



HAL
open science

Analytical and Simulation Tools for Optical Camera Communications

Alexis Duque, Razvan Stanica, Hervé Rivano, Adrien Desportes

► **To cite this version:**

Alexis Duque, Razvan Stanica, Hervé Rivano, Adrien Desportes. Analytical and Simulation Tools for Optical Camera Communications. *Computer Communications*, 2020, 160, pp.52-62. 10.1016/j.comcom.2020.05.036 . hal-02909386

HAL Id: hal-02909386

<https://inria.hal.science/hal-02909386>

Submitted on 30 Jul 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Analytical and Simulation Tools for Optical Camera Communications

Alexis Duque^{a,1}, Razvan Stanica^a, Herve Rivano^a, Adrien Desportes^b

^a*Univ Lyon, INSA Lyon, Inria, CITI, F-69621 Villeurbanne, France.*

^b*Rtone, Lyon, France.*

Abstract

The use of LED-to-camera communication opens the door to a wide range of use cases and applications, with diverse requirements in terms of quality of service. However, while analytical models and simulation tools exist for all the major radio communication technologies, the only way of currently evaluating the performance of a network mechanism over LED-to-camera is to implement and test it. Our work aims to fill this gap by proposing a Markov-modulated Bernoulli process to model the wireless channel in LED-to-camera communications, which is shown to closely match experimental results. Based on this model, we develop and validate *CamComSim*, the first network simulator for LED-to-camera communications.

Keywords: visible light communications, LED-to-camera, simulation tools, Markov-modulated Bernoulli process

1. Introduction

2 Visible-light communication (VLC) is an enabling technology that ex-
3 ploits illumination to provide a short-range wireless communication link.
4 VLC systems take advantage of the license-free light spectrum and their
5 immunity to radio frequency (RF) interference. In such systems, informa-
6 tion is often relayed by modulating the output intensity of a light-emitting
7 diode (LED). Any electronic device which can detect the presence or absence
8 of visible light can be utilized as a VLC receiver. While most of the work
9 in the field is focused towards using photo-diodes as receivers [1, 2], because
10 of their fast response and high bandwidth, some studies demonstrated that
11 smartphone cameras can also be used to detect high-frequency light patterns
12 [3].

13 Indeed, nowadays smartphone cameras widely use two types of image
 14 sensors, Charge Coupled Devices (CCD) or Complementary Metal Oxide
 15 Semiconductors (CMOS). These two technologies have some similarities, but
 16 one major distinction is the way each sensor exposes its pixels to light. CCD
 17 sensors use the Global Shutter readout mode, where all pixels are exposed
 18 simultaneously and then each pixel is read sequentially. This mechanism
 19 helps in capturing a still image of a moving object. On the other hand, CMOS
 20 sensors use the Rolling Shutter readout mode [4], where each row is exposed
 21 in a row-sequential way with fixed time delay. Due to this mechanism, there
 22 is a significant time difference between the beginning of the exposure of the
 23 first and the last row, making them no longer simultaneous. When an LED
 24 is modulated at a frequency higher than the rolling shutter speed, stripes
 25 of different light intensity are captured in the image. A row of pixels appears
 26 illuminated when the LED was ON during the row exposure time. On the
 27 other hand, a row appears dark when the LED was OFF during the exposure
 28 time, as shown in Fig. 1. The intensity and width of the strip depend on
 29 the transmitter modulation frequency, allowing us to encode information in
 30 these illuminated and dark bands, similarly to the use of a bar code.

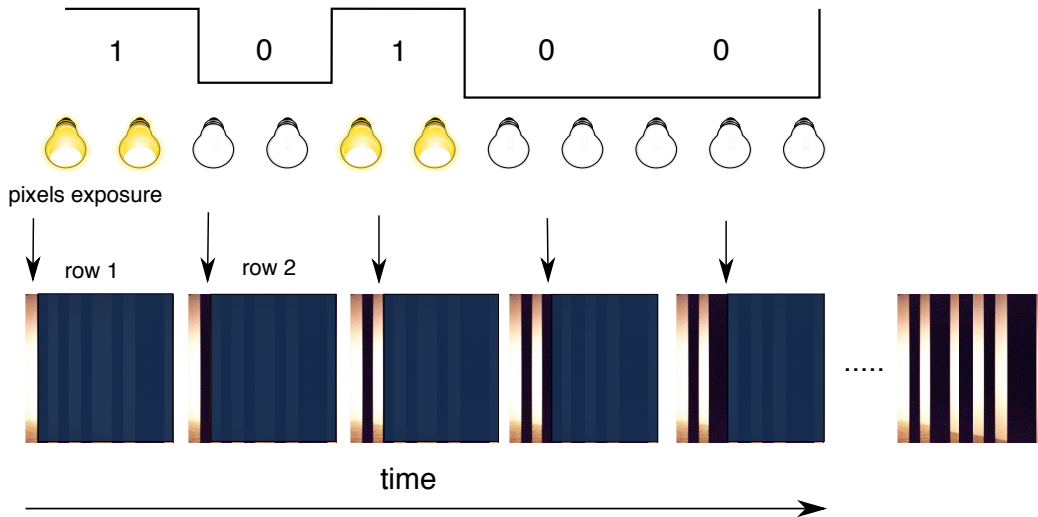


Figure 1: Stripe formation and LED state. The width of each strip corresponds to the duration of the LED state (ON or OFF).

31 This LED-to-camera communication based on the Rolling Shutter effect

32 opens the door to a wide range of use cases and applications, with diverse
33 requirements in terms of quality of service [5]. To cite a few examples, both
34 line-of-sight (LOS) [6, 7] and non-line-of-sight (NLOS) [8, 9] communications
35 have been demonstrated in these settings, as well as ultra-reliable localiza-
36 tion solutions [10, 11], sensing [12], or even scene protection against intrusive
37 photographs [13]. However, while analytical models and simulation tools exist
38 for all the major RF technologies, the only way of currently evaluating
39 the performance of a network mechanism over LED-to-camera is to imple-
40 ment and test it. This results in heavy measurement and parameterisation
41 campaigns that need to be repeated anytime a new VLC protocol or feature
42 is imagined. Having access to standard performance evaluation tools in this
43 type of network would certainly accelerate studies in the field, and nicely
44 complement experimental field tests.

45 The work described in this paper aims to fill this gap by proposing models
46 and tools that help in the assessment of LED-to-camera communication net-
47 work mechanisms. Our contributions are threefold. First, we propose an an-
48 alytical model for LED-to-camera communication systems, based on the the-
49 ory of Markov-modulated Bernoulli processes (MMBP) [14], and show that it
50 is significantly more accurate than a classical Gilbert-Elliott model [15]. Sec-
51 ond, to facilitate performance evaluation tasks, we design, implement and
52 validate *CamComSim*, the first LED-to-camera communication simulator.
53 Finally, the model and the simulator allow us to benchmark several network
54 redundancy mechanisms proposed in the literature. By checking against ex-
55 perimental results, we are able to confirm the correctness of our models in
56 this context.

57 The remainder of this study is structured as follows. We discuss related
58 works in Sec. 2 and we describe the testbed used for experiments in Sec. 3.
59 We detail our analytical modeling of the LED-to-camera propagation channel
60 in Sec. 4, complemented by a model of the receiver in Sec. 5. The *CamCom-*
61 *Sim* simulator is presented, validated and tested in Sec. 6, before concluding
62 remarks in Sec. 7.

63 2. Related Works

64 The use of visible light for communications actually predates the one of
65 RF. In fact, the first phone call in history was actually made using a photo-
66 phone [16] and optical communications are a major component of telecom-
67 munication systems nowadays, through their use in optical fibers [17].

68 The use of wireless VLC gained a lot of interest in the last two decades,
69 with the development of wireless local area networks (WLAN), as a competi-
70 tor to the WiFi technology [18]. The term *LiFi* was coined in this sense [19]
71 and numerous works discussed the advantages and disadvantages of this ap-
72 proach [5]. However, the use of VLC in WLANs requires dedicated receivers,
73 usually incorporating sensitive photodiodes [1, 2]. In this sense, CoLight
74 [20] interfaces a photodiode with a smartphone, using the audio jack and
75 obtaining a throughput of 80 kbits/s.

76 In 2012, Danakis *et al.* [3] demonstrated for the first time that the rolling
77 shutter effect of smartphone cameras can be used to receive information
78 transmitted through visible light, creating the field of optical camera com-
79 munications [21]. Using the smartphone camera as a receiver can also be used
80 for other services. For example, *LiTell* [10] is a localization scheme that em-
81 ploys unmodified fluorescent lights (FLs) as location landmarks and relies on
82 the observation that each FL has an inherent characteristic frequency which
83 can serve as a discriminant feature, providing sub-meter accuracy. Using a
84 similar idea, *iLAMP* [11] further introduces a sensor-assisted photogramme-
85 try technique to estimate the 3D location with a small 90-percentile error of
86 3.5 cm. Another example is that of privacy, where *LiShield* [13] leverages
87 the rolling shutter artifact to protect a physical scene from photographing,
88 by illuminating it with smart LEDs transmitting modulated light.

