

EnLighting: An Indoor Visible Light Communication System Based on Networked Light Bulbs

Stefan Schmid
Disney Research & ETH Zurich
Zurich, Switzerland

Thomas Richner
Disney Research & ETH Zurich
Zurich, Switzerland

Stefan Mangold
Lovefield Wireless
Berne, Switzerland

Thomas R. Gross
Dept. of Computer Science
ETH Zurich, Switzerland

Abstract—The Internet of Things (IoT) envisions that many devices can connect to a network. Visible Light Communication (VLC) based on Light Emitting Diodes (LEDs) is an attractive communication fabric for the IoT, as LEDs are readily available and can serve as transmitters as well as receivers. LED light bulbs, enhanced with photodiodes, provide an attractive path to extend device-to-device communication to room area networking. This paper describes a VLC system, called EnLighting, based on light bulbs that support an embedded Linux version (and its complete networking stack). These programmable light bulbs can both send and receive; they can communicate with objects in a room as well as with other light bulbs nearby. Bidirectional communication allows a light bulb to actively participate in networking and simplifies deployment, maintenance, configuration, and controllability of indoor lighting installations and services. EnLighting supports low-bandwidth communication services in a room (and via a gateway, beyond the room), which provide the base for other applications, e.g., a location service. This paper includes an initial evaluation of a room area communication and localization network based on prototype Linux-enabled light bulbs, reporting the performance for different network traffic types, and shows the benefits obtained from bidirectional communication. This proof-of-concept system illustrates that simple devices can provide an attractive solution to future communication challenges in the saturated radio spectrum.

I. INTRODUCTION

The Internet of Things (IoT) envisions that many devices (appliances, wearables, sensors, utilities, toys, etc) can connect to a network to communicate with data centers, controllers, and other devices. Many of these devices have modest data rate requirements, but all-direction (uplink, downlink, mesh) connections are often a central requirement to adapt and control the device behavior, and to provide network redundancy and communication reliability. However, a communication infrastructure that aims to connect many devices should be low-cost, non-intrusive, and ubiquitous.

Visible Light Communication (VLC) is an attractive choice and has many desirable properties. Light Emitting Diodes (LEDs) allow the construction of low-cost communication systems [1]. VLC does not interfere with the use of the (scarce) radio spectrum, and VLC cannot be easily overheard in another room – to observe a message exchange that is communicated via a VLC channel, the eavesdropping party needs line-of-sight access. This paper presents *EnLighting*, a system of distributed and fully connected LED light bulbs that communicate through free space optics to enable indoor communication and localization services.

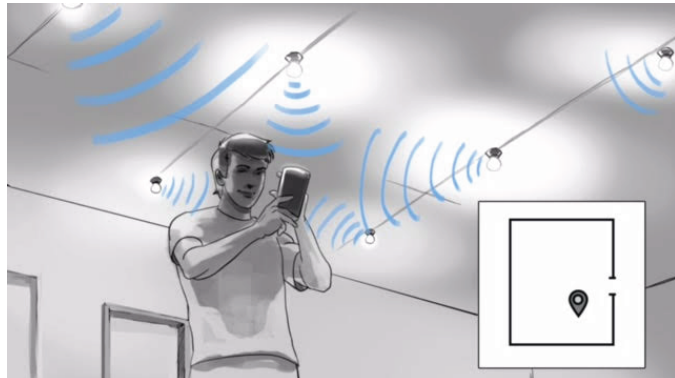


Fig. 1. Concept art (© Disney). Room area network based on VLC-enabled light bulbs providing a localization service.

LED light bulbs combine multiple LEDs and are significantly brighter than single LEDs. Lenses or diffuse bulbs mounted on top of the LEDs transform them into an omnidirectional light source. Such light bulbs are low-cost [2], [3] and have been proposed for many novel indoor applications. But some of the desirable properties for illumination (dispersion of light in all directions, white LEDs) make them unsuitable to directly receive signals from other light sources as proposed in LED-to-LED communication systems [1]. To enable the sensing of incoming signals from other light-emitting devices – light bulbs, single LEDs, or smartphones [4], LED light bulbs can be enhanced with simple light receiving electronics based on photodiodes. Such LED light bulbs are powerful enough to establish a communication link over several meters, but are still based on a software-defined physical (PHY) and Medium Access Control (MAC) layer [5]. These light bulbs can communicate with other light bulbs as well as with low-cost LED-only systems that can be integrated in many devices.

LED light bulbs mounted on the ceiling (or in free-standing floor lamps) easily cover a room, still serving as illumination devices, and at the same time create a room area network that allows data exchange between light-emitting devices. Some light bulbs may even act as gateway to the Internet to provide connectivity beyond a single room. Each light bulb contains a simple System-on-a-Chip (SoC) running an embedded version of Linux. The light bulbs provide the physical communication channel between the devices in the room.

This paper reports the design, implementation and an initial practical evaluation of a room area network based on Linux-enabled light bulbs and shows that simple devices distributed

in a room can provide an attractive solution for today’s communication challenges without relying on the already crowded radio spectrum. The paper is structured as follows and presents the following contributions:

- Description of a low-complex room area network based on Linux- and VLC-enabled light bulbs (Section II).
- Presentation of a testbed architecture with distributed and networked light bulbs and software tools to enable the operation of large-scale light bulb network deployments (Section III).
- A communication link and protocol evaluation for different network topologies: Traffic of the Internet Control Messaging Protocol (ICMP), the User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP) are evaluated and analyzed (Section IV-A).
- Discussion and evaluation of an indoor localization service built on top of the presented communication system. (Section IV-B).

II. SYSTEM DESIGN

This section summarizes the VLC light bulb hardware and software protocol design. Commercial off-the-shelf LED light bulbs are used as a starting point and are then modified to host a SoC running Linux, a microcontroller running the VLC communication protocols, and an additional power supply to support the added electronics. Each light bulb system is built from parts of the original light bulb and additional custom built parts, circuitry and casing. The main motivation for using consumer light bulb as the starting point is they are readily available at low cost and can be used in any lamp providing standard sockets. This leads to an easy-to-setup and flexible testbed with quick replication.

A. System Architecture

A detailed description on how to turn consumer-grade lamps into smart light bulbs that can run an embedded version of Linux is already available [2].

The VLC controller runs the software-based PHY and MAC layer and enables low-level networking between multiple devices. To make use of higher level network protocols, the VLC controller is extended with a SoC running a Linux distribution for embedded wireless systems: OpenWrt [6]. Figure 2 shows the overall system architecture. The microcontroller (MCU) running the VLC controller firmware is connected via the Universal Asynchronous Receiver Transmitter (UART) interface to the SoC. Since the VLC controller is real-time critical, it is kept on a dedicated device, in the same way as other networking hardware.

