Continuous Synchronization for LED-to-LED Visible Light Communication Networks

Stefan Schmid
Disney Research & ETH Zurich
8006 Zurich, Switzerland

Giorgio Corbellini
Disney Research
8006 Zurich, Switzerland

Stefan Mangold
Disney Research
8006 Zurich, Switzerland

Thomas R. Gross
ETH Zurich
8092 Zurich, Switzerland

***Abstract*—Off-the-shelf Light Emitting Diodes (LEDs) and low-cost microcontrollers provide the foundation for networking using the visible light as communication medium. These networks require fast and stable synchronization and a distributed protocol to handle shared medium access. We present and evaluate a physical layer and a distributed, contention-based medium access control protocol that enables reliable communication over room-range distances; both are implemented in software using low-cost commercial off-the-shelf building blocks. Experiments with a testbed consisting of embedded devices equipped with only LEDs demonstrate the scalability of this approach. The performance evaluation indicates that Visible Light Communication is a reliable solution for more than ten devices to bring low-cost and non-complex connectivity to a large number of devices.**

I. Introduction

Many consumer devices such as smartphones, tablets, wearables, or toys already embed Light Emitting Diodes (LEDs). LED-based Visible Light Communication (VLC) provides an attractive alternative to radio as a communication medium reusing already existing hardware. LED-based VLC networks have a number of appealing properties: LEDs are low-cost and are already embedded into many consumer devices. LED networks also allow the integration of illumination and communication and make communication visible. VLC, as other forms of low-power optical communication, is also safe: As any transmission can be captured only by nodes with a line-of-sight connection, eavesdropping by other (hidden) nodes is no longer possible. Furthermore, VLC systems do not use the (possibly overcrowded) radio spectrum and do not interfere with existing radio devices.

LED-based VLC exploits that an LED can be used as a receiver by measuring the voltage discharge during a time interval when reverse-biased [1]. Recent advances allow the implementation of a protocol for VLC in software using off-the-shelf microcontrollers [2]. Hence, in LED-based communication systems, LEDs modulate and encode an outgoing message but also measure the incoming light (direct detection) to convert light intensity into electrical signals.

A. Contributions

This paper presents two contributions and evaluates their effectiveness: (1) A physical (PHY) layer focusing on a low-cost and flexible software-defined approach is presented. The only hardware parts needed are a microcontroller and an LED, making the reuse of already built-in components possible. (2) We present a reliable and scalable synchronization technique that takes advantage of the low sensitivity of LEDs when used as light detectors.

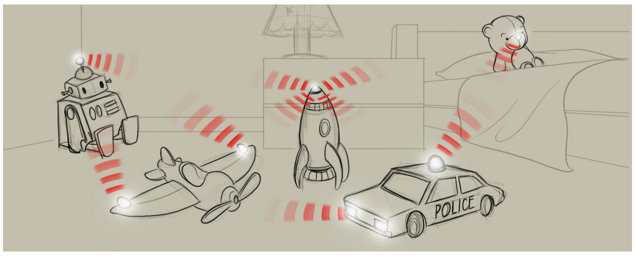


Fig. 1. (© Disney) Different toys or devices can use VLC to communicate at low cost.