89 However, the main usage of the rolling shutter effect remains LED-to-
90 camera communication, which raises a lot of interest because it enables short-
91 range communication with no extra financial cost between any LED-equipped
92 machine or instrument and any regular smartphone. Both LOS and NLOS
93 communication has been demonstrated, and a real competition in terms of
94 throughput began. Lee *et al.* [6] extensively studied the rolling shutter ef-
95 fect and found an unpredictable and varying jitter between two consecutive
96 frames. Their solution, *RollingLight*, uses a simple coding approach and pro-
97 vides a throughput of 20 bytes/s. Other solutions, such as *CeilingCast* [7]
98 leverage rateless codes to face the unavoidable packet erasure caused both by
99 the camera and the distance, reaching a throughput of 1.35 kbits/s. *MAR-*
100 *TIAN* [8] exploits NLOS reflections and an original modulation technique
101 to achieve a throughput of 1.6 kbits/s. In our own previous work [22], we
102 relied on random linear coding to reach a throughput of 2.2 kbits/s. Relying
103 on colored LEDs, *ColorBars* [23] demonstrates a throughput between 2.9
104 kbits/s and 7.7 kbits/s, depending on the receiving smartphone.

105 However, despite this significant interest in the subject, practically every

106 study on this topic uses an experimental approach. While experiments are
107 essential in evaluating new protocols and services, running an experimental
108 campaign every time one wishes to evaluate a new idea can become cumbersome.
109 It is also very difficult to reproduce other solutions and fairly compare
110 different techniques. Several VLC testbeds have been proposed in this sense
111 [24, 25], but non-experimental performance evaluation tools in this context
112 are poorly studied. A rare example is a channel model recently proposed
113 by Zhang *et al.* [26], where the light attenuation between the LED and the
114 camera is modeled depending on different physical parameters.

115 Designing analytical models and implementing them in simulation tools
116 is standard practice in the wireless networking field in order to accelerate the
117 evaluation of new protocols and mechanisms. Indeed, network simulation has
118 been largely studied in the case of wireless communication [27]. Nonetheless,
119 VLC performance evaluation remains poorly investigated and VLC simulation
120 tools are still missing. The main efforts on simulating VLC systems have
121 focused on indoor channel simulation [28], or on the 802.15.7 PHY [29, 30]
122 and MAC [31] layers. These approaches rely on classical network simulation
123 frameworks, such as those used for wireless and ad hoc networks, e.g. ns-2
124 [31], ns-3 [29], OMNET++ [30] or MATLAB [28]. All these works consider
125 LED-to-Photodiode communication, hence OCC is completely unexplored.
126 Our work is the first effort in LED-to-Camera simulation reported in the literature,
127 making *CamComSim* the first implementation of a LED-to-Camera
128 VLC simulator.

129 3. Testbed description

130 The analytical and simulation models proposed in this paper are compared
131 and validated using an extensive experimental campaign. These experiments
132 were conducted using a testbed detailed in our previous work [32].
133 We briefly describe its main components below.

134 We designed a 40x20 mm printed circuit board (PCB) shown in Fig. 2
135 that supports a red 5 mm LED, a micro-controller unit (MCU), a surface
136 mount device (SMD) RGB LED whose color can be easily changed, and a
137 temperature sensor, to broadcast the ambient temperature through the light.
138 To evaluate the system, we also use a Nucleo development board to interface
139 the MCU with more convenience, and so, change without hardware modification
140 the LED type, the General Purpose Input/Output (GPIO) mapping or
141 the firmware implementation. The development board is depicted in Fig. 3.

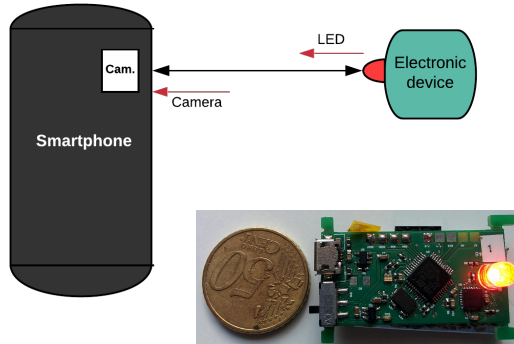


Figure 2: On the left, our LED-to-Camera VLC system. On the right, our smart VLC device prototype.

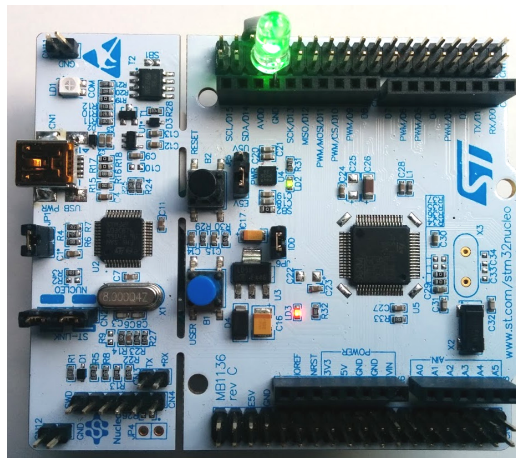


Figure 3: The Nucleo STM32L0 development board.

142 We choose the STM32L051 low cost and low power MCU from ST Micro-
 143 electronics, an MCU similar to those already integrated in most household
 144 appliances. The core is a Cortex M0+, running up to 32 MHz, with 32 Ko
 145 Flash and 8 Ko RAM. To get a better clock accuracy and avoid clock bias
 146 due to the temperature, we use an 8 MHz high speed external crystal oscilla-
 147 tor as the clock source and make the core run at this speed. As proposed in
 148 previous studies [3, 6], the LED signal is modulated using the on-off keying
 149 modulation scheme. We consider a clock-rate varying from 2 to 10 KHz,
 150 which is a suitable bandwidth for OCC when using the rolling shutter effect

151 [9]. To ensure a balanced duty cycle signal and avoid any flickering effect,
152 we use the Manchester coding proposed in [3, 9].

153 On the receiver side, we use a LG Nexus 5 smartphone running Android
154 Marshmallow version number 6.0.1. It has a Qualcomm Snapdragon 800
155 quad-core CPU 2,26 GHz CPU and 2 Go RAM. Its 8 megapixels 1080p
156 1/3.2" CMOS sensor with 1.4 μm pixel size can capture up to to 30 frames per
157 second and supports advanced imaging application provided by the Camera2
158 API. We have developed an Android application that sets up the camera
159 parameters to observe the rolling shutter effect produced by the modulated
160 LED. For that, based on [9], we set a very short exposure time and an
161 increased sensor sensitivity, respectively to 100 μs and ISO 10000. As soon
162 as a new frame is available, the application creates and starts a new thread
163 to process and decode the picture on the background.

164 4. Modeling LED-to-camera communication

165 In the follwoing, we describe an OCC communication channel model.
166 Based on the theory of Markov-modulated Bernoulli processes (MMBP) [14],
167 discussed in Sec. 4.1, this model can be applied to all LED-to-camera com-
168 munication systems. The proposed model is not only generic, but also very
169 accurate, as demonstrated by its validation with extensive experimental re-
170 sults in Sec. 4.4. As an example, we use the analytical model to compare two
171 simple redundancy mechanisms, required to cope with the inherent losses of
172 LED-to-camera communication and described in Sec. 4.2 and Sec. 4.3.

173 4.1. Model design

174 In a LED-to-camera system, data is received as a series of dark and
175 illuminated stripes appearing in a region of a picture frame captured by the
176 camera. We denote this part of the picture as a region of interest (ROI). In
177 the following, we note by f_i the i -th frame captured by the camera and by δ_f
178 the time between the beginning of two consecutive frames. Obviously, even
179 at the highest frame rate allowed by the camera, data is not continuously
180 received, as a minimum time δ_g exists between two frames. This is denoted
181 as the inter-frame gap (IFG). Moreover, as depicted in Fig. 4, the distance
182 between the LED and the camera also has an impact: when the camera is
183 farther away, the LED transmission is captured for a shorter time, resulting
184 in a smaller ROI.

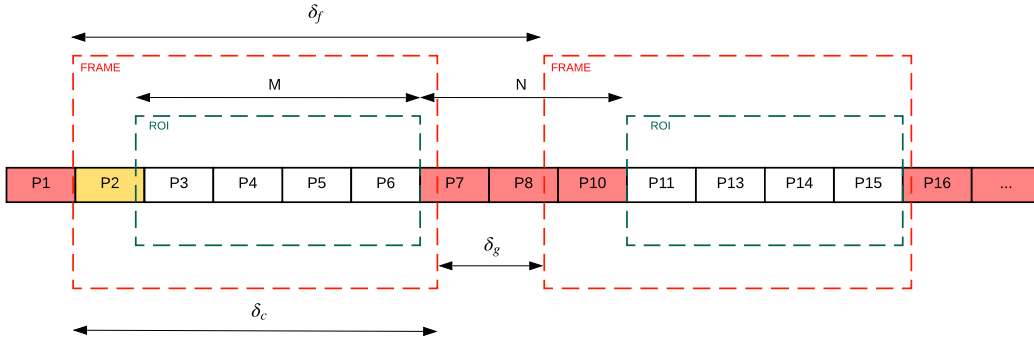


Figure 4: Frame capture time and inter-frame interval, and their relation with the MMBP parameters.