To simplify the operation of a testbed for such light bulbs, the ability to communicate via another link is needed, in addition to the VLC link. The SoC is also equipped with a Wi-Fi interface, which provides an additional control channel for the testbed operation.

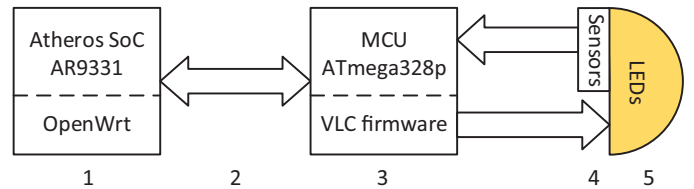


Fig. 2. Light bulb architecture: (1) Wi-Fi-enabled SoC module, (2) communication interface between SoC and MCU (UART), (3) MCU running VLC controller firmware, (4) photodiodes and signal amplification, (5) LED plate.

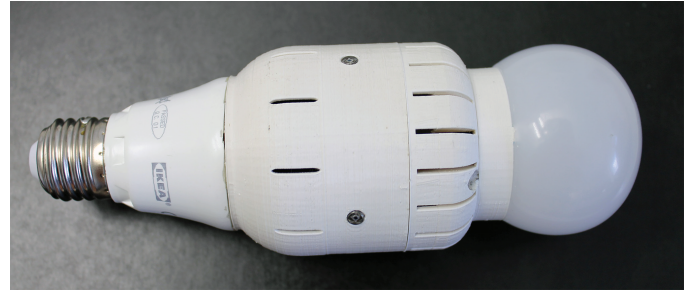


Fig. 3. Fully assembled light bulb containing the SoC running Linux, VLC controller, additional power supply, and four photodiodes. The base (seen on the left) and the diffusion dome (right) are from the commercial LED light bulb, the center section holds the additional electronics, with two photodiodes visible.

B. VLC Controller

The VLC system described in this paper is based on a software-defined and hardware independent platform. It is hosted on a low-cost microcontroller; the PHY and MAC layers are implemented in software. This setup enables a low-complexity system that is flexible and can be ported to different hardware platforms without effort. The optical channel is generated by modulating the light intensity. Pulse Width Modulation (PWM) based on simple ON-OFF Keying (OOK) is used to generate the PHY layer symbols. Hence, it is possible to use only a microcontroller without complicated electronic circuitry to implement the transmitting and receiving path. An LED can directly be connected to the microcontroller’s pins to transmit and receive. If a photodiode is used for receiving, it can either be connected directly to one of the microcontroller’s analog pins (which are then sampled by the controller’s firmware) or it can be fed to an amplifier first. The VLC controller runs the communication protocols, based on the software-defined PHY and MAC layer developed for LED- to-LED networks [1], [5].

1) *Flexible Hardware Platform:* To keep the software protocols hardware (microcontroller architecture) independent, the processors peripherals are abstracted and the actual software is built on top of this abstraction. The hardware abstraction architecture (HAA) consists of three layers between the hardware and application as seen in Figure 4 on the left side. While the adaption and presentation layer are platform specific, the interface layer exposed to the application is platform independent.

The Hardware Presentation Layer (HPL) is a thin layer above the bare metal hardware, it presents the underlying hardware to the programmer in the form of human readable function calls and symbolic memory addresses and definitions.

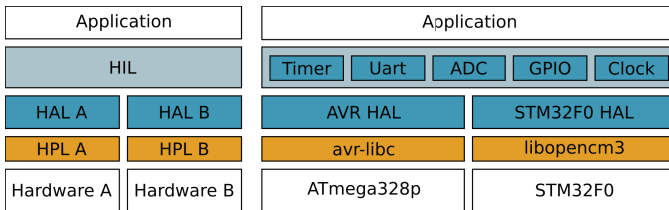


Fig. 4. Left: Hardware Abstraction Architecture based on three layers: Hardware Interface (HIL), Hardware Adaption (HAL) and Hardware Presentation Layer (HPL). Right: HAA applied to two microcontroller platforms.

It does not keep any state and presents only available hardware features to the programmer.

A Hardware Adaption Layer (HAL) is the implementation of the hardware interface for a specific platform, it is allowed to keep state and is built on top of the platform specific HPL. To support a new hardware platform for an application, only the platform specific hardware adaption layer needs to be implemented, as the HPL is provided by the manufacturer, and the Hardware Interface Layer (HIL) as well as the application is platform independent.

The HIL is a platform independent interface to specific hardware components and peripherals; it sits above the HAL. Due to its generic nature, the interfaces are narrow and hide hardware features that may not be available on all platforms or emulate missing hardware features in software on inferior platforms.

Figure 4 (on the right) shows the HAA concept applied to two microcontroller platforms. When switching to a new hardware platform, it is only necessary to create a HAL for the new architecture and available peripherals. The application does not need to be changed since it only depends on the HIL. This decouples the software-based VLC protocols from the hardware, provides flexibility in the choice of hardware platform, and makes it possible to switch to new platforms without effort.

C. Linux Integration

The transparent integration of a VLC communication channel based on the software-defined VLC firmware into the Linux networking stack demands the implementation of a VLC-enabled data link layer. The VLC network driver resides in the kernel space of OpenWrt with root execution privileges. It controls data transfer within the CPU, and handles asynchronous interrupts triggered either by the network device (VLC controller) through the UART interface or by the higher levels of the Linux network stack. It is designed as a loadable kernel module.

The VLC network driver deals with making the VLC controller's MAC layer available to the network protocol suite of OpenWrt by exposing a VLC-enabled Ethernet-class network interface, which is in charge of sending and receiving data packets. The network interface registers itself within specific kernel data structures, to be invoked when packets are exchanged with the outside world. It responds to asynchronous requests received from the outside (incoming packets from the VLC controller), as well as to requests triggered by the kernel for outgoing packets to the MAC/PHY layers, without knowing

how individual transactions mapped to the actual packets are being transferred.

A core part of the VLC network driver's functionality is the communication between Linux and VLC-controller through the UART interface. The serial interface is the communication channel between the SoC that hosts the network driver and the microcontroller where the VLC PHY/MAC layers are implemented. It must be ensured that both, the network driver and the VLC firmware, exchange data packets that are compatible with the format that the other end expects to receive.

III. ENLIGHTENING TESTBED

This section describes the testbed infrastructure and software built for *EnLighting* which is used for software and firmware deployment and protocol debugging and to run measurement campaigns.

