can be provided. If the receiver is positioned in the same direction as the main lobe of a transmitting LED, then a transmission distance of up to 2 meters can be obtained.

The are two trends in Figure 8 that require further explanation. First, while beyond a distance of 15cm the expected decay of electromagnetic signals is observed; between 0 and 15cm, the signal strength is lower than expected. This is due to the saturation at the operational amplifiers, which are calibrated to amplify weak signals. The second, and more important, observation is that adding more LEDs to the same communication channel can have detrimental effects. We observe that when three LEDs are used instead of one, we obtain a higher SNR, but when we further increase the number of active LEDs to five, the SNR decays. Packets are still received, but there is no real gain in SNR. We hyphotesize that this occurs due to destructive interference. At close distances LEDs pointing at different radial directions are likely to increase the SNR due to constructive interference, but as the distance increases, the signals get progressively out of synch causing destructive interference. Contrary to most VLC platforms which consist of a single LED or LEDs facing the same direction, the 360° coverage of Shine will allow us to investigate the problem of directionality.

B. Neighbor Discovery

Contrary to omnidirectional coverage, where neighbor discovery is a not a problem; with directional coverage, identifying a node's neighbors requires a discovery phase [13]. To minimize the discovery time, a trivial solution is to turn *on* all LEDs. Minimizing the energy-consumption is however not as straightforward. First, we will provide a simple mathematical model to minimize the energy consumption during the discovery phase, and then, we will provide empirical measurements validating the model.

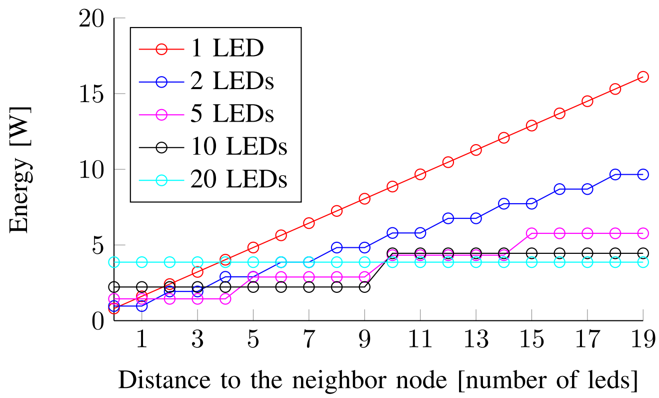
Considering the transmission request in Table III, the neighbor discovery process could be done by checking the entire 360° coverage at once, or polling sector by sector. Denoting C_o as the default energy cost of turning on the platform, C_ℓ as the cost of turning one led, L as the number of leds available in the board, and ℓ as the number of leds used concurrently at each step of the discovery process; the expected cost of discovering a node randomly placed between 0 and 360° using ℓ leds at each step is given by:

$$
\begin{aligned}
E_\ell[cost] &= \sum_{i=1}^{L/\ell} P(\text{discovering node at step i})\text{Cost}_i \\
&= \sum_{i=1}^{L/\ell} \frac{\ell}{L}\left(i(C_o + \ell C_\ell)\right) \\
&= \frac{\ell}{L}(C_o + \ell C_\ell)\left(\frac{L}{\ell}+1\right)\left(\frac{L}{2\ell}\right)
\end{aligned}
$$

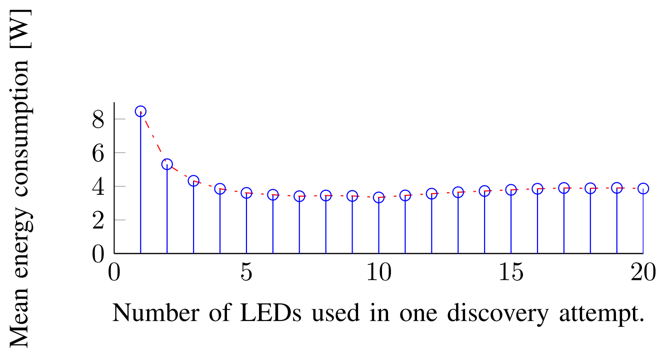
By taking the derivative of $E_\ell[cost]$ with respect to ℓ, we obtain that the number of LEDs that minimize energy consumption is given by:

$$\ell^* = \sqrt{\frac{C_o}{C_\ell}L} \tag{1}$$

To evaluate empirically the neighbor discovery properties of Shine, we developed a simple Data Link Layer in C++. Two Shine nodes were connected via the UART interface to a Laptop and each run an independent network stack. The Data Link Layer is a simple non-persistant CSMA, where the sender first checks if the channel is clear and if so transmits a packet, else it refrains from transmitting. We performed experiments where a receiver was positioned at each of the 20 individual sectors covered by Shine's LEDs, and we used five different LED configurations for the discovery process: turning on 1, 2, 5, 10 and 20 LEDs at a time. Figure 9a shows the results. We observe that the 20-LED line is flat because, independently of the position of the neighbor, all LEDs are turned *on*. The 1-LED case is linear and has a very high slope (i.e. high energy consumption).



