KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

LightTour: Enabling Museum Audio Tour with Visible Light

Lennert Vanmunster

Thesis voorgedragen tot het behalen
van de graad van Master of Science
in de ingenieurswetenschappen:
elektrotechniek, optie Ingebedde
systemen en multimedia

Promotor:
Prof. dr. ir. Sofie Pollin

Assessoren:
Prof. dr. ir. Danny Hughes
Prof. dr. ir. Guy Vandenbosch

Begeleiders:
Ir. Jona Beysens
Dr. ir. Qing Wang

Academiejaar 2018 – 2019

Preface

I would like to thank my promotor, Prof. Sofie Pollin, for introducing me to this new and exciting topic. I would like to thank my supervisors, Jona Beysens and Qing Wang, for their expert guidance over the past year. Lastly, I would like to thank my parents for always supporting me.

Lennert Vanmunster

Contents

Abstract

Visible light communication (VLC) is a rapidly growing technology which uses light-emitting diodes (LEDs) to provide both illumination and low-cost, high-speed wireless communication. In this thesis we develop a VLC system for museum audio tours using the indoor LED lighting. In most current audio tour systems, the museum visitors carry a handheld device which stores audio locally and requires manual user input for audio track selection. Wireless audio tour systems which transmit audio to the visitors over a RF channel like Bluetooth also exist. Replacing RF with VLC technology can significantly lower installation cost and total power consumption. Like RF systems, VLC also offers the ability for automatic audio playback based on the user's position.

In this thesis we design, implement and test new high-efficiency communication protocols which can serve audio data to a large number of users. We develop a full duplex system where the VLC access points (APs) and user devices (UDs) each contain only a single LED and photodiode. Our measurements show that the self-interference at the AP, caused mainly by floor reflections, cannot be ignored. To solve this problem, we develop a digital self-interference cancellation (SIC) mechanism for VLC capable of 30dB of cancellation. Due to the properties of VLC channels SIC can be implemented in a very low-complexity way using a lookup table (LUT) in combination with real time estimates of the path loss. The established full duplex link is used to provide collision detection for a downlink broadcast VLC-CSMA/CD protocol. Through simulations and a small-scale hardware implementation we demonstrate the increased performance of VLC-CSMA/CD. For unicast links, the bandwidth efficiency is increased from 87% to 96%. In a large-scale simulation which also simulates the museum audio tour application layer, the reliable broadcast capability of VLC-CSMA/CD results in a decrease in mean user wait time from 25s to 5s at a user density of 1.5 UDs/m^2.

Additionally, the full duplex link created through SIC is used for uplink communication. We develop a novel uplink MAC protocol which solves the hidden UD problem by periodically interleaving the downlink data transmission with very short 'busy slots' (BS) which indicate the state of the uplink channel. The new uplink protocol has a bandwidth efficiency of 70% under high loads. A proof-of-concept is implemented with simple transmitter and receiver front-ends and low-cost Arduino Due microcontrollers.

Samenvatting

Visible light communication (VLC) ofwel ‘communicatie via zichtbaar licht’ is een sterk groeiende technologie die gebruik maakt van ‘light-emitting diodes’ (LED’s) voor zowel verlichting als razendsnelle en goedkoope draadloze communicatie. In dit eindwerk ontwikkelen we en VLC-systeem voor museum audiotours gebruikmakend van LED binnenverlichting. In bestaande audiotour systemen heeft elke museumbezoeker een draagbaar apparaat waarop geluidsopnames zijn opgeslagen. Om een opname te laten afspelen moet de bezoeker dit manueel ingeven op het apparaat. Verder bestaan er ook draadloze systemen waar de audio via radiogolven wordt verzonden naar de bezoeker (bv. via Bluetooth). VLC kan de kostprijs en het stroomverbruik van dergelijke draadloze system sterk verminderen. Verder laat VLC net zoals radiogolfsystemen ook een automatische selectie en transmissie van de geluidsopnames toe, afhankelijk van de positie van de bezoeker.