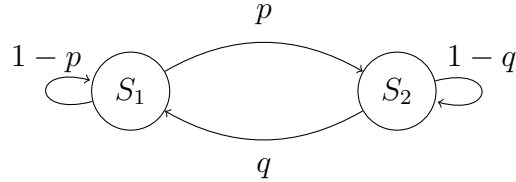


Figure 5: The *Gilbert-Elliott* model of the LED-to-camera channel.

185 *4.1.1. Gilbert-Elliott model:*

186 A first idea to model LED-to-camera communication would be the Gilbert-
 187 Elliot model, which is widely used to model bursty losses [15]. This unique
 188 type of channel can intuitively be modeled by a two states Markov chain,
 189 as depicted in Fig. 5. In state S_1 , the system is capturing a frame. The
 190 camera is receiving packets, and the reception probability is $1 - p_e$, where
 191 p_e is the packet decoding error probability. In state S_2 , the camera is not
 192 capturing any pictures, therefore we consider the reception probability is 0.
 193 The transition probability between S_1 and S_2 (respectively S_2 and S_1)
 194 is denoted p (respectively q). The model assumes that p, q, p_e are independent
 195 and constant.

196 In this case, the probability of being in state S_1 under the steady-state
 197 regime can be easily computed as $p_{s_1} = \frac{p}{p+q}$. The probability of being in
 198 state S_2 is so $p_{s_2} = \frac{q}{p+q}$.

199 The values of p and q are function of the duration of the IFG, δ_g , the
 200 frame duration, δ_f , and the camera capture time δ_c , depicted in Fig. 4, and
 201 linked as the following:

$$p = 1 - q = \frac{\delta_f - \delta_g}{\delta_f} = \frac{\delta_c}{\delta_f} \quad (1)$$

202 *4.1.2. MMBP model:*

203 If the model introduced just before is straightforward and widely used,
 204 it lacks realism in our case, where the transitions between ON and OFF
 205 states are almost deterministic. Practically, in our system, the transition
 206 probability from a state to another depends on the residence time in this
 207 state.

208 To improve this approach, we model the LED-to-camera channel using
 209 a Markov-modulated Bernoulli process (MMBP), represented in Fig. 6. In
 210 this figure, we depict a Markov chain with a total number of $M + N$ states.
 211 Each of these states represents a reception time slot, i.e. the time duration
 212 needed in order to receive one physical layer message (denoted as PHY-SDU
 213 in the following). The transition between two states representing successive
 214 time slots is automatic, i.e. it happens with a probability of 1.

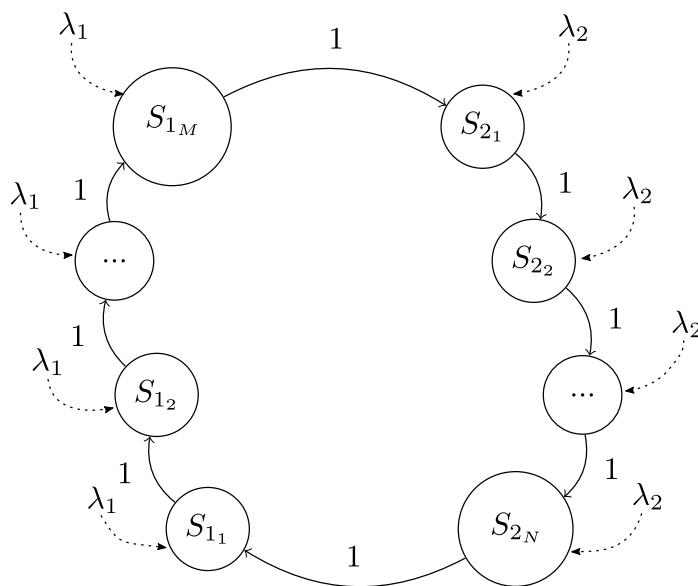


Figure 6: The MMBP model of the LED-to-camera channel.

215 Practically, the $M + N$ states in Fig. 6 represent a δ_f time interval,
 216 and they are divided in two groups: M states corresponding to the camera

217 capture time δ_c (S_{ON} states), and N states corresponding to the inter-frame
 218 time δ_g (S_{OFF} states). A Bernoulli arrival process is associated with each of
 219 these $M + N$ states, representing the reception of a packet.

220 In S_{ON} states, the camera is receiving packets, and the arrival rate is
 221 $\lambda_1 = (1 - p_e)$, where p_e is the packet decoding error probability. In S_{OFF}
 222 states, the camera is not capturing any pictures, therefore we consider the
 223 arrival rate $\lambda_2 = 0$.

We denote as s a state in the Markov chain and we define state $s + j$ as
 the state reached after j transitions, starting from state s . The probability
 of being in state s under the steady-state regime can be easily computed as
 $\pi_s = \frac{1}{N+M}$. At the same time, the probability of noticing no arrivals (i.e. no
 packet reception) in state s is $p_0(s)$. This can be written as:

$$p_0(s) = \begin{cases} 1, & \text{if } s \in N \\ p_e, & \text{if } s \in M \end{cases} \quad (2)$$

224 As it can be seen from both models, the relatively high packet loss prob-
 225 ability (compared with RF technologies) is an intrinsic property of the LED-
 226 to-camera communication channel. To overcome this problem, redundancy
 227 mechanisms are needed.

228 In the following, using classical redundancy mechanisms as an example,
 229 we show that the classical Gilbert-Elliot model is inaccurate, which highlights
 230 the need to rely on the MMBP theory. Then, we use the MMBP channel
 231 model to compare two simple, but widely used redundancy solutions: repeat-
 232 ing a packet or repeating a sequence of packets.

233 4.2. Repeat Packet

234 The first strategy to cope with the inherent losses in the LED-to-camera
 235 communication system, used for example by Ferrandiz-Lahuerta *et al.* [9], is
 236 to send each packet twice in a row, to increase the probability that at least
 237 one of the transmissions will be fully captured by the smartphone camera.
 238 We generalize this approach in the Repeat Packet (RP) strategy, where each
 239 packet is repeated r times, one after the other. In this case, the r value needs
 240 to be chosen in order to attain a desired reception probability, its optimal
 241 value depending on the inter-frame time and on the packet size.

242 In the following, we study the probability of receiving a packet at least
 243 once when considering the RP strategy, for the two models introduced above.

244 4.2.1. *Gilbert-Elliott Model:*

If we consider the Gilbert-Elliott model, the probability of receiving a packet at least once can be written as $p_s^{RP} = 1 - p_0^{RP}$, where p_0^{RP} represents the probability of failing to receive a packet r times in a row, written as:

$$p_0^{RP} = (P_{S_1} \cdot p_e + P_{S_2})^r = \left(\frac{p \cdot p_e + q}{q + p} \right)^r \quad (3)$$

245 4.2.2. *MMBP Model:*

If we consider the MMBP model, the probability of failing to receive a packet r times in a row p_0^{RP} can be written as:

$$p_0^{RP} = \frac{\sum_s (p_0(s) \cdot p_0(s+1) \cdot p_0(s+2) \cdot \dots \cdot p_0(s+r-1))}{N+M} \quad (4)$$

The value of p_0^{RP} will depend on m_s , defined as the number of S_{ON} states during the r retransmissions, when the first packet transmission is in state s :

$$p_0^{RP} = \frac{1}{N+M} \cdot \sum_{s=1}^{M+N} (p_e)^{m_s} \quad (5)$$

246 Depending on the values of r , M and N , several cases can be distin-
247 guished. We present results for the two most current cases:

Case 1: $r < M, N$. This means that the number of retransmissions does not always cover the δ_g period (which counts N states). In this case, p_0^{RP} can be written as follows:

$$\begin{aligned} p_0^{RP} &= P[s \in N \wedge (s+r) \in N] + P[s \in M \wedge (s+r) \in M] \\ &\quad + P[s \in M \wedge (s+r) \in N] + P[s \in N \wedge (s+r) \in M] \\ &= \frac{N-r+1}{M+N} + \frac{M-r+1}{M+N} \cdot p_e^r + \frac{1}{M+N} \sum_{i=1}^{r-1} p_e^i + \frac{1}{M+N} \sum_{i=1}^{r-1} p_e^{r-i} \\ &= \frac{N-r+1}{M+N} + \frac{M-r+1}{M+N} \cdot p_e^r + \frac{2}{M+N} \cdot \frac{p_e - p_e^r}{1 - p_e} \end{aligned} \quad (6)$$