(a) Power consumption vs. radial distance



(b) Average power consumption

Fig. 9: Neighbor discovery evaluation

To obtain the ratio $\frac{C_o}{C_\ell}$, we look at positions six and seven in the x-axis of Figure 9a, where the energy cost is the same for the 20-LED and 2-LED cases. This overlap in energy consumption obeys the following equality:

$$C_o + 20\ell = 4(C_o + 2\ell)$$
$$\frac{C_o}{C_\ell} = 4,$$

which based on equation 1, leads to $\ell^* \approx 9$. Considering that $\frac{L}{\ell}$ needs to be an integer, ℓ^* is better set to 10. Figure 9b depicts the average energy consumption when using different number of LEDs, and we observe that 10 LEDs is indeed the optimal number.

It is important to highlight the various trade-offs that Shine is exposed to at the MAC Layer. To minimize the discovery delay, the best option is to use all LED's at once. At the other extreme, to minimize interference, it is better to use one LED at a time. And the point minimizing energy costs lies somewhere in the middle, depending on the $\frac{C_o}{C_\ell}$ ratio. As part of our future work we plan to investigate these trade-offs.

C. Multi-hop Communication

Ultimately, our goal is to form distributed multihop adhoc VLC networks. This goal requires a significant amount of work. At this stage, we choose a toy example to showcase the multi-hop capability of Shine. The setup of the experiment is shown in Figure 10a. The top two nodes have static and known VLC links. When one of these nodes receives a new piece of information, it forwards it to the other node. The node at the bottom needs to discover the first static node within range and transfer a packet. The forwarding mechanism required for this simple setup was done in C++ on top of the MAC layer described in the previous subsection. Figure 10b shows an instance of the experiment. The node at the bottom (Node A) requires two attempts to discover the closest static node in range (Node A used 5 Leds at each discovery step and its reference node had a 90° angle with the closest static node, Node B in this case). After sending an ack, at approximately 600ms, Node B forwards the information to node C. The process takes 1.6s in total.

This simple example is a preliminary test for a miniature SmartCity where communication is needed between a static network (say light posts) and a dynamic one, like cars. For example, a car parking application where, upon arriving or leaving parking spots, cars inform the surrounding street lights about these actions, and the street lights can then exchange this information among themselves or forward it to a particular location.

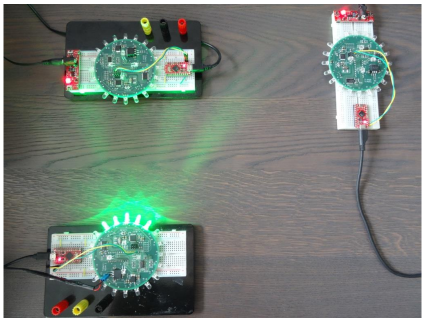
V. Open Research Problems

Besides the principled approach followed in the design of Shine and the engineering effort put in the implementation, we believe that another important contribution of our work is the ample opportunities that Shine opens for future research.

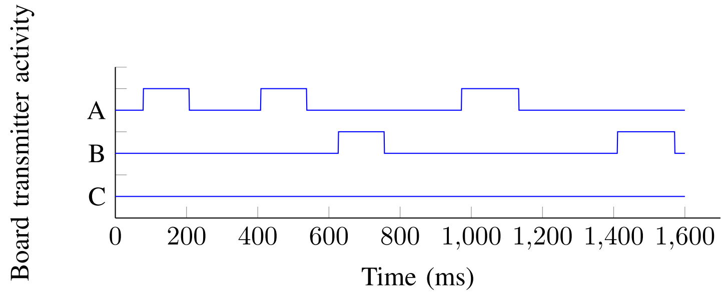
Physical Layer. An immediate opportunity for future work is to improve Shine's range and data rate, while maintaining its simplicity and low cost. For this, we are getting in touch with VLC experts. But the most important challenge at the Physical Layer is to build adequate models and estimators for VLC links. Visible light waves are widely different to radio waves in terms of directionality and multipath effects, and a decade of work in sensor networks research have shown the importance of re-designing MAC and Routing algorithms based on the unique properties of the underlying link [14].

Data Link Layer. As pointed by Singh *et. al.* [13], directionality flips the research focus given to MAC algorithms during the last decades. With traditional omnidirectional sources, MAC research focused on interference avoidance because neighbor discovery was a trivial outcome. With directional transmitters, the problem is reversed: neighbor discovery is the main challenge, while interference is an important but second order problem. Directional communication research is thus far focused on 60GHz radios and it is mainly theoretical. Shine can not only serve as a bridge that benefits from and contributes to the 60 GHz effort, but it would allow empirical insights to improve an motivate further theoretical work.