A. Testbed Infrastructure

Building, debugging, and running a testbed with devices connected directly to the power grid is dangerous and complex. Wired connections to the light bulbs should therefore be avoided, and light bulbs should be remounted and reassembled as rarely as possible as they are fragile prototypes. These requirements are easiest met by a wireless control channel to enable administrative tasks, efficient debugging, software updates, and logging data from experiments and measurement campaigns. However, the SoC employed to host Linux includes a Wi-Fi interface and provides therefore an attractive avenue to implement a control channel.

All light bulbs are configured to join a dedicated wireless access point enabling remote login into the Linux system running on the light bulbs. Having a wireless control channel also scales for large testbed deployments, where individual devices are further apart and an extensive wired installation is not possible. While remote access to every light bulb gives great flexibility, additional software is necessary to simplify testbed management and to make running experiments more efficient. The software tools are explained in the following subsection.

B. Testbed Software

The testbed software (up to now) consists of two parts: The Command and Control (CnC) panel and the remote logging and visualization system (called Showtime). The CnC panel can be used to retrieve status information about each light bulb, update the firmware or run experiments and collect measurement data using a simple web interface. Showtime (which also runs in a web browser) is used to display live logging information. The VLC controller firmware is heavily instrumented to provide extensive logging about the PHY and MAC layer which is collected in the Showtime view to provide a global overview of the ongoing communication (data and control frames). This real-time visualization tool can give additional valuable insights while running experiments and provide useful information while debugging communication protocols.

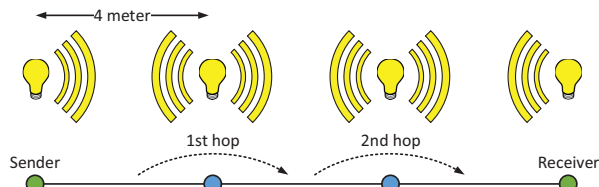


Fig. 5. Testbed setup with four light bulbs, each four meters apart from each other and with the ability to detect light from all directions.

IV. EVALUATION

Two dimensions were considered for an initial evaluation of the communication system based on the light bulb hardware and software described above. First, results for the light bulb communication link are reported. The measurements were conducted at Linux level for different Internet Protocol (IP) and transport layer protocols and for several network topologies. In the second part, a localization service built on top of *En-Lighting* is described and evaluated to illustrate the flexibility afforded by light bulbs that can send *and* receive.

A. Communication and Networking

A simple testbed of four customized consumer light bulbs was used. Figure 5 shows a schematic representation of the testbed setup. The figure also explains the notion of multi-hop communication. If network packets are forwarded by one (two) intermediate node(s), the link is called a one (two) hop communication link. If there is no intermediate node participating, and packets can directly be sent to the receiver, the term direct communication is used.

The testbed setup entails the hidden station problem, since each light bulb can only talk to (and see) its direct neighbors, but the experiments for this evaluation section are designed in a way so that hidden stations are not an issue (not completely true for UDP traffic). Within the scope of this paper, we focus on a proof of concept for IP communication over a VLC channel and show that reliable communication is possible using inexpensive devices. The employed MAC protocol also provides a Ready to Send / Clear to Send (RTS/CTS) scheme to decrease the collision probability if hidden stations are present, but this mechanism is not enabled for the presented measurements. Wang et al. [7] show that there are also different promising approaches to mitigate the hidden terminal problem in VLC.

The following subsections describe the evaluation of direct link and multi-hop topologies for different network traffic types (ICMP, UDP, and TCP) and different communication ranges. Error bars in figures show the standard deviation.

1) *Testbed Setup*: The setup of Figure 5 is realized by a simple testbed shown in Figure 6. Each light bulb is placed in an off-the-shelf floor lamp (torchiere) with a height of 175 cm, and the light bulbs are placed in a line, four meters apart from each other. This setup also illustrates a benefit obtained by using standard light bulbs. Light bulbs can be placed in any available lamp with a fitting socket, and they can be placed anywhere in a room as long as a power connection is available.

2) *ICMP Network Traffic*: To evaluate the direct link performance for different communication distances, ICMP data



Fig. 6. Testbed setup. The light bulbs are placed in floor lamps 1.75 meter above floor level and line up each 4 meters apart.

traffic is used to measure the Round Trip Time (RTT) for bidirectional data exchanges using `fping` (a more powerful version of the standard ping command). Only two of the testbed light bulbs are used for these measurements. All measurement results always show the specific protocol payload and the payload at the VLC MAC layer. Currently, the VLC MAC layer supports payload sizes up to 200 bytes (limited by the low-cost microcontroller running the VLC firmware). Further, due to the low symbol rate, payloads higher than 200 bytes have high error probabilities due to long transmission times. We then vary the distance of the direct link between the two participating bulbs; the results are depicted in Figure 7.

As expected, larger payload size leads to longer RTT, reaching from one to four seconds. The communication link stays stable up to five meters with minimal changes of the RTTs. At five meters, the error bars (indicating the standard deviation) show more variation in RTTs and also a slight increase of the average RTTs. This means that due to the longer distance, signal decoding at the PHY layer fails, and packets are retransmitted by the MAC layer. At distances higher than five meters, the error probability increases profoundly, and we do not explore that space further.

The measurements show that stable communication is possible at up to five meters. To build a room area network, bridging five meters are sufficient to communicate with devices within a (residential) room or small office. To reach further, e.g., to communicate within large open spaces, or to create communication links to neighboring rooms, multiple light bulbs and multi-hop communication can be used.

Figure 8 shows ICMP RTT measurements for a direct link as well as for the one and two hop topologies (as described in the section's introduction), for different packet sizes and a direct link covering a distance of 4 m. To route the network traffic, static routes are used. Since the VLC controller is interfaced via a Linux Ethernet interface, any routing protocol available for Linux (e.g., Optimized Link State Routing) could be used to discover routes (with parameter tuning due to the low system throughput). The results show that RTTs are approximately doubling for every additional hop, and communication stays stable also for larger payload sizes, and we can therefore expect stable communication for a higher number of intermediate hops. Increasing the hop number is likely to introduce hidden stations, and the VLC MAC protocol implements a standard RTS/CTS scheme to decrease collisions in hidden station scenarios for this reason, but a discussion of this feature is beyond the page budget of this paper.