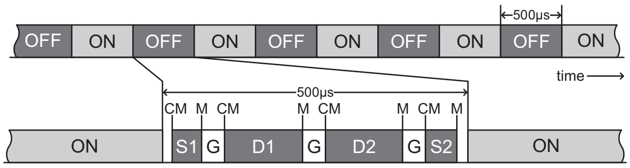
B. Related Work

The domain of VLC faces many interesting research challenges [3]–[5] as there exists a wide range of devices that can be employed. We are primarily interested in simplicity and the reuse of existing hardware components, e.g., use LEDs embedded in consumer products or off-the-shelf LED light bulbs, and rely on software (executed by a simple microcontroller) to investigate proof-of-concept prototypes with an emphasis on low-cost solutions [6]. LEDs as receivers [1] and a simple PHY and MAC protocol can be used for ad hoc communication involving up to five LED devices in a network [2]. However, difficulties with synchronization for scenarios of more than a handful of LED devices in communication range of each other led to the novel synchronization method reported in this paper. The Medium Access Control (MAC) protocol described [2] is a simplified variation of known Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocols such as used in IEEE 802 standards. This paper presents a version that improves fairness and protocol efficiency (as described in Section III). Application scenarios and use cases for consumer electronics have also been demonstrated [7], [8]. Further, LED-to-LED networks might also communicate to smartphones using its camera and flashlight [6].

C. Paper Organization

The paper is organized as follows: Section II introduces the novel VLC PHY layer describing the slot structure and the continuous synchronization procedure. It is followed by Section III, which provides the description of the latest implementation for the VLC CSMA/CA MAC layer. The experimental evaluation of the novel PHY (for a network scenario) is shown in Section IV.

978-1-4799-6676-9/14/$31.00



Abbr.	Description
S1	First synchronization interval
S2	Second synchronization interval
G	Guard interval
D1	First data interval
D2	Second data interval
M	Measure light (start ADC conversion)
C	Charge LED (reverse bias LED)

Fig. 2. OFF slot intervals: There are seven intervals, two for synchronization and two for data reception.

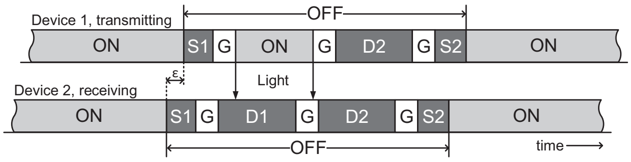


Fig. 3. Guard intervals prevent light leakage into synchronization or data intervals. The transmitting device enables light output during the first data slot to transmit a bit. Since its phase is shifted relative to the receiving device, the center guard slot is collecting all the light which would have been received during the second data slot if no guard slot was present.

II. Physical Layer Protocol

It has been shown that reliable performance in LED-to-LED networks can be realized for small networks with up to six nodes and at short distances of up to two meters [2]. The PHY layer introduced by Schmid et al. [2], however, is based on a discontinous synchronization scheme that does not scale well with the number of VLC devices limiting the communication to small groups of nodes (up to 6) transmitting short messages (up to 50 byte). In the following we discuss a novel approach showing how to obtain continuous stable synchronization and resilient data encoding.

A. LED transceiver

The light output of an LED in forward bias is modulated so that it can transmit a signal. In reverse-biased mode, an LED is able to detect incoming light [1], [2], [9]. Since an LED cannot output light and detect at the same time, a slotted structure must be introduced to alternate between forward bias (ON slot) and reverse bias (OFF slot), as illustrated in Figure 2. Devices can communicate in a network only when the OFF slots are aligned (synchronized) so that all devices detect incoming light at the same time [2]. The PHY layer is based on a slotting structure with guard intervals that enables synchronization during packet transmission (not only during idle mode), to scale the VLC system properties towards higher number of devices and longer packet sizes.

B. Slot structure

Figure 2 outlines the slot partition. OFF slots and ON slots alternate with each other. Each slot has a duration of 500μs. The pattern generated this way is a 1 kHz signal that is perceived by a human as constant light. The slot duration of both ON and OFF slot can be decreased, to generate more OFF slots per time and therefore to also increase the theoretically possible symbol rate, but with the loss of light sensitivity. The OFF slot is partitioned into smaller intervals and structured as depicted in Figure 2. The amount of light received is measured during the two synchronization intervals and the two data intervals. Intervals used to measure light begin with a short charge (reverse bias, 1-2μs) of the LED. Immediately after the charging, the microcontroller starts an analog-to-digital conversion (ADC) to measure the charged voltage. At the end of these intervals, a new ADC is started to assess the remaining voltage. The voltage difference of these two measurements is proportional to the received amount of light and the duration of the interval. The synchronization intervals S1 and S2 mark the start and the end of the OFF slot. The data encoding and decoding scheme can be outlined as follows: Either during D1 or D2 light is received from the transmitting device and can be decoded to 0 or 1 (2-pulse position modulation). To minimize decoding errors caused by imprecise synchronization, guard intervals (G) prevent light leakage into wrong data slots as shown in Figure 3.