Case 2: $N < r < M$. . This is the most common case, where the number of repetitions is chosen to cover the entire inter-frame period. However, a

reception is still not certain in this case, because of the decoding error p_e . In this case:

$$\begin{aligned}
p_0^{RP} &= P[s \in M \wedge (s+r) \in M] + P[s \in M \wedge (s+r) \in N] \\
&\quad + P[s \in N \wedge (s+r) \in M] \\
&= \frac{M-r+1}{M+N} \cdot p_e^r + \frac{1}{M+N} \sum_{i=1}^{r-1} p_e^i + \frac{1}{M+N} \sum_{i=1}^N p_e^{r-i} \\
&= \frac{M-r+1}{M+N} \cdot p_e^r + \frac{1}{M+N} \cdot \frac{p_e - p_e^r}{1 - p_e} + \frac{1}{M+N} \sum_{i=1}^N p_e^{r-i} \\
&= \frac{M-r+1}{M+N} \cdot p_e^r + \frac{1}{M+N} \cdot \frac{p_e - p_e^r}{1 - p_e} + \frac{1}{M+N} \cdot \frac{p_e^{r-N}(p_e - p_e^N)}{1 - p_e}
\end{aligned} \tag{7}$$

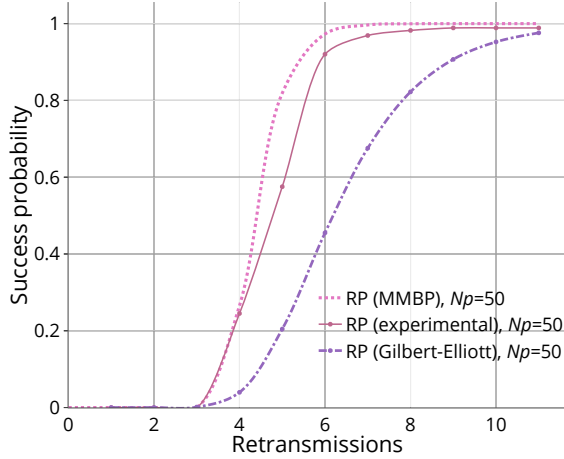


Figure 7: Probability to successfully receive N_p packets for the RP strategy. Dotted-lines show analytical results for the Gilbert-Elliott and MMBP models, while plain lines represent experimental results.

248

249 We compare the analytical results given by the two aforesaid models to
250 experimentation results, obtained using our testbed, in Fig. 7. This figure
251 shows the success probability of receiving a message of $N_p = 50$ packets of
252 data, as a function of the number of retransmissions r .

253 The very different results between the Gilbert-Elliott model and the ex-
254 perimentation confirms that the stochastic transition assumptions of this

255 model are quite far from reality. On the other hand, MMBP approximates
 256 quite well the experimental behavior, highlighting the need for this more
 257 complex, but finer grained model.

258 4.3. Repeat Sequence

259 A different approach to improve reliability is the Repeat Sequence (RS)
 260 strategy, consisting in the transmission of a sequence of N_p packets, repeated
 261 r times. In contrast with the previous mechanism, RS does not try to cover
 262 the inter frame time at the packet level, and it does not ensure that a packet
 263 is received before sending the next one. Instead, the reliability and presumed
 264 efficiency is based on the fact that the probability of losing the same packets
 265 over different transmitted sequences is low.

In the case of an RS strategy with a sequence of N_p packets retransmitted
 r times, the probability of receiving a packet at least once can be written as
 $p_s^{RS} = 1 - p_0^{RS}$. Using the MMBP model, the probability of failing to receive
 a packet r times in a row, p_0^{RS} , can be written as:

$$p_0^{RS} = \frac{1}{M + N} \cdot \sum_{s=1}^{M+N} \prod_{j=0}^{r-1} p_0(s + rN_p) \quad (8)$$

266 4.4. Evaluation results

267 We use our MMBP analytical model to study the RP and RS strategies
 268 by focusing on the probability of delivering the entire quantity of information
 269 in a given number of transmissions. We provide both analytical and
 270 experimental results, allowing us to validate the proposed MMBP model.

271 Fig. 8 shows, for the two mechanisms, the probability of integrally receiving
 272 N_p packets of data as a function of the number of retransmissions
 273 r . In this figure, we set $M = 5$ and $N = 2$; these values are in line with
 274 the packet length, the transmitter frequency and the camera capture interval
 275 experimentally observed for a distance of 5 cm between LED and camera.
 276 The results show quite a nice fit between the analytical and experimentation
 277 results, despite the assumptions required by our MMBP model.

278 To better understand the performance of the two retransmission strategies,
 279 we compare them in Fig. 9. This figure shows that, for the RS strategy,
 280 3 retransmissions are needed to achieve a reception probability higher than
 281 0.9, while this value raises to 6 for the RP strategy. On the right side of the
 282 figure, we show that the performance of the two strategies depends on the
 283 ratio between the number of S_{ON} and S_{OFF} states, $M : N$. When this ratio

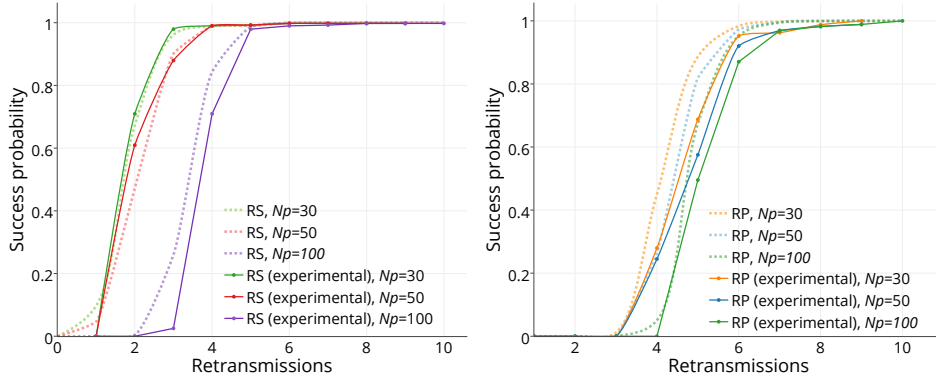


Figure 8: Probability to successfully receive N_p packets for the RS (left) and RP (right) strategies as a function of the number of retransmissions r . Dotted-lines show analytical results while plain lines represent experimental results.

284 changes from 5 : 2 to 2 : 5, which practically corresponds to increasing the
 285 distance between the LED and the camera, RP gives better results than RS.
 286 Indeed, for the RS method, the success probability sharply decrease when
 287 $M < 3$ and stays below 0.6 even for 10 retransmission.

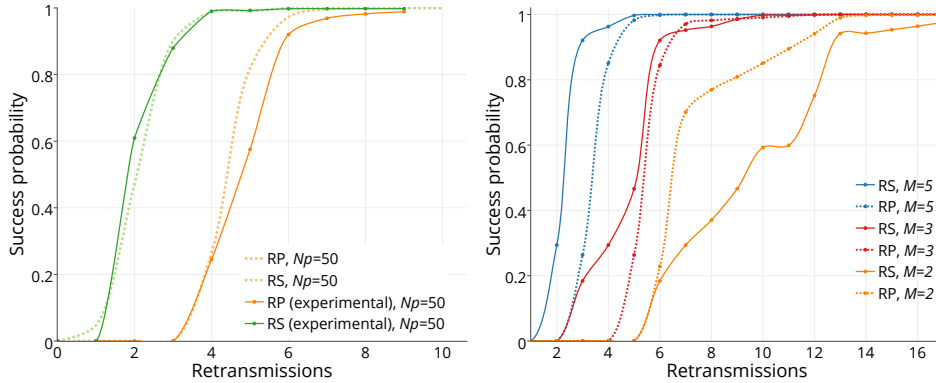


Figure 9: Comparison between RS and RP. On the left, analytical and experimental results for $M = 5$ and $N = 2$. On the right, analytical results when $M + N = 7$, but the $M : N$ ratio changes. In both cases, $N_p = 50$.

288 Practically, this means that RP is more suitable when the distance between
 289 the LED and the camera is higher, while RS is better for short commu-
 290 nication distances. This phenomenon was previously unknown in the research
 291 community, but it is straightforward to study with our analytical model

292 **5. ROI model**

293 An important phenomenon in LED-to-camera communications comes as a
 294 direct consequence of the distance between the LED and the camera. Indeed,
 295 as this distance increases, the size of the ROI in the picture reduces and, as a
 296 consequence, cuts down the number of messages that the camera can receive
 297 per frame, i.e. the M states in Fig. 4. To include this performance factor into
 298 our model, we propose an analytical function that gives the ratio between
 299 the ROI and the picture size. In the model discussed in Sec. 4.1, this is the
 300 ratio of M states in the $M + N$ states.

301 We apply photogrammetry rules to give the ROI ratio as a function of the
 302 distance d , the LED size l , the camera CMOS sensor size ss , the image size
 303 on the sensor i and the camera focal distance fc . According to the optical
 304 system depicted in Fig 10, basic lens optic rules give the following Eq. 9:

$$\frac{i}{fc} = \frac{l}{d} \iff i = \frac{l \cdot fc}{d} \quad (9)$$

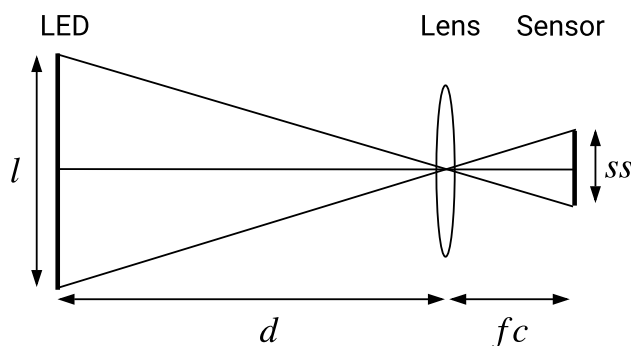


Figure 10: Formation of an image on a sensor by a converging lens.