3) *UDP and TCP Network Traffic*: UDP and TCP measurements are executed with the well-know `iperf` tool. Figure 9

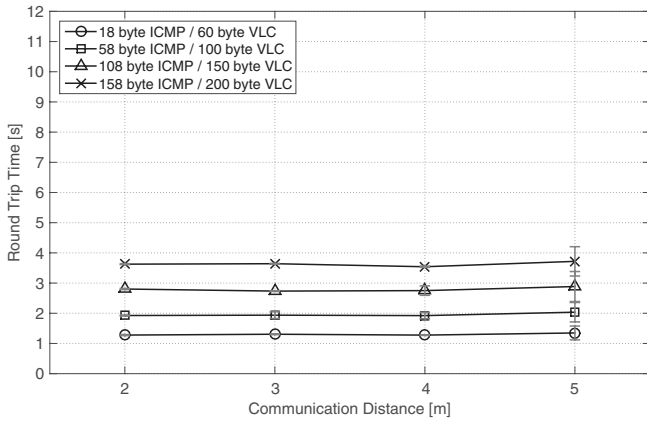


Fig. 7. ICMP RTT measurements for different distances and payload sizes. Payload size is indicated for the current protocol level and VLC MAC layer.

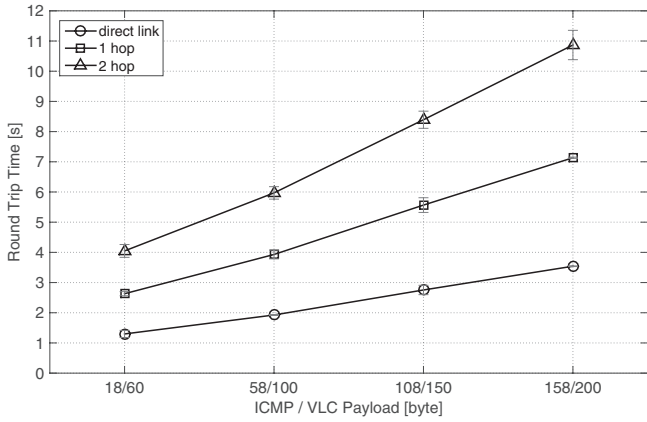


Fig. 8. ICMP RTT measurements for direct link as well as one hop and two hop topologies and different payload sizes. Payload size is indicated for the current protocol level and VLC MAC layer.

shows saturation throughput for a direct link as well as for one and two hop topologies. For small packet sizes the protocol headers are limiting throughput. For a 18 bytes UDP payload, data throughput of around 200 b/s (bits per second) can be reached. For the maximum UDP payload of 158 bytes, a throughput of 600 b/s is possible. Also, for one and two hop links, UDP data transfer is still possible, reaching maximal throughput also at the maximum packet size, i.e., 300 b/s for the one hop and 200 b/s for the two hop scenario.

TCP measurements are conducted for a static TCP window size of one packet, i.e., the next packet is only dispatched if the corresponding TCP acknowledgment (ACK) has been received. The Maximum Segment Size (MSS) is also fixed to 34, 84 and 134 bytes relating to 100, 150 and 200 bytes VLC MAC payload size. TCP generates more overhead than UDP which is visible in the reached maximal data throughput. For a direct link topology, up to 400 b/s are possible. For one and two hops 200 b/s, respectively 150 b/s, are reachable. The throughput is low, but a stable TCP communication channel is available. Since there is always only one packet in the channel (due to the static TCP window of one) there are no collisions caused by hidden terminals.

These measurements in a testbed act as a proof of concept; they show that stable communication over a VLC channel

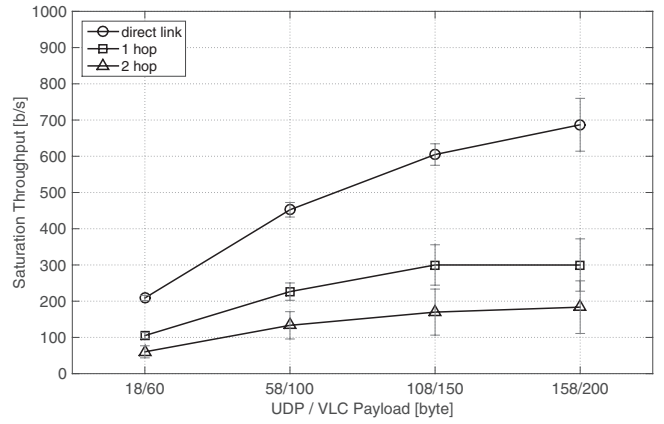


Fig. 9. UDP saturation throughput for direct link, one hop and two hop topologies and different payload sizes. Payload size is indicated for the current protocol level and VLC MAC layer.

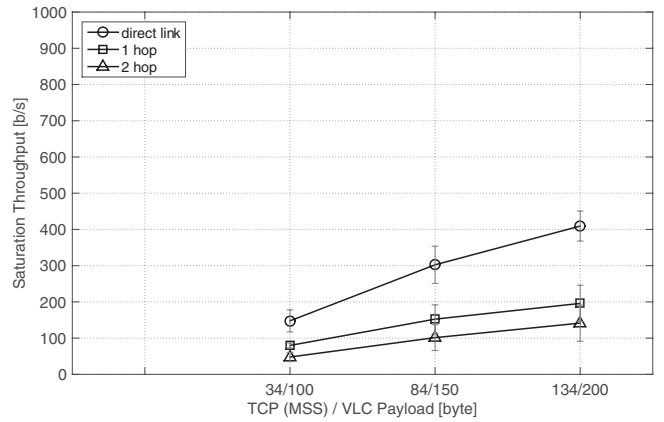


Fig. 10. TCP saturation throughput for direct link, one hop and two hop topologies, static MSS of 34, 84 and 134 bytes, and static TCP window of 1. MSS and VLC MAC payload size is indicated.

using a standard Linux environment and protocols is possible. Employing faster microcontrollers could further improve the physical data rate, and protocol optimization (e.g., reducing protocol overhead for an IoT deployment) could increase the overall system data throughput.

B. Localization

In addition to basic networking, communicating light bulbs can also support other services and we discuss how a sample application (localization) is implemented on top of the networked light bulbs. LED light bulbs have been used for localization in specialized scenarios (see Section V) before. We show here that the light bulb's ability to send and receive provides tangible benefits.

The service is implemented at the Linux level and does not require changes in the VLC firmware. Since the VLC link is transparently integrated into Linux as an Ethernet interface, any technology supporting IP sockets can be used as a base. The application described in the following is written in node.js [8].