C. Synchronization

The two synchronization intervals S1 and S2 can detect light leakage from the previous or the next ON slot, respectively. As long as the amount of light received during S1 and S2 stays the same (within a margin), the devices are in-sync with other measurement slots. As soon as the difference between S1 and S2 increases, the direction of phase correction can be computed (e.g., if S1 receives more light than S2, the OFF slot started too early, and the phase offset must be corrected in the positive time direction). To compensate a phase offset, the following ON slot duration is increased or decreased. The exact phase offset cannot be determined directly from S1 and S2 and is approximated stepwise. The usage of two synchronization intervals prevents that two out-of-sync devices try to correct their phase offset simultaneously in the same direction. When not synchronized to each other, the devices always detect increased light levels in the opposite synchronization slot, and consequently they influence their correction direction pointing towards each other. The synchronization is not only responsible to align two devices in the beginning of the operation, but also to keep the devices synchronized during active operation to compensate clock drifts. To speed up synchronization for completely out-of- sync devices, a method with exponential decay of the phase correction step size is introduced. This achieves fast synchronization for newly paired devices and a stable and smooth correction for already synchronized devices. The synchronization is described for two devices, but it is also applicable for multiple devices, i.e., for a network. Two in-sync devices are in phase and from the point of view of a third device, they appear like one device.

Figure 4 shows a comparison between the synchronization method with constant correction steps and the method with exponentially decaying correction steps. The experiment is performed for 2 to 12 stations and its goal is to measure the time until a stable synchronization has been reached. Two (or more) devices achieve a stable synchronization if S1 and S2 values are equal (within a margin) for at least ten consecutive synchronization algorithm invocations. The two methods perform similarly for fewer than seven devices. The time to synchronize is always below one second. Starting from seven devices and up to twelve devices, the constant step method performs significantly worse than the exponential method. The high standard deviation of the constant step method indicates that the time needed to synchronize is heavily dependent on the starting point and may take fifteen seconds and more. The exponential method always stays below one second, independent of the number of stations and starting point. This result also correlates with previous work where the performance starts to decrease for six devices and more (cf. Figure 13 in [2]).

Figure 5 shows the synchronization stability for both the constant correction step method and the exponential method. The stability is measured as follows: The Y-axis of the plot shows the absolute difference between the two synchronization interval measurements. As mentioned before, if they detect the same amount of light, the devices are synchronized, if one is larger than the other, light is leaking from the previous or next ON slot. The lower this difference is the better the synchronization. More important than the absolute average value are the error bars showing the standard deviation. This value is an indicator of how stable the synchronization is. The plot shows that even while using occasionally larger correction steps, the exponential method provides a more stable synchronization than the constant method. It also shows that the instability increases with the number of devices in the network for the exponential method whereas for the constant method the instability stays the same.

D. Data encoding and decoding scheme

Figure 6 illustrates how data is transmitted and received. The transmitting device enables light output during the D1 or D2 intervals of its OFF slots. A synchronized receiving device measures its two data intervals and compares the two values. If D1 is significantly larger than D2, meaning that there was light present during D1, the received symbol is decoded to a 0. If D2 is larger than D1 the received symbol results to a 1. Because there is always light emission involved in transmitting 0 or 1, it is straightforward to derive a medium busy or idle scheme. As long as light is detected in one of the data intervals, there is a transmission going on and the medium is busy. If D1 and D2 are very close, only ambient light is present and the medium is therefore idle. This channel sensing is fully independent from the data message being transmitted.