To obtain the ROI as the ratio of the total number of pixels in the picture, we need as input the CMOS sensor size ss . We apply the $\min()$ function to normalise the $ROI \in [0, 1]$ even if the image size on the sensor, i , is larger than the sensor size, ss . The result is given in the following Eq. 10:

$$ROI = \min \left(1, \frac{l \cdot fc}{d \cdot ss} \right) \quad (10)$$

305 To validate the results given by Eq. 10, we measure the ROI experimen-
 306 tally, using the testbed described in Sec. 3, for distances from 0 to 40 cm.

307 Fig. 11 plots in orange the ROI ratio we observed during our experiments
 308 and in green the analytical results computed with a Nexus 5 sensor with the
 309 following characteristics: $fc = 35$, $ss = 5.7$ and $l = 10$. This shows that
 310 the analytical curve approximates quite well the experimental ROI ratio.
 311 However, we notice that the experimental results are better for a distance
 312 between 10 and 30 cm, and they become worse than the model at 35 cm. In
 313 fact, the light radiance on the camera lens, that our model does not take into
 314 account, artificially increases the LED size on the picture when the camera
 315 is close to the LED. The difference at larger distance is a consequence of the
 316 ambient light which was measured at 650 lux during the experiments, also
 317 neglected in Eq. (10).

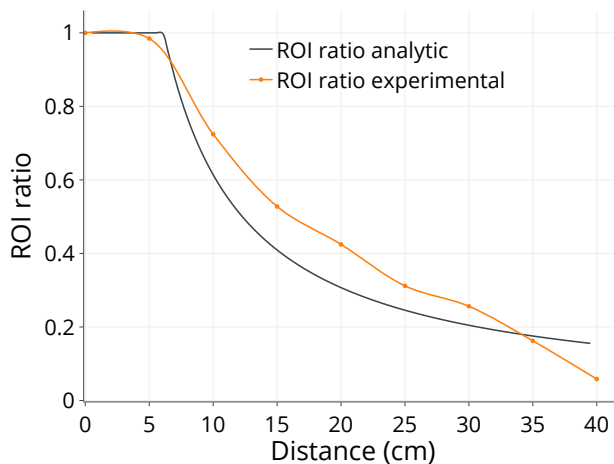


Figure 11: ROI as a function of distance. The orange line shows experimental results, while the green line represents analytical results given by Eq. 10.

318 The ROI model described in this section and the MMBP reception model
 319 validated in the previous section are the basis of the simulator implementa-
 320 tion discussed in the following.

321 6. The CamComSim Simulator

322 As discussed in Sec. 2, the simulation of LED-to-camera communication
 323 remains completely unexplored in the field. Our work is the first such effort
 324 reported in the literature, making *CamComSim* the first implementation of
 325 a LED-to-camera VLC simulator.

326 *6.1. Motivation*

327 The channel and receiver models described in the previous sections allow
 328 us to obtain very accurate results, closely matching the experimental values.
 329 However, the two models are deterministic by nature, and they can lead to
 330 strange artifacts in some specific cases.

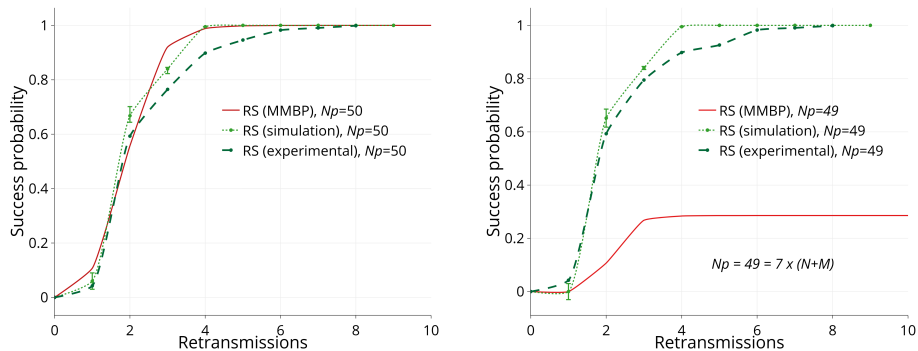


Figure 12: Analytical, simulation and experimental results obtained for $N_p = 50$ (left) and $N_p = 49$ (right).

331 One such example is provided in Fig. 12, where analytical and experimen-
 332 tal results are compared for two different numbers of transmitted packets,
 333 $N_p = 49$ and $N_p = 50$. While the analytical model is very accurate for
 334 $N_p = 50$, things are very different for $N_p = 49$, where the MMBP model
 335 indicates that less than 30% of the packets should be received, regardless the
 336 number of retransmissions. This is an artifact of the MMBP model, which
 337 considers the time between two consecutive pictures taken by the camera to
 338 be perfectly constant. Because of this, when the number of packets N_p is
 339 a multiple of the number of states $M + N$ in the MMBP model, a repeti-
 340 tive pattern appears and all the retransmissions of some packets should be
 341 lost, resulting in a maximum packet reception probability of $N/(M + N)$.
 342 However, the experimental results indicate that this property is not actually
 343 present in real systems, where the inter-frame gap sufficiently variable to re-
 344 move this phenomenon. Accounting for this analytically would be possible,
 345 but it would require an even more complex modeling approach.

346 While built on the models described above, a simulation approach can
 347 address this problem in a much simpler way. Furthermore, a simulator can
 348 easily account not only for inter-frame variability, but also for more com-
 349 plex events, such as user motion. This is demonstrated in Fig. 12, where

350 the results obtained with *CamComSim* closely match those obtained on the
 351 testbed.

352 6.2. Simulator Implementation

353 6.2.1. Software architecture

354 *CamComSim* is an event-driven LED-to-camera simulator developed in
 355 Java, which makes it easy to maintain and distribute code, and it provides
 356 built-in multi-platform compatibility for systems with a Java Virtual Ma-
 357 chine. Fig. 13 shows the *CamComSim* software architecture that consists
 358 of a simulator kernel class and four core packages. For interested readers,
 359 *CamComSim* is already available as an open-source software under Apache
 360 license at <http://vlc.project.citi-lab.fr/camcomsim>.

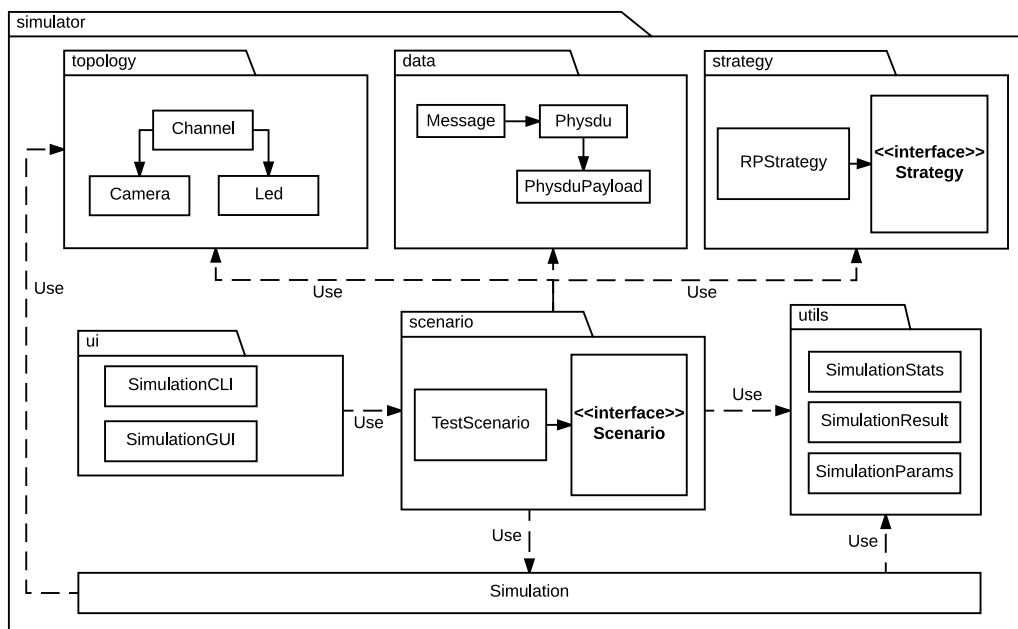


Figure 13: The *CamComSim* software architecture and packages dependency graph.

361 The `topology` package groups classes that describe the system com-
 362 ponents: `Led`, `Camera` and `Channel`. The classes in the `data` package
 363 implement the data encapsulation. For this, a `Message` is a set of PHY-
 364 SDU that encapsulates a `PhysduPayload`. A `Packet` is a `PhysduPayload`
 365 child class, with a sequence number as header and a payload that contains

366 data. Before each simulation, a `Message` is created according to the user
367 settings. The resulting set of `Physdu` is initialized with a `Packet` filled
368 with arbitrary data in the payload (real data could be used if available) and
369 a unique sequence number in the header.