Given a room illuminated by the LED light bulbs, we can use the Received Signal Strength Indicator (RSSI) to estimate the distance from the receiver to the transmitting light bulbs. The RSSI is defined as the difference between a low and

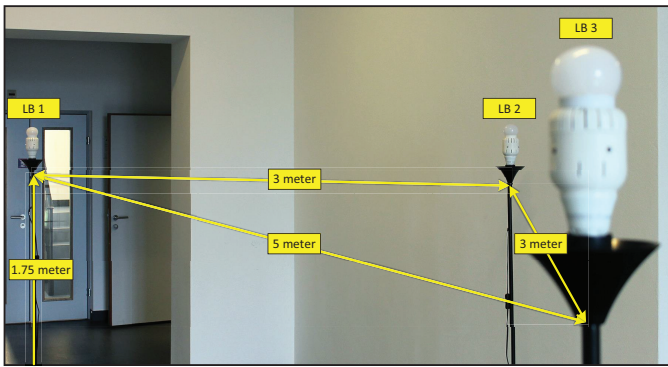


Fig. 11. Localization testbed setup with three light bulbs placed in a triangle, running the localization service software.

high pulse of a symbol in the PHY layer, averaged over a complete data frame. The RSSI is provided by the VLC firmware and the Linux driver module and is available through the `sysfs` interface [9] for any application. The localization service running on every light bulb is repetitively transmitting a beacon with a fixed and configurable time interval. A beacon consists of a UDP packet sent to the broadcast address.

A receiver placed somewhere in the room can now estimate its position by looking at the received UDP packet, which contains the sender address, and the RSSI for the received packet. When receiving beacons from multiple light bulbs, it is possible to refine the estimated location with well known trilateration techniques [10].

1) *Localization Testbed Setup:* The system is deployed as shown in Figure 11. Light bulb 1 (LB1) and light bulb 3 (LB3) are placed six meter apart; light bulb 2 (LB2) is placed 3 meter from the other two bulbs forming a triangle. The light bulbs are placed in standard floor lamps at a height of 1.75 m.

2) *RSSI Quality Measurements:* For the first measurement setup, only LB1 and LB3 are used, LB2 is disabled. With these measurements, the quality of the RSSI value is assessed to determine its behavior under different conditions and whether it is usable for localization or not. The light bulbs are sending location beacons at an interval of one second plus a random jitter between 0 and 200 ms. The communication link between the light bulbs is also enabled, which means they are synchronized [5]. The light bulbs can communicate with each other and receive each other's beacons. A receiving device (same hardware as the light bulbs) is placed at different locations on the direct line between the two light bulbs (from 40 cm to 560 cm). The receiving photodiode is pointed towards the ceiling, not favoring any direction.

Figures 12-14 display measured RSSI values for different receiver locations and three different receiver heights (0, 100, and 175 cm). The measurements shown in one plot were recorded concurrently, i.e., at the same time, and not after each other. E.g., all data reported at the distance of 100 cm from LB1 were collected during the same measurement interval. The RSSI values show a quadratic decay with distance which is plausible for a light intensity based value. The almost in-existing error bars (showing the standard deviation) also underline the stability of the measured values. Since the two light bulbs are synchronized, the MAC layer can handle medium access, and beacons only collide with a small probability.

Therefore beacons from both light bulbs can be received at the same time (one shortly after each other). Even when the receiver is close to one light bulb, it is still possible to receive beacons from the light bulb further away. This feature is useful since any collision does not falsify RSSI measurements, further, any additionally received beacon can help refine the localization process.

Figures 15-17 show the same experiment, but this time with the PHY layer synchronization *disabled*. This setup mimics a VLC system where the light bulbs can send but cannot receive. Then the light bulbs do not support networking and the light bulbs act only as transmitting devices. As a result beacons are not coordinated by the MAC protocol anymore and can collide. The plotted results show that most of the time only *one* beacon from one light bulb is received (from the stronger, closer one). During a collision, the beacon from the closer light bulb might still be receivable due to the capture effect [11], but the suppressed beacon also contributes to the RSSI value. This evidence is visible in the curve shape and the standard deviation depicted by the error bars.

The testbed allows us to shed light on the impact of the absence of communication between the light bulbs. As a result of the missing communication link, beacons are lost (i.e., not received at the receiver). Figure 19 shows how many beacons are received per second, for all receiver distances and heights considered. The beacons from both light bulbs are summed up. When synchronization is enabled, most of the beacons sent are also received, and a high rate of receiving beacons enables accurate localization in the time domain. Without synchronization, most of the beacons collide and are discarded by the receiver. Fewer beacons per second at the receiver side imply inaccurate localization, e.g, if the object to localize is moving.

These initial measurements show that the chosen RSSI metric and VLC system can be used for localization applications. Further, the light bulbs' ability to receive clearly improves RSSI value quality and therefore localization performance.

3) *Three Light Bulbs:* We also report measurements conducted with all three light bulbs shown in Figure 11. We concentrate on a receiver height of 100 cm and enabled synchronization. Figure 18 summarizes the results. To improve the figure's readability, the RSSI values for each light bulb are shown in separate plots, but the data was recorded at the same time. The view is from above (the ceiling); the axes show the x- and y-direction, and the black square denotes the light bulb's position. The results clearly show that the RSSI values are highest closest to the respecting light bulb and decay over distance. The figure also shows that the light bulb networking scales at least up to three devices. When combining the three figures, it is clear that the retrieved RSSI values can be used to determine the receiver's position by applying well-known trilateration techniques.

4) *Practical Issues:* Section IV-B2 discussed the contribution of synchronization to the quality of the localization service. Because light bulbs can receive, the localization service can be extended with useful features for setting up and managing the lighting infrastructure. As long as neighboring light bulbs are in communication range (e.g., on floor lamps or uplights in a room, or when mounted in the ceiling), they

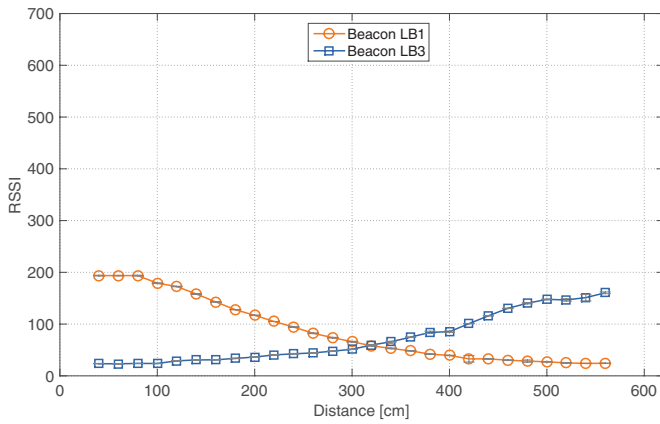


Fig. 12. Measured RSSI between LB1 (at 0 cm) and LB3 (at 600 cm) for a receiver at the floor level (0 cm height).