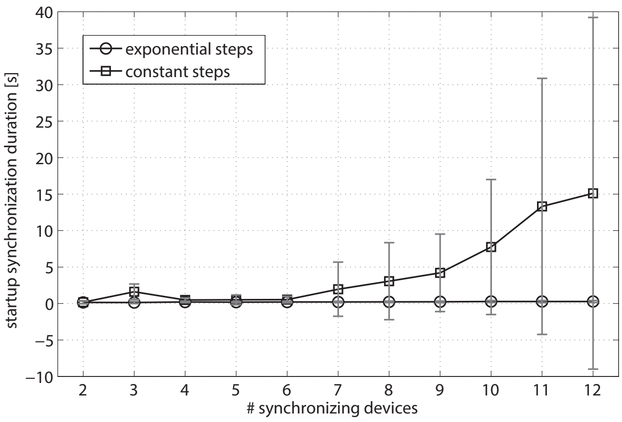


Fig. 4. Startup synchronization duration for up to twelve devices and two synchronization methods: constant and exponential step approximation. Error bars indicate the standard deviation.

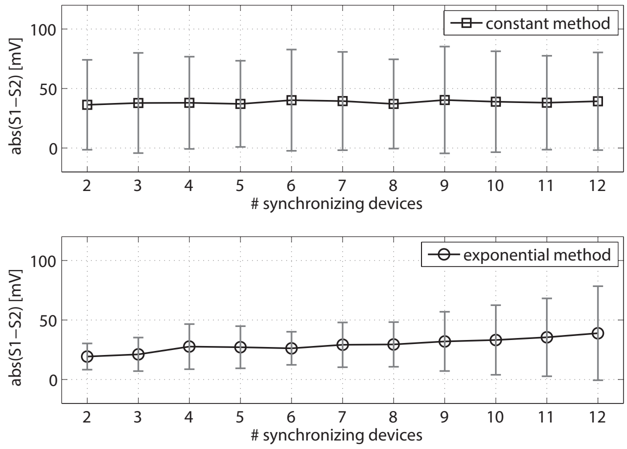


Fig. 5. Network synchronization stability for up to twelve synchronizing devices and the two synchronization methods. Error bars indicate the standard deviation.

Using the data intervals for light output during the transmission of a message increases the amount of light emitted per time for the duration of the transmission. This is perceived by the human observer as an increment of brightness. Figure 7 shows how this effect can be compensated. To reduce the unwanted visible brightness changes (flickering), the light output must remain constant within a short time frame (around one millisecond). For this, the same amount of light emitted during one of the data intervals must be simply removed from the following ON slot. This keeps the light output during one millisecond constant. The fast on-off changes within this millisecond are not perceived by a human eye.

E. PHY header

The PHY header consists of a 1 byte Start Frame Delimiter (SFD) preamble and 3 additional bytes. The SFD is used to announce the beginning of a data transmission and informs the receiver to expect an incoming packet. The first byte of the header is reserved and, e.g., can be used to announce different PHY modes. The second byte carries the length of the following data payload, and the third byte represents a 8-bit Frame Check Sequence (FCS) to protect the reserved and length field. If the FCS cannot be matched, reception is aborted. The structure of a PHY Protocol Data Unit (PDU) is summarized inTable I.

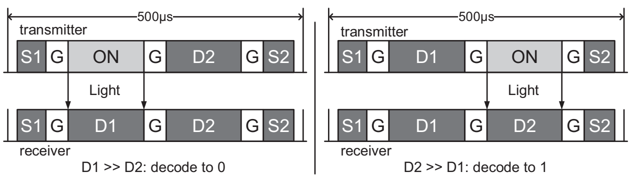


Fig. 6. Data encoding and decoding scheme: The transmitter either enables light emission during its first data interval (of its OFF slots) or during its second data slot.

TABLE I. PHY AND MAC HEADER STRUCTURES.