In dit eindwerk ontwikkelen, implementeren en testen we nieuwe hoog-efficiënte commumicatieprotocollen bedoeld voor audio transmissie naar een groot aantal bezoekers. Het ontwikkelde systeem maakt gebruik van een full-duplex link waar de VLC ‘access points’ (AP’s) en ‘user devices’ (UD’s) elk maar één enkele LED en photodiode bevatten. Onze metingen tonen aan dat aan de AP’s de eigen-interferentie, (self-interference) veroorzaakt door vloerreflectie, niet mag worden genegeerd. Daarom ontwikkelen we een digitaal self-interference-cancellation (SIC) mechanisme voor VLC. Onze oplossing kan de eigen-interferentie verminderen met 30dB. Wij tonen aan dat door de unieke eigenschappen van VLC kanalen de vereiste rekenkracht voor het SIC mechanisme sterk verminderd kan worden. Dit gebeurt door middel van een opzoektabel in combinatie met een real-time schatting van de path loss. Bestaande protocollen voor VLC laten geen vertrouwbare broadcast-transmissie op een goedkope en simpele manier toe. De full-duplex link gecreëerd met SIC laat wel een broadcast VLC-CSMA/CD-protocol toe. Met Monte Carlo simulaties en op een hardware platform tonen we aan dat de unicast doorvoer van VLC-CSMA/CD 10% hoger is dan het bestaande IEEE 802.15.7 CSMA/CA-protocol. Daarenboven toont een tweede simulatie in een kamer met een grid van AP’s sterkverbeterde prestaties. Bijvoorbeeld, de gemiddelde wachttijd voordat audio naar de gebruiker verzonden kan worden wordt verminderd van $25s$ in CSMA/CA naar $5s$ in CSMA/CD als de gebruikersdichtheid 1.5^2 UD’s/m^2 is. Bijkomend wordt de full duplex link gebruikt voor uplink transmissies. We ontwikkelen een nieuw uplink MAC protocol dat het ‘hidden node’ probleem oplost gebruikmakend van zeer korte ‘busy slots’ die periodisch in de verzonden downlink data zijn tussengevoegd. De busy slots geven een indicatie van de status van het uplinkkanaal. We noemen dit nieuw protocol ‘busy slot multiple access’ (BSMA). Monte Carlo simulaties van BSMA tonen een bandbreedte-efficiëntie van 70% aan. Alle protocollen worden volledig geïmplementeerd op een hardware setup gebaseerd op Arduino Due microcontrollers.

List of Figures and Tables

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

A-ACK	aggregated acknowledgement
AC	alternating current
ACK	acknowledgement
ADC	analog-to-digital converter
AP	access point
AWGN	additive white Gaussian noise
BCH	Bose-Chaudhuri-Hocquenghem
BE	backoff exponent
BEB	binary exponential backoff
BER	bit error rate
BS	busy slot
BSMA	busy slot multiple access
BTMA	busy tone multiple access
CAP	contention access period
CCA	clear channel assessment
CCR	continuous clock recovery
CD	collision detection
CELP	code-excited linear prediction
CFP	contention free period
CPU	central processing unit
CRC	cyclic redundancy check
CSMA	carrier-sense multiple access
CSMA/CA	carrier-sense multiple access with collision avoidance
CSMA/CD	carrier-sense multiple access with collision detection
CSK	color-shift keying
CT	continue transmission
DC	direct current

DL	downlink
DMA	direct memory access
EMA	exponential moving average
EMV	exponential moving variance
FEC	forward error correction
FER	frame error rate
FIFO	first in, first out
FIR	finite impulse response
FOV	field of view
GPIO	general-purpose input/output
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
IM/DD	intensity modulation / direct detection
I/O	input/output
IR	infrared
ISR	interrupt service routine
LED	light-emitting diode
LMS	least mean square
LOS	line-of-light
LUT	lookup table
MAC	medium access control
NB	number of backoffs
NLOS	non-line-of-light
NVIC	nested vector interrupt controller
NMOS	n-channel metal-oxide-semiconductor (field-effect transistor)
OCR	optical clock rate
OFDM	orthogonal frequency-division multiplexing
OOK	on-off keying
PAM	pulse-amplitude modulation
PC	personal computer
PD	photodiode
PHY	physical layer
PLL	phase-locked loop
ppm	parts per million
PPM	pulse-position modulation
PWM	pulse-width modulation
RAM	random access memory
RGB	red, green and blue
RSIC	residual self-interference after cancellation