Network Layer. Multiple directional sources are known to provide advantages and disadvantages. The main advantage is to allow concurrent transmissions, that is, communication with more than one neighbor simultaneously [15]. These concurrent transmissions can increase substantially the capacity of the network. The main disadvantage is that the trade-offs among delay minimization (all-LED transmission), interference minimization (single LED transmission) and energy consumption



(a) Layout



(b) Timeline of events

Fig. 10: Multi-hop evaluation. Node A is the one with the LED's on. Node B is the one on the top left corner, and node C is on the top right. Upon discovering node B, node A transmits a packet to B, which is also forwarded to node C.

minimization (somewhere in between the two extremes) makes routing a complex problem. Hitherto, most work is mainly theoretical [16], and deriving *practical* routing protocols remains an open and challenging problem.

Applications. Besides allowing the investigation of core VLC networking problems, we hope that Shine will enable the exploration of areas related to human computer interactions (HCI), in a similar way to the work carried by Disney Research Labs where VLC technology is used in toy cars [17] and magic wands [12].

VI. Conclusions

Our work was motivated by the observation that, in the near future, devices having LEDs could start forming communication networks of their own. To realize this vision, we need to provide VLC nodes with distributed multi-hop communication capabilities, and to develop such capabilities we require an experimental platform.

After delving into the state-of-the-art of visible light communication, we took a networking perspective to design and implement a novel VLC platform named Shine. Our platform has a performance comparable to SoA alternatives with similar resources (in terms of data rate and transmission range), but is unique in two important aspects: exposing the directionality of LED communication and being generic enough to be connected to various embedded systems. We hope that the unique features of Shine will allow members of the pervasive and wireless networks community to investigate and fully exploit this new type of networks.

Acknowledgment

The authors would like to thank Domenico Giustiniano, Qing Wang and Daniele Puccinelli for their recommendations during the design of our board. We also would like to thank Harald Haas, Thiemo Voigt and Koen Langendoen for the insightful conversations and encouragement to investigate distributed multi-hop VLC. This work has been funded by the European Union 7^{th} Framework Programme (FP7-ICT-2011-8) under grant agreement number 317826 (RELYonIT) and by the Dutch Technology Foundation STW and the Technology Program of the Ministry of Economic Affairs, Agriculture and Innovation (D2S2 project).

References

[1] H. Haas. (2011) Wireless data from every light bulb. [Online]. Available: http://www.ted.com/talks/harald_haas_wireless_data_from_every_light_bulb

[2] D. Tsonev, H. Chun, S. Rajbhandari, J. McKendry, S. Videv, E. Gu, M. Haji, S. Watson, A. Kelly, G. Faulkner, M. Dawson, H. Haas, and D. O'Brien, "A 3-gbps single-led ofdm-based wireless vlc link using a gallium nitride μled," *IEEE Photonics Technology Letters*, 2014.

[3] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," *ACM Mobicom*, 2014.

[4] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: a visible light based positioning system," *USENIX NSDI*, 2014.

[5] U.S. Energy Information Administration. (2012) How much electricity is used for lighting in the united states? [Online]. Available: http://www.eia.gov/tools/faqs/faq.cfm?id=99&t=3

[6] B. Warneke, M. Last, B. Liebowitz, and K. S. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *IEEE Computer*, 2001.

[7] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross, "Led-to-led visible light communication networks," *ACM Mobihoc*, 2013.

[8] Q. Wang, D. Giustiniano, and D. Puccinelli, "Openvlc: Software-defined visible light embedded networks," *ACM VLCS*, 2014.

[9] A. Ashok, M. Gruteser, N. Mandayam, J. Silva, M. Varga, and K. Dana, "Challenge: mobile optical networks through visual mimo," *ACM Mobicom*, 2010.

[10] H. Yang and A. Pandharipande, "Full-duplex relay vlc in led lighting triangular system topology," *ISCCSP*, 2014.

[11] D. Giustiniano, N. O. Tippenhauer, and S. Mangold, "Low-complexity visible light networking with led-to-led communication," *IFIP Wireless Days*, 2012.

[12] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross, "An led-to-led visible light communication system with software-based synchronization," *IEEE Globecom Workshops*, 2012.

[13] S. Singh, R. Mudumbai, and U. Madhow, "Interference analysis for highly directional 60-ghz mesh networks: The case for rethinking medium access control," *IEEE/ACM Trans. Netw.*, 2011.

[14] N. Baccour, A. Koubaa, L. Mottola, M. A. Zuniga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: a survey," *ACM Trans. on Sensor Networks*, 2012.

[15] J. Qiao, L. X. Cai, X. Shen, and J. W. Mark, "Enabling multi-hop concurrent transmissions in 60 ghz wireless personal area networks," *IEEE Trans. Wireless Communications*, 2011.

[16] A. Desai and S. Milner, "Autonomous reconfiguration in free-space optical sensor networks," *IEEE Journal on Selected Areas in Communications*, 2005.

[17] N. O. Tippenhauer, D. Giustiniano, and S. Mangold, "Toys communicating with leds: Enabling toy cars interaction," *IEEE CCNC, Demo Session*, 2012.