PHY header (4 byte)		MAC header (6 byte)	
Element	**byte**	**Element**	**byte**
SFD	1	type and flags	1
service (PHY modes)	1	source address	1
payload size	1	destination address	1
PHY FCS	1	sequence number	1
		MAC FCS	2

III. MAC LAYER PROTOCOL

A simplified 802.11-like approach is discussed in this section. The MAC layer uses Acknowledgments (ACK), retransmissions, MAC header, and a packet structure further explained in Table I. Every data packet starts with the PHY and MAC header, followed by the payload, and is terminated with a 16-bit FCS. If a received data frame can be matched to its FCS, an ACK is generated and immediately transmitted. If no ACK is received for a previously transmitted data frame until the ACK timeout interval expires, the frame is retransmitted until a configurable number of retransmissions is reached. To guarantee that an ACK does not collide with a data frame, it is transmitted directly after a successful reception, whereas a data frame is obligated to wait at least for an Inter Frame Space (IFS). To prevent simultaneous data transmissions from multiple devices in a network, a device waits for an IFS duration plus a random time before transmitting. The random time depends on the current Contention Window (CW) size for that device. If an SFD is detected during this waiting phase, or light is detected during the last OFF slot, the CW counter is stopped. As soon as the end of the packet is recognized, the device waits again for IFS and then resumes the CW counter.

Resuming the contention instead of generating a new random contention interval provides basic fairness since no device waits forever, and each device is eventually able to transmit. If the end of the CW is reached but a busy medium is detected, the device also backoffs and can transmit during the next free medium slice after waiting for the initial IFS. Hence, we can detect a busy medium in two different ways.

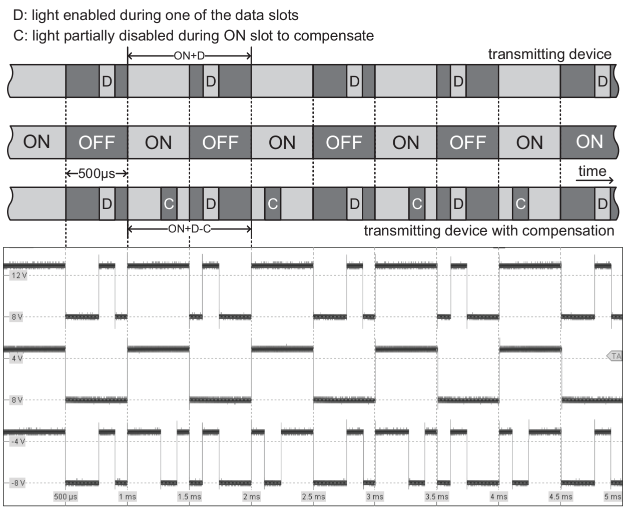


Fig. 7. Flicker compensation during data transmission. The middle sequence shows alternating ON and OFF slots to make the other two sequences easier to read. ON+D indicates the duration when light is emitted without compensation. ON+D-C indicates the behavior for compensation.

First, incoming SFDs set the device in receive mode, which prohibits the transmission of messages. Second, the channel is sensed during every OFF slot with the method presented in the last section. On unsuccessful transmission, the CW is doubled starting from the CW minimum (CW_{min}) up to the CW maximum (CW_{max}). This step reduces the probability that the CW of two or more devices are equal. On successful packet delivery, the current CW size is reset to CW_{min}.

Figure 8 illustrates and describes the medium access for a network of four devices. An oscilloscope registers the output from digital pins of the microcontroller. At the starting point of a transmission, the used pin is set to high, and at the end it is set to low again. While a device has a data frame pending to transmit and it is waiting for the duration of its contention window, the same pin is set to high and immediately low again. The oscilloscope displays this activity as a dark gray area.