370 The broadcast strategy abstraction is given in the `strategy` package.
371 Here, the `Strategy` interface lets the users implement their transmission
372 strategy. This package also contains the straightforward implementation of
373 the RP strategy, `ReapeatPhysduStrategy`, which consists in repeating each
374 PHY-SDU r times, one after the other, as described in Sec. 4.2. When the
375 last PHY-SDU of the message is reached, the process is repeated from the
376 beginning.

377 Finally, the `scenario` package proposes an interface to build a simula-
378 tion, wiring together the `Message`, the `Channel`, the `Led`, the `Camera` and
379 the `Strategy` with the `Simulator` kernel. Besides, the package `utility`
380 provides helper classes used to compute the simulation results statistics, for-
381 mat and save the results as a JSON file and load or save the simulation
382 parameters. The `ui` package contains a command-line interface (CLI) used
383 to run a simulation scenario.

384 6.2.2. Simulator parameters

385 Our simulator exposes a set of finely grained parameters to describe the
386 LED-to-camera communication system behavior. Table 1 shows the param-
387 eters we use in this work and expose through the *CamComSim* CLI. The
388 performance of the LED-to-camera communication is significantly affected
389 by the distance d , the IFG noted δ_g in Fig. 6, and the LED size l . The values
390 of these parameters should therefore be carefully chosen, according to the
391 available hardware and envisioned scenario.

392 Further parameters, introduced in Sec. 5, that refer to the CMOS sen-
393 sor characteristics, are optional but can be considered to refine the channel
394 model, as they impact the ROI. However, smartphone manufacturers rarely
395 provide this information, e.g. regarding the sensor size ss and the focal
396 distance fc .

397 The PHY-SDU error rate (PER) p_e is the consequence of the errors oc-
398 ccurring in a M state when a PHY-SDU is well included in a picture but
399 is wrongly decoded by the smartphone. These errors are bits substitutions
400 induced by interference, low SNR, and artifacts on the picture. This value
401 varies from a smartphone to another, but we did not observe major differ-

Param.	Description	Default Value
d	Distance between camera and LED (cm)	5
l	LED size (mm)	4
d_g	Camera inter-frame gap ratio	0.1
p_e	Decoder PHY-SDU Error Rate	0.001
f	Modulation frequency (Hz)	8000
P	<code>PhysduPayload</code> Header length (bit)	8
H	<code>PhysduPayload</code> Payload length (bit)	16
r	PHY-SDU repeat number	1
G	Message size (bytes)	50

Table 1: Simulator parameters and default values.

ences in our tests.

The `PhysduPayload` payload size P and the `PhysduPayload` header size H configure the data encapsulation mechanism. Given these two settings, the PHY-SDU size is computed as $P + H + SYNC + PB$, where $SYNC$ is the PHY-SDU delimiter symbol (4 bits in our tests), and PB is the number of parity bits (2 bits in our settings). This PHY-SDU size, along with d , l , δ_f and the modulation frequency f , determines the number of the M time slots in Fig. 6.

A transmission strategy among the `Strategy` interface implementations needs to be chosen as well. For now, we have implemented and considered only the RP strategy, for which the parameters are the size of the message to broadcast, G , and the number of consecutive PHY-SDU emissions, r .

6.2.3. Kernel Implementation

The `CamComSim` kernel is implemented in the `Simulator` class. Its role is to produce PHY-SDU emission events (TX) and manage their result. The number of events, i.e. the number of PHY-SDU sent, is noted c and is updated at runtime. At each clock tick, c is incremented, a TX event is created and processed as follow: (1) the next PHY-SDU in the transmission strategy queue is associated with this event; (2) considering p_e , f , P , H , c , the channel response function gives the event result. The result is one among: *reception success*, *reception with errors* or *loss during IFG*; (3) this result is stored in a list to further determine if all the PHY-SDU forming a

424 message are received.

425 The simulator loops over (1), (2) and (3) until the stop condition is met,
426 i.e. c has reached the maximum number of PHY-SDU emissions or the com-
427 plete message is received. The simulation is repeated n_r times using the Java
428 class for multi-threading purpose `ThreadPoolExecutor`. Finally, the simu-
429 lations results and statistics are saved in a JSON file for further processing.

430 6.3. *CamComSim* validation

431 In this section, we present LED-to-camera simulation results given by
432 *CamComSim*. To assess the correctness of our simulator, we conduct a se-
433 ries of experiments with the testbed presented in Sec. 3. We set the emitter
434 symbol rate to 8 kHz and place it in standard indoor illumination conditions,
435 near a window and illuminated with neon lights. The illuminance has been
436 measured with a luxmeter at around 650 lux. We compare the testbed per-
437 formance with the results given by *CamComSim* for a set of key parameters:
438 the message size G , the number of consecutive PHY-SDU emissions r , the
439 distance d and the PHY-SDU payload length P .

440 6.3.1. PHY-SDU Retransmission

441 As discussed in Sec. 4.2, to face the IFG bits erasure and ensure that
442 all the packets are well received, a possibility is to transmit consecutively
443 each PHY-SDU r times in a row. The `RepeatPhysduStrategy` class in
444 *CamComSim* implements this retransmission strategy.

445 Fig. 14 shows the goodput at 5 cm for different values of PHY-SDU
446 consecutive retransmissions r , with $G = 50$, $P = 19$, $H = 5$. When each
447 PHY-SDU has been transmitted r times, the message transmission restarts,
448 until the message is completely received. To avoid infinite loops, we stop
449 the simulation when 50000 PHY-SDU are sent, even if the message is not
450 received entirely. In such case, we consider the goodput is 0.

451 The results highlight that the simulation and testbed goodput follow the
452 same tendency when r varies. The best case is when $r = 1$, for which
453 the goodput is 1.6 kbit/s according to *CamComSim* and 1.7 kbit/s for the
454 testbed, an estimation error of only 6%. Based on these results, for all the
455 simulations that follow we use the RP strategy implementation with $r = 1$.

456 6.3.2. Message size

457 We now consider the impact of the message size G on the goodput at a 5
458 cm distance, with $r = 1$ and a 24 bits length PHY-SDU. Fig. 15 shows that

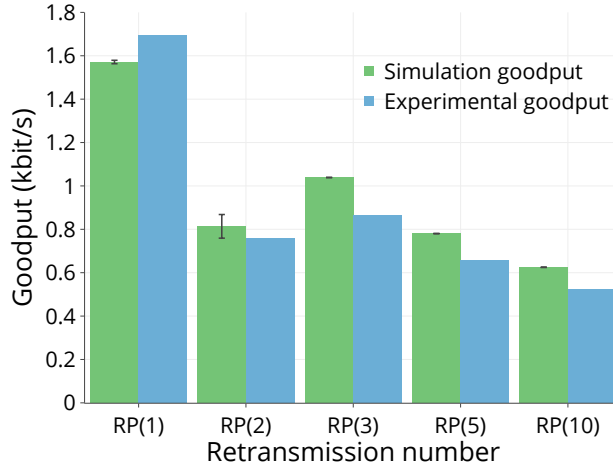


Figure 14: The experimental goodput (blue) compared with the simulation goodput (green) as a function of the number of consecutive PHY-SDU emissions. The bars on top are 95% confidence intervals.

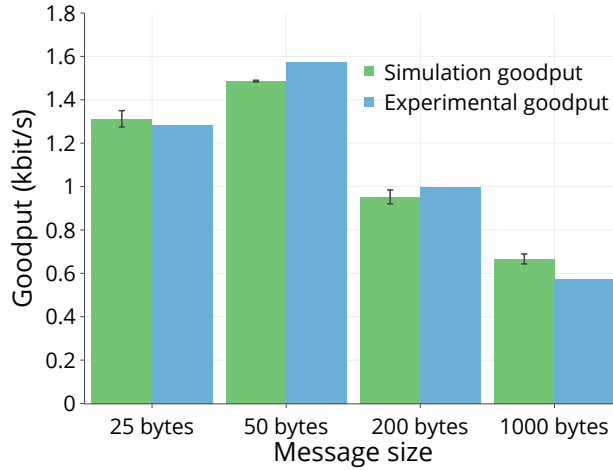


Figure 15: The experimental goodput (blue) compared with the simulation goodput (green) for different message size (G , bytes).