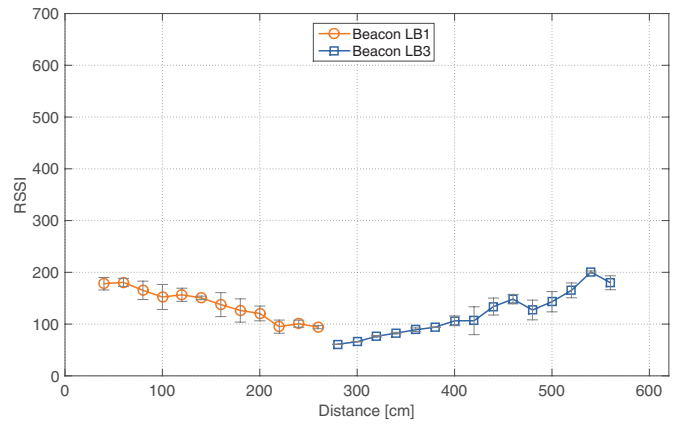


Fig. 15. Same setting as Figure 12 (on the left), but the PHY layer synchronization between the two light bulbs is *disabled*.

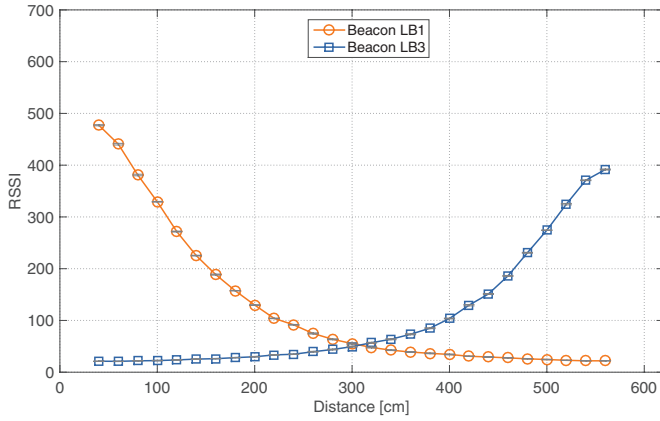


Fig. 13. Measured RSSI between LB1 (at 0 cm) and LB3 (at 600 cm) for a receiver placed at a height of 100 cm.

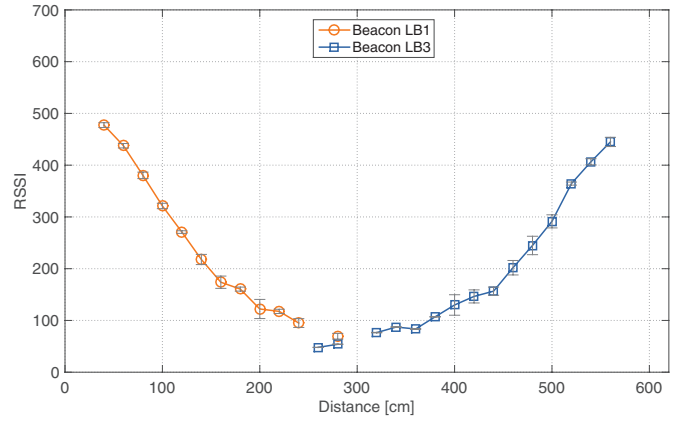


Fig. 16. Same setting as Figure 13 (on the left), but the PHY layer synchronization between the two light bulbs is *disabled*.

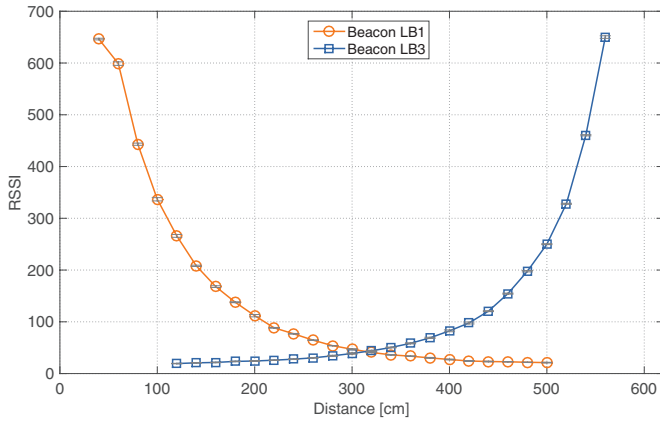


Fig. 14. Measured RSSI between LB1 (at 0 cm) and LB3 (at 600 cm) for a receiver placed at the light bulb level (175 cm height).

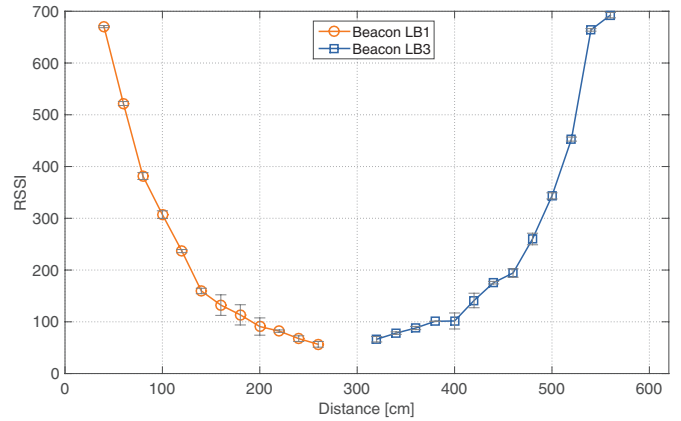


Fig. 17. Same setting as Figure 14 (on the left), but the PHY layer synchronization between the two light bulbs is *disabled*.

are reachable. There is no need for any special wiring of the room (other than to provide power), and configuration can be done through the optical links and does not interfere with any Wi-Fi or other radio system that may be deployed. Furthermore, the light bulbs overhear each other's beacons and therefore can store the transmitter's IP addresses together with the corresponding RSSI value. Light bulbs that do not illuminate a room (but are connected to the power grid) can still receive data; only sending light bulbs illuminate. If the lighting infrastructure provides the localization service and a light bulb

fails and must be replaced, the replacement bulb (when inserted into the failed bulb's socket) can ask the neighbors for the predecessor's address. The neighbors can use their history of received beacons and RSSI values to infer a possible address for the replaced bulb. This capability simplifies maintenance and minimizes configuration errors (in case of single failures which should be the common case). Furthermore, light bulbs can also be configured wirelessly using a VLC link through a mobile light source (e.g., flashlight). Such a flashlight can be programmed with configuration information and simply be

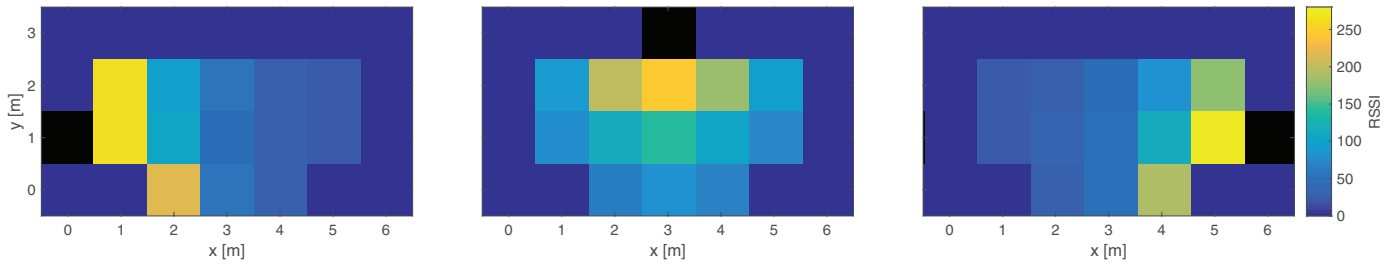


Fig. 18. RSSI values for measurements with three light bulbs. Sight from above. The black square marks the position of each light bulb. From left to right: LB1, LB2, and LB3. The closer to yellow the higher the RSSI and the bluer the closer to zero. The measurements for the same square in each figure were collected at the same time.