IV. NETWORK EVALUATION

Every testbed device consists of an ATmega328P microprocessor, an LED, and a USB interface to provide power and to enable configuration and logging. The LED type used for all measurements is Kingbright L-7113SEC-J3. It supports a peak wavelength of 640 nm, has a 20 degree beam angle and a brightness of 12000 mcd. The form factor is 5 mm with a transparent housing. The devices are configured using the serial-to-USB interface and measurement data is also logged through the same interface. Each device runs the discussed PHY and MAC layer on its microcontroller with the testbed application on top. The application is responsible to apply the received configuration and to generate the desired network traffic. All experiments are conducted inside a standard office space (with windows). The testbed is not shielded from ambient light and was influenced by the office lighting and sunlight; no special provisions were made to conduct experiments on specific days or at specific hours.

The experiment aims at evaluating a network of up to 12 devices deployed in a circle as shown in Figure 9. Every LED can detect incoming light from all other LEDs. To evaluate if networking is possible and if the proposed MAC protocol works correctly, 1 to 11 devices generate traffic, try to access the medium, and transmit data to one sink (always the same device) at the same time. The measurements are run for 1 to 11 transmitters, different packet sizes and saturation data traffic. This star configuration is allows for a large number of nodes; in many practical settings each LED may be in range of only a few other LEDs. Only the MAC payload is counted as data, and it is accounted for in the throughput results only if an ACK

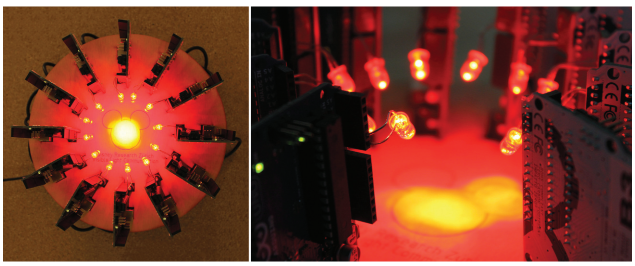


Fig. 9. Testbed used for the protocol evaluation. The figure shows the testbed for a single cell: All devices are in range of each other.

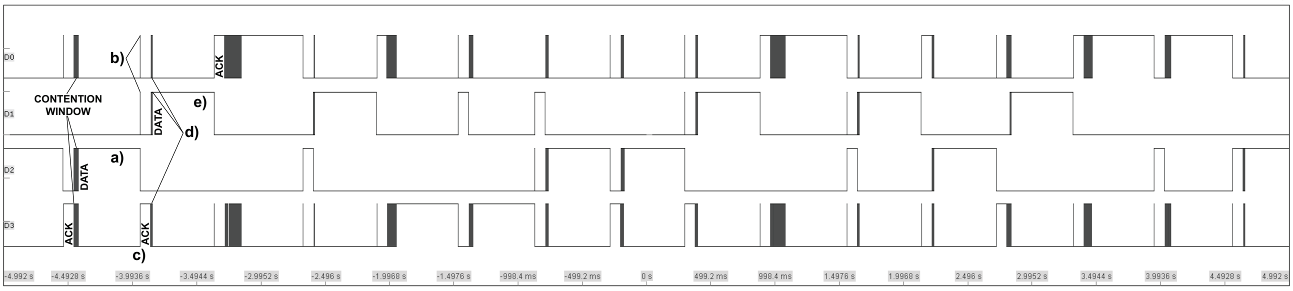


Fig. 8. Network traffic visualized with an oscilloscope. Data and ACK frames together with contention windows are shown. a) D2 finished sending its data frame; b) D0 and D1 start counting down their contention window (IFS); c) D3's ACK has priority over data frames and is immediately sent; d) D0, D1, D3 compete for the medium; e) D1 wins the race for the medium and transmits its packet.