459 *CamComSim* results are very close to those of the testbed, confirming that
 460 the simulator well considers the impact of the message size. The goodput
 461 reduces when the message size increases, as the RP strategy leads to a large
 462 number of useless transmissions: the simulator gives 1.6 kbit/s of goodput
 463 for $G = 50$ bytes, while this falls to 670 bit/s when $G = 1000$ bytes. These

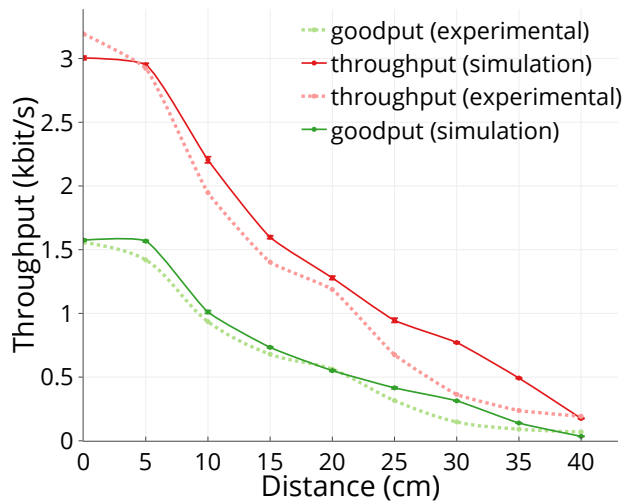


Figure 16: The throughput (red) as a function of the distance, compared with the goodput (green). Dotted-lines show experimental results while plain lines represent simulation results.

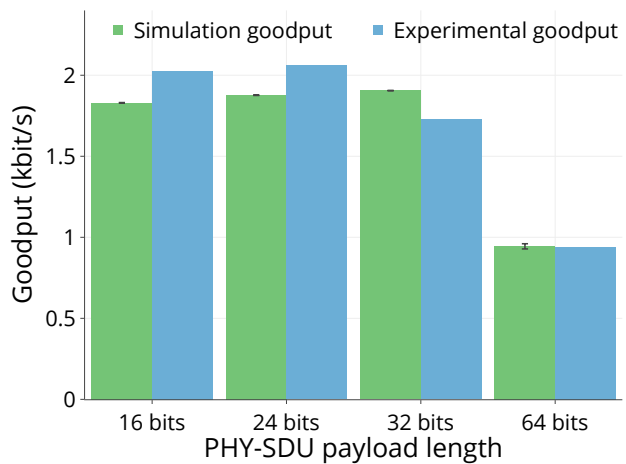


Figure 17: The experimental goodput (blue) compared with the simulation goodput (green) for different PHY-SDU payload size (bit).

464 results differ from the testbed in no more than 7%.

465 6.3.3. Distance

466 Fig. 16 shows the goodput and the throughput as a function of the dis-
 467 tance, when the LED broadcasts a 50 bytes message. The PHY-SDU payload

468 is set to 24 bits, with $P = 19$ and $H = 5$. The results show a good match
469 between the simulation and real life results. At 10 cm, *CamComSim* gives
470 2.2 kbit/s of throughput, against 1.94 kbit/s experimentally. The results are
471 closer for the goodput: 0.94 and 1.0 kbit/s respectively for simulation and
472 experimentation, that is only 6% of difference.

473 6.3.4. PHY-SDU length

474 Fig. 17 shows the impact of the PHY-SDU payload size on the goodput
475 at 5 cm, with $G = 50$. The packets are built using the best value for P and
476 H , that is to say with just enough bits in the header to label each packet
477 with a unique sequence number. Both experimental and simulation results
478 show that the optimal PHY-SDU payload size is 24 bits: *CamComSim* gives
479 a goodput of 1.9 kbit/s, while the testbed reaches 2.1 kbit/s. These results
480 outline that using large PHY-SDUs reduces the encapsulation overhead, but
481 increases the probability that the IFG and a small ROI truncate the PHY-
482 SDU. Fig. 17 brings out that *CamComSim* well considers this behavior: the
483 goodput reaches 0.9 kbit/s for a PHY-SDU payload of 64 bits, very close to
484 the experimental results.

485 Overall, this entire evaluation highlights that *CamComSim* gives results
486 very close to the testbed for all the parameters we have studied. The differ-
487 ence is around 10% and often less. For all the cases we consider, *CamComSim*
488 respects the behavior of the LED-to-camera communication system imple-
489 mented by the testbed.

490 6.4. Use case

491 In this section, we detail a case study for *CamComSim*, applied to a
492 real life scenario. A common issue with cheap consumer electronics is the
493 lack of diagnostics when a dysfunction happens. Manufacturers often blink
494 the state LED with a pattern and color that match with an error code.
495 Such a mechanism is easy to implement but leads to inaccurate diagnostics.
496 For these cases, we propose to benefit from this LED to perform LED-to-
497 camera communication and broadcast a log file that would include helpful
498 information to diagnose a dysfunction. We consider a worst case file size of
499 1 kbyte that is large enough for events history or debug traces.

500 Fig. 18 compares the goodput given by *CamComSim* with the goodput
501 that our testbed achieved for the transfer of a 1 kbyte log file as a function
502 of the number of PHY-SDU retransmissions r . Note that this is equivalent
503 to $G = 1000$ bytes in Fig. 15. The transmission restarts until the message

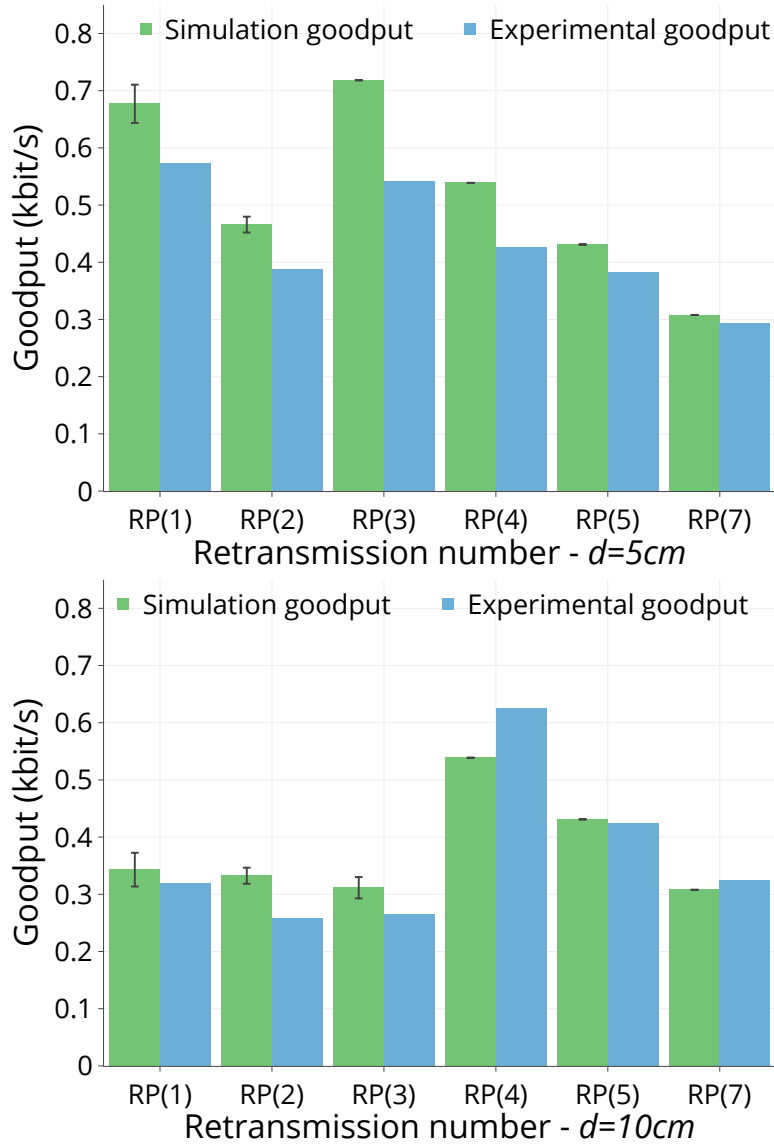


Figure 18: The experimental goodput (blue) compared with the simulation goodput (green) for the use case as a function of the number of consecutive PHY-SDU emission at 5 cm (left) and 10 cm (right).

504 is received. The left side plot shows the results when the LED and the
 505 smartphone are 5 cm apart, while the distance is 10 cm on the right
 506 side figure. At 5 cm, the simulation brings out that, to obtain the higher goodput,

507 the emitter should send each PHY-SDU one or three times consecutively, i.e.
508 $r = 1$ or $r = 3$. The goodput is respectively 680 and 720 bit/s in these cases.
509 This finding is similar to the testbed, where the goodput is 570 bit/s when
510 $r = 1$ and 540 bit/s when $r = 3$.

511 Because the ROI decreases with the distance, the behavior is different
512 when the smartphone is 10 cm far from the LED. In this situation, $r = 4$
513 stands out clearly to be the best choice both for the simulation and the
514 experiments. The goodput then becomes 620 bit/s on the testbed and 540
515 bit/s with *CamComSim*.

516 Since the results are very close to the reality, using *CamComSim* highly
517 reduces the search space for the experimental optimization of a system. As
518 shown by these results, the best value for r can be decided using simulations
519 only, removing the need for a lengthy experimental campaign.