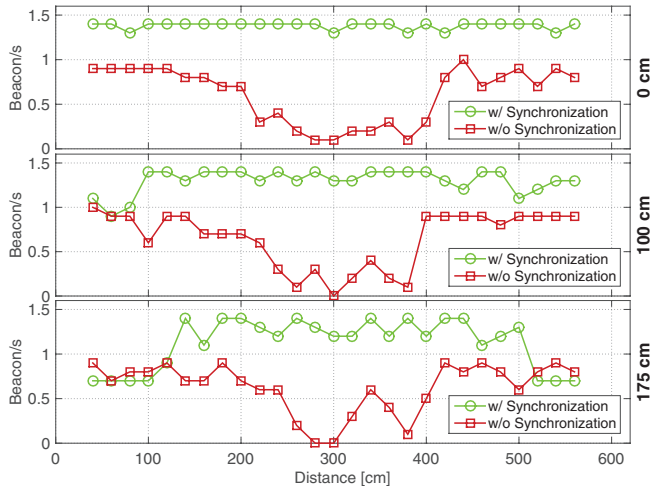


Fig. 19. Received beacons per second for different receiver distances and heights, for enabled and disabled PHY layer synchronization

pointed towards the devices that should be reconfigured or updated, providing an intuitive interface and connectivity.

V. RELATED WORK

Information about the VLC standard IEEE 802.15.7 can be found in [12]–[14]. There are multiple PHY and MAC layers defined, of which some are similar to our software-based implementation. The platform used in this paper compromises performance (mainly throughput) for the benefit of flexible prototyping: Schmid et al. [1] describe a flexible platform suitable for ad hoc communication involving a small number of up to five LED devices, but without IP connectivity. Their enhanced PHY and MAC implementations from [5] together with the practical TCP/IP adaptation presented in [2], are used in the light bulb localization testbed in this paper. OpenVLC [15] presents a simple, low-cost hardware/software platform centered around a printed circuit board (OpenVLC1.0 cape) that implements an optical front-end; its primary application is prototyping MAC and PHY protocols.

Indoor localization (positioning) of mobile devices has so far mainly been approached with Wi-Fi or Bluetooth (Low Energy) beacons that are already available in many residential, office, or commercial buildings (malls, shops) [16]. Accurate and reliable positioning (around one meter), however, requires computational effort such as RSSI and time-of-flight estimates, triangulation and trilateration, and the use of multi-antenna systems [17]. Sometimes, ultrasound sensors are deployed to

complement the radio infrastructure, in cases where higher accuracy indoor localization is required [18].

The idea of exploiting the building lighting as additional data beacons with receiving photodiodes (cameras) at devices to receive location identities from light bulbs, has already been discussed in the literature, and various approaches have been demonstrated [10], [19]–[26].

EnLighting described in this paper differs from such previous approaches as it enables light pattern synchronization and data exchange between light bulbs *directly through the visible light*. There is no additional radio transceiver required: Light bulbs transmit and receive data that is hidden in the visible light. This is an important feature not only to increase the accuracy of indoor localization (as neighboring light bulbs do not interfere with each other’s transmissions) but also to realize visible light networking through neighbor and route discovery [27]. Beacon identities can thus be assigned dynamically in case a light bulb is switched off, replaced, or relocated.

Since the vision of using VLC light bulbs for data communication has been formulated in [28], many research challenges have been addressed in the field of low-cost practical VLC systems [29]–[33].

VI. CONCLUSION

Networked light bulbs can combine illumination and communication: to a human observer, the light bulb is on to illuminate an area. Devices with appropriate sensors can receive data from, as well as send data to, other devices. *EnLighting*, the VLC system described, provides a software-defined hardware independent platform, which can be hosted on inexpensive hardware building blocks. LED light bulbs with photodiodes offer an ideal platform for room area communication networks that allow communication with low-cost LED-only systems.

Current SoC chips are powerful enough to run complex operating systems (e.g., Linux) and combined with an off-the-shelf light bulb result in smart light bulbs that support the IP stack. Without any tuning or customization, such a system provides stable connections and adequate performance for low-bandwidth applications in room area setups. *EnLighting* builds the basis for communication services and can be enriched with custom built applications. This paper reported on an example application for localization that was implemented without much effort.

The light bulbs communicate with each other, and this property is crucial not only to support a wide range of IP protocols but also for application scenarios. As neighboring light

bulbs can synchronize their activities, the number of collisions is reduced and the connections are stable. Furthermore, the ability to synchronize makes the system flexible, there is no need for another control channel or specialized wiring. A side effect of this flexibility is that the light bulbs can be placed everywhere, in floor lamps or mounted on the ceiling.

As the Internet of Things (IoT) is formed, there is clearly a need to support a wide range of devices to exploit the IoT's potential. For applications with low and moderate bandwidth demands, networked light bulbs can set up a room area network that provides connectivity. Supporting the IP stack is important as it allows many other applications that are built on top of this protocol suite to work without modifications. Many issues remain to make the vision of the IoT a reality, but the proof of concept provided by the simple VLC system described in this paper is an encouraging result that shows that the IP suite can be extended to reach the many devices of the IoT.