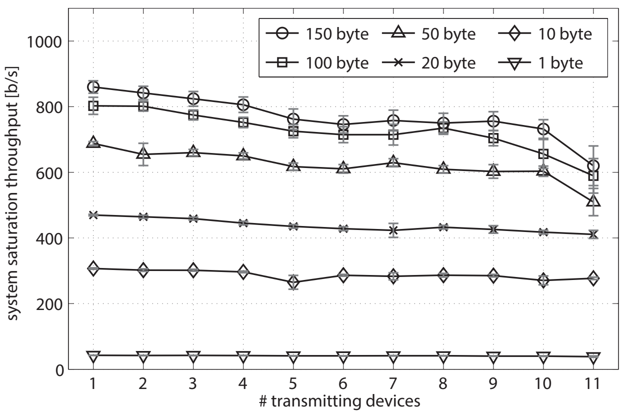


Fig. 10. System saturation throughput for different packet sizes and varying number of devices.

for a packet is received. Figure 10 shows the data throughput measurements. For small packet sizes ($\leq$ 20 byte), the PHY and MAC layer overhead and also the time spent in the contention window are significant and reduce the throughput. On the other hand, small packets are less affected by bit errors, and lost packets can be retransmitted without losing a lot of valuable medium time. Therefore the throughput achieved by the 1 byte, 10 byte, and 20 byte MAC payload stays almost constant for an increasing number of transmitting stations. For larger packet sizes lost data carries weight, and since the collision and error probability (it takes more than one second to transmit a 150 byte packet) increase with the number of stations, the throughput drops within reasonable amounts. The experiments show that with the novel synchronization the network can easily be scaled to 12 devices; larger networks are possible but have not been evaluated.

V. Conclusions

Efficient synchronization is a key aspect of LED-to-LED VLC networks. We present a technique that is practical and has been integrated into the PHY layer of an experimental LED-based optical network. This PHY layer improves performance over earlier work and provides a stable foundation for the MAC layer that handles shared medium access to enable network scalability. The network throughput of an LED-to-LED network as presented here can reach up to 800 b/s and scales well with the number of communicating devices without noticeable detriment of throughput for larger networks. The PHY layer modulation and coding scheme and the MAC protocol are completely implemented in software. This design decision provides numerous benefits: First, the resulting system is flexible, allowing us to explore various directions. Furthermore, a software-based solution allows a fast turn-around. And in addition, the complete system is fairly simple and uses only off-the-shelf components. LED-based VLC systems are attractive for a number of reasons. The protocol described here allows us to build realistic reasonable-sized networks with interesting properties: the combination of mobile devices that contain LEDs with LED light bulbs creates a new class of all-optical networks that we expect to be popular in the future. The synchronization technique presented in this paper is key to stable communication in such networks.

References

[1] P. Dietz, W. Yerazunis, and D. Leigh, "Very Low-Cost Sensing and Communication Using Bidirectional LEDs," in *UbiComp 2003: Ubiquitous Computing*, vol. 2864, pp. 175–191, 2003.

[2] S. Schmid, G. Corbellini, S. Mangold, and T. Gross, "LED-to-LED Visible Light Communication Networks," in *MobiHoc, 2013 ACM*, Aug. 2013.

[3] D. O'Brien, "Visible Light Communications: Challenges and Potential," in *IEEE Photonics Conference*, pp. 365–366, Oct. 2011.

[4] 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.

[5] S. Mangold, "Visible Light Communications for Entertainment Networking," in *Photonics Society Summer Topical Meeting, 2012 IEEE*, pp. 100 –101, July 2012.

[6] 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.

[7] G. Corbellini, S. Schmid, S. Mangold, T. R. Gross, and A. Mkrtchyan, "LED-to-LED Visible Light Communication for Mobile Applications," in *Demo at ACM SIGGRAPH Mobile 2012*, Aug. 2012.

[8] S. Schmid, J. Ziegler, T. R. Gross, M. Hitz, A. Psarra, G. Corbellini, and S. Mangold, "(In)visible light communication: Combining illumination and communication," in *ACM SIGGRAPH 2014 Emerging Technologies*, SIGGRAPH '14, 2014.

[9] S. Schmid, G. Corbellini, S. Mangold, and T. Gross, "An LED-to-LED Visible Light Communication System with Software-based Synchronization," in *Optical Wireless Communication. Globecom Workshops (GC Wkshps), 2012 IEEE*, pp. 1264–1268, Dec. 2012.