520 7. Conclusion

521 In this paper, we introduced *CamComSim*, the first simulator for the de-
522 sign, the prototyping and the development of protocols and applications for
523 LED-to-camera communication. Our event driven simulator is based on an
524 MMBP channel model, and it relies on a standalone Java application that
525 is easily extensible through a set of interfaces. We have validated *CamCom-*
526 *Sim* comparing simulation results with the performance reached by a real life
527 testbed. Then, we illustrated with a practical use case the complete usage of
528 *CamComSim* to tune a broadcast protocol that implements the transmission
529 of a 1 kbyte log file. The results highlight that our simulator is very precise
530 and can predict the performance of a LED-to-camera system with less than
531 10% of error in most cases. The availability of accurate performance evalu-
532 ation tools offers a great ease of use and the opportunity to tune protocols
533 without the burden of always realizing experiments on a testbed.

- 534 [1] M.M. Galal, A.A. El Aziz, H.A. Fayed, M.H. Aly. “Employing Smart-
535 phones Xenon Flashlight for Mobile Payment”. *Proc. IEEE SSD 2014*,
536 Barcelona, Spain, Feb. 2014.
- 537 [2] Q. Wang, M. Zuniga, D. Giustiniano. “Passive Communication with Am-
538 bient Light”. *Proc. ACM CoNEXT 2016*, Irvine, CA, USA, Dec. 2016.
- 539 [3] C. Danakis, M. Afgani, G. Povey, I. Underwood, H. Haas. “Using a
540 CMOS Camera Sensor for Visible Light Communication”. *Proc. IEEE*
541 *OWC 2012*, Anaheim, CA, USA, Dec. 2012.

- 542 [4] T. Nguyen, Y.M. Jang. “High-speed Asynchronous Optical Camera Com-
543 munication using LED and Rolling Shutter Camera”. *Proc. ICUFN 2015*,
544 Sapporo, Japan, Jul. 2015.
- 545 [5] P.H. Pathak, X. Feng, P. Hu, P. Mohapatra. “Visible Light Communi-
546 cation, Networking, and Sensing: A Survey, Potential and Challenges”.
547 *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2047–2077,
548 Oct. 2015.
- 549 [6] H.-Y. Lee, H.-M. Lin, Y.-L. Wei, H.-I. Wu, H.-M. Tsai, C.-J. Lin.
550 “RollingLight: Enabling Line-of-Sight Light-to-Camera Communica-
551 tions”. *Proc. ACM MobiSys 2015*, New York, NY, USA, May 2015.
- 552 [7] J. Hao, Y. Yang, J. Luo. “CeilingCast: Energy Efficient and Location-
553 bound Broadcast through LED-camera Communication”. *Proc. IEEE IN-
554 FOCOM 2016*, San Francisco, CA, USA, Apr. 2016.
- 555 [8] H. Du, J. Han, X. Jian, T. Jung, C. Bo, Y. Wang, X.-Y. Li. “Martian:
556 Message Broadcast via LED Lights to Heterogeneous Smartphones”.
557 *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp.
558 1154-1162, May 2017.
- 559 [9] J. Ferrandiz-Lahuerta, D. Camps-Mur, J. Paradells-Aspas. “A Reliable
560 Asynchronous Protocol for VLC Communications Based on the Rolling
561 Shutter Effect”. *Proc. IEEE GlobeCom 2015*, San Diego, CA, USA, Dec.
562 2015.
- 563 [10] C. Zhang, X. Zhang. “LiTell: Robust Indoor Localization using Unmod-
564 ified Light Fixtures”. *Proc. ACM MobiCom 2016*, New York, NY, USA,
565 Oct. 2016.
- 566 [11] S. Zhu, X. Zhang. “Enabling High-Precision Visible Light Localization
567 in Todays Buildings”. *Proc. ACM MobiSys 2017*, Niagara Falls, NY, USA,
568 Jun. 2017.
- 569 [12] T. Li, C. An, Z. Tian, A.T. Campbell, X. Zhou. “Human Sensing Us-
570 ing Visible Light Communication”. *Proc. ACM MobiCom 2015*, Paris,
571 France, Sep. 2015.

- 572 [13] S. Zhu, C. Zhang, X. Zhang. “Automating Visual Privacy Protection
573 Using a Smart LED”. *Proc. ACM MobiCom 2017*, Snowbird, UT, USA,
574 Oct. 2017.
- 575 [14] S. Ozekici. “Markov Modulated Bernoulli Process”. *Mathematical Meth-*
576 *ods of Operations Research*, vol. 45, no. 3, pp. 311-324, Mar. 1997.
- 577 [15] E.O. Elliott. “Estimates of Error Rates for Codes on Burst-Noise Chan-
578 nels”. *Bell System Technical Journal*, vol. 42, no. 5, pp. 1977-1997, May
579 1963.
- 580 [16] F.M. Mims. “Alexander Graham Bell and the Photophone: The Centen-
581 nial of the Invention of Light-Wave Communications, 1880-1980”. *Optics*
582 *News*, vol. 6, no. 1, pp. 8-16, Jan. 1980.
- 583 [17] T. Li. “Advances in Optical Fiber Communications: An Historical Per-
584 spective”. *IEEE Journal on Selected Areas in Communications*, vol. 1,
585 no. 3, pp. 356-372, Apr. 1983.
- 586 [18] S. Naribole, S. Chen, E. Heng, E. Knightly. “LiRa: A WLAN Architec-
587 ture for Visible Light Communication with a Wi-Fi Uplink”. *Proc. IEEE*
588 *SECON 2017*, San Diego, CA, USA, Jun. 2017.
- 589 [19] H. Haas, L. Yin, Y. Wang, C. Chen. “What is LiFi?”. *Journal of Light-*
590 *wave Technology*, vol. 34, no. 6, pp. 1533-1544, Mar. 2016.
- 591 [20] Y. Yang, J. Luo, C. Chen, Z. Chen, W.-D. Zhong, L. Chen. “Pushing the
592 Data Rate of Practical VLC via Combinatorial Light Emission”. *IEEE*
593 *Transaction on Mobile Computing*, online, Feb. 2020.
- 594 [21] N. Saeed, S. Guo, K.-H. Park, T.Y. Al-Naffouri, M.-S. Alouini. “Optical
595 Camera Communications: Survey, Use Cases, Challenges, and Future
596 Trends”. *Physical Communication*, vol. 37, no. 12, pp. 1-17, Dec. 2019.
- 597 [22] A. Duque, R. Stanica, H. Rivano, A. Desportes. “SeedLight: Harden-
598 ing LED-to-Camera Communication with Random Linear Coding”. *Proc.*
599 *ACM VLCS 2017*, Snowbird, UT, USA, Oct. 2017.
- 600 [23] P. Hu, P. Pathak, H. Zhang, Z. Yang, P. Mohapatra. “High Speed LED-
601 to-Camera Communication using Color Shift Keying with Flicker Miti-
602 gation”. *IEEE Transaction on Mobile Computing*, online, Apr. 2019.

- 603 [24] Q. Wang, D. Giustiniano, D. Puccinelli. “An Open Source Research
604 Platform for Embedded Visible Light Networking”. *IEEE Wireless Com-*
605 *munications*, vol. 22, no. 2, pp. 94-100, Apr. 2015.
- 606 [25] A. Ageev, E. Luci, C. Petrioli, N. Thakker. “VuLCAN: A Low-cost, Low-
607 power Embedded Visible Light Communication And Networking Plat-
608 form”. *Proc. ACM MSWiM 2019*, Miami Beach, FL, USA, Nov. 2019.
- 609 [26] H. Zhang, F. Yang. “Push the Limit of Light-to-Camera Communica-
610 tion”. *IEEE Access*, vol. 8, pp. 55969-55979, Mar. 2020.
- 611 [27] M. Sharif, A. Sadeghi-Niaraki. “Ubiquitous Sensor Network Simulation
612 and Emulation Environments: A Survey”. *Journal of Network and Com-*
613 *puter Applications*, vol. 93, pp. 150181, Sep. 2017.
- 614 [28] D. Tagliaferri, C. Capsoni. “Development and Testing of an Indoor VLC
615 Simulator”. *Proc. IEEE IWOW 2015*, Istanbul, Turkey, Sep. 2015.
- 616 [29] A. Aldalbahi, M. Rahaim, A. Khreishah, M. Ayyash, R. Ackerman, J.
617 Basuino, W. Berreta, T.D. Little. “Extending ns3 to Simulate Visible
618 Light Communication at Network-level”. *Proc. IEEE ICT 2016*, Thessa-
619 loniki, Greece, May 2016.
- 620 [30] C. Ley-Bosch, R. Medina-Sosa, I. Alonso-Gonzalez, D. Sanchez-
621 Rodriguez. “Implementing an IEEE802.15.7 Physical Layer Simulation
622 Model with OMNET++”. *Proc. DCAI 2015*, Salamanca, Spain, Jun.
623 2015.
- 624 [31] A. Musa, M. D. Baba, H. M. Haji Mansor. “The Design and Implemen-
625 tation of IEEE 802.15.7 Module with ns-2 Simulator”. *Proc. IEEE I4CT*
626 *2014*, Lisbon, Portugal, May 2014.
- 627 [32] A. Duque, R. Stanica, H. Rivano, A. Desportes. “Unleashing the Power
628 of LED-to-Camera Communications for IoT Devices”. *Proc. ACM VLCS*
629 *2016*, New York, NY, USA, Oct. 2016.