REFERENCES

- [1] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross, "LED-to-LED Visible Light Communication Networks," in *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '13, (New York, NY, USA), pp. 1–10, ACM, 2013.
- [2] S. Schmid, T. Bourchas, S. Mangold, and T. R. Gross, "Linux Light Bulbs: Enabling Internet Protocol Connectivity for Light Bulb Networks," in *Proceedings of the 2Nd International Workshop on Visible Light Communications Systems*, VLCS '15, (New York, NY, USA), pp. 3–8, ACM, 2015.
- [3] S. Schmid, J. Ziegler, G. Corbellini, T. R. Gross, and S. Mangold, "Using Consumer LED Light Bulbs for Low-cost Visible Light Communication Systems," in *Proceedings of the 1st ACM MobiCom Workshop on Visible Light Communication Systems*, VLCS '14, pp. 9–14, ACM, 2014.
- [4] G. Corbellini, K. Akşit, S. Schmid, S. Mangold, and T. R. Gross, "Connecting Networks of Toys and Smartphones with Visible Light Communication," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 72–78, 2014.
- [5] S. Schmid, G. Corbellini, S. Mangold, and T. R. Gross, "Continuous Synchronization for LED-to-LED Visible Light Communication Networks," in *Optical Wireless Communications (IWOW), 2014 3rd International Workshop in*, pp. 45–49, Sept 2014.
- [6] OpenWrt, "OpenWrt," May 2015. <https://openwrt.org>.
- [7] Q. Wang and D. Giustiniano, "Communication Networks of Visible Light Emitting Diodes with Intra-Frame Bidirectional Transmission," in *Proceedings of the 10th ACM International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '14, pp. 21–28, ACM, 2014.
- [8] Node.js, "Application Framework for Cross-Platform Server-Side Environments." <https://nodejs.org/>, 2015.
- [9] P. Mochel, "The sysfs Filesystem." <https://www.kernel.org/pub/linux/kernel/people/mochel/doc/papers/ols-2005/mochel.pdf>, viewed 09-10-2015.
- [10] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A Visible Light Based Positioning System," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, NSDI'14, (Berkeley, CA, USA), pp. 331–343, USENIX Association, 2014.
- [11] K. Ramakrishnan and H. Yang, "The Ethernet Capture Effect: Analysis and Solution," in *Proceedings of the 19th Conference of Local Computer Networks*, pp. 228–240, IEEE, Oct 1994.
- [12] C. Gavrinca, J. Baranda, and P. Henarejos, "Rapid Prototyping of Standard-Compliant Visible Light Communications System," *Communications Magazine, IEEE*, vol. 52, pp. 80–87, July 2014.
- [13] A. Boucouvalas, P. Chatzimisios, Z. Ghassemlooy, M. Uysal, and K. Yiannopoulos, "Standards for Indoor Optical Wireless Communications," *Communications Magazine, IEEE*, vol. 53, pp. 24–31, March 2015.
- [14] S. Nobar, K. Mehr, and J. Niya, "Comprehensive Performance Analysis of IEEE 802.15.7 CSMA/CA Mechanism for Saturated Traffic," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 7, pp. 62–73, February 2015.
- [15] Q. Wang, D. Giustiniano, and O. Gnawali, "Low-Cost, Flexible and Open Platform for Visible Light Communication Networks," in *Proceedings of the 2nd International Workshop on Hot Topics in Wireless, HotWireless '15*, (New York, NY, USA), pp. 31–35, ACM, 2015.
- [16] A. M. Hossaina and W.-S. Soh, "A Survey of Calibration-free Indoor Positioning Systems," *Comput. Commun.*, vol. 66, pp. 1–13, July 2015.
- [17] K. Joshi, S. Hong, and S. Katti, "PinPoint: Localizing Interfering Radios," in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, NSDI'13, (Berkeley, CA, USA), pp. 241–254, USENIX Association, 2013.
- [18] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, "Accurate GSM Indoor Localization," in *Proceedings of the 7th International Conference on Ubiquitous Computing, UbiComp'05*, (Berlin, Heidelberg), pp. 141–158, Springer-Verlag, 2005.
- [19] M. Liu, K. Qiu, F. Che, S. Li, B. Hussain, L. Wu, and C. Yue, "Towards indoor localization using Visible Light Communication for consumer electronic devices," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 143–148, Sept 2014.
- [20] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, (New York, NY, USA), pp. 447–458, ACM, 2014.
- [21] J. Li, A. Liu, G. Shen, L. Li, C. Sun, and F. Zhao, "Retro-VLC: Enabling Battery-free Duplex Visible Light Communication for Mobile and IoT Applications," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, HotMobile '15, (New York, NY, USA), pp. 21–26, ACM, 2015.
- [22] X. Zhou and A. T. Campbell, "Visible Light Networking and Sensing," in *Proceedings of the 1st ACM Workshop on Hot Topics in Wireless, HotWireless '14*, pp. 55–60, ACM, 2014.
- [23] A. Kumar, A. Mihovska, S. Kyriazakos, and R. Prasad, "Visible Light Communications (VLC) for Ambient Assisted Living," *Wirel. Pers. Commun.*, vol. 78, pp. 1699–1717, Oct. 2014.
- [24] P. Hu, L. Li, C. Peng, G. Shen, and F. Zhao, "Pharos: Enable Physical Analytics Through Visible Light Based Indoor Localization," in *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, HotNets-XII, pp. 5:1–5:7, ACM, 2013.
- [25] T. Komine and M. Nakagawa, "Integrated System of White LED Visible-light Communication and Power-line Communication," *IEEE Trans. on Consum. Electron.*, vol. 49, pp. 71–79, Feb. 2003.
- [26] N. Rajagopal, P. Lazik, and A. Rowe, "Visual Light Landmarks for Mobile Devices," in *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, IPSN '14, (Piscataway, NJ, USA), pp. 249–260, IEEE Press, 2014.
- [27] S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies," *ACM Comput. Surv.*, vol. 36, pp. 335–371, Dec. 2004.
- [28] H. Elgala, R. Mesleh, and H. Haas, "Indoor Broadcasting via White LEDs and OFDM," *Consumer Electronics, IEEE Transactions on*, vol. 55, no. 3, pp. 1127–1134, 2009.
- [29] D. O'Brien, "Visible Light Communications: Challenges and Potential," in *IEEE Photonics Conference*, pp. 365–366, Oct. 2011.
- [30] H. Elgala, R. Mesleh, and H. Haas, "Indoor Optical Wireless Communication: Potential and State-of-the-Art," *IEEE Commun. Mag.*, vol. 49, no. 9, pp. 56–62, 2011.
- [31] S. Mangold, "Visible Light Communications for Entertainment Networking," in *Photonics Society Summer Topical Meeting, 2012 IEEE*, pp. 100–101, July 2012.
- [32] A. Jovicic, J. Li, and T. Richardson, "Visible Light Communication: Opportunities, Challenges and the Path to Market," *Communications Magazine, IEEE*, vol. 51, pp. 26–32, December 2013.
- [33] S.-K. Lim, K. Ruling, I. Kim, and I. S. Jang, "Entertainment Lighting Control Network Standardization to Support VLC Services," *Communications Magazine, IEEE*, vol. 51, pp. 42–48, December 2013.