RF	radio frequency
RFID	radio-frequency identification
RLL	run-length limited
RS	Reed-Solomon
RSSI	received signal strength indication
RX	receiver
SFD	start frame delimiter
SIC	self-interference cancellation
SNR	signal-to-noise ratio
SRAM	static random access memory
TC	timer counter
TDMA	time-division multiple access
TIA	transimpedance amplifier
TTP	training-test-pad
TX	transmitter
TW	wait time
UBP	unit backoff period
UD	user device
UL	uplink
USART	universal synchronous/asynchronous receiver transmitter
USB	Universal Serial Bus
USD	United States dollar
VLC	visible light communication
VLCC	Visible Light Communication Consortium
OWPAN	optical wireless personal area network
VPPM	variable-pulse-position modulation
XOR	exclusive or

Symbols

A	amplitude
$C(t)$	clock drift at time t
d_{hor}	horizontal distance between APs
d_{ver}	vertical distance between APs
f_{adc}	ADC sampling frequency
$f_{o,dl}$	downlink optical clock rate
$f_{o,ul}$	uplink optical clock rate
G	load
H	overall system gain
h_c	channel impulse response
h_r	receiver circuit impulse response
i	phase
i^*	best phase; the phase where samples as closest to the pulse center in the data stream
I	current
J	Jain's fairness index
k	sample index
$\ddot{k}$	Reed-Solomon message length
n	noise
$\ddot{n}$	Reed-Solomon block length
l_{bsi}	busy slot interval length in number of downlink symbols
O	overhead
p	probability parameter of p-persistent backoff
$P_{w,comm}$	average power consumption of AP TX front-end in communication mode
$P_{w,ill}$	power consumption of AP TX front-end in illumination mode
P_n	power of received noise n
P_s	power of signal of interest s
P_y	power of received signal y
P_ρ	power of the residual self-interference after cancellation
Q_{ul}	uplink MAC transmission frame buffer at the UD
s	received signal of interest
R	data rate
S	throughput, defined as the fraction of time the channel is used for successful transmission

S_u	downlink goodput, calculated as the throughput S times the average number of users receiving the same broadcast frame
t_{bsi}	busy slot interval length in seconds
$t_{f,dl}$	downlink frame length in seconds
$t_{f,ul}$	uplink frame length in seconds
t_{ct}	'continue transmission' deadline
T_{dl}	downlink symbol period
T_{ul}	uplink symbol period
V	DC bias added to received signal by RX front-end before ADC
W	oversampling rate in blind clock recovery
x	transmitted signal
y	received signal (including noise)
γ	photodiode responsivity
$\delta()$	error
η	channel efficiency
ι	lookup table index
ξ	luminous intensity
ρ	RSIC
σ_{cca}^2	variance in the CCA window
σ_{cd}^2	variance in the test sequence RSIC
τ_i	cross-correlation of the preamble with a template at phase i
$\phi_{1/2}$	LED half power semi-angle
χ	signal to residual self-interference after cancellation (RSIC) power ratio
Δ	MAC layer frame delay
Γ	LUT order
Λ	LUT

Chapter 1

Introduction

Audio tours have become ubiquitous in museums over the past few decades. An audio tour provides the visitor with spoken commentary about the museum's exhibition through a hand-held device. The visitor can experience the museum in a more personalized and effortless way. Furthermore, audio tours benefit people with (visual) disabilities [42]. This thesis aims to develop an audio tour system based on visible light communication (VLC). Audio transmission is done using white light emitting diodes (LEDs) which also provide room illumination. VLC enables an audio tour with an automatic location-based service. Consequently, minimal user-device interaction is required which greatly improves the user experience.

Figure 1.1: Audio guide set for Louvre [52]

1.1 Motivation

Modern audio tour systems can broadly be divided into four categories based on their operating principles [63]:

1. Locally stored audio with manual user input: these systems store the audio locally on the handheld device. They require user input (usually via buttons or a touchscreen) for audio track selection and playback.

2. Locally stored audio with automatic location-aware audio track selection: these systems sense the user's location using the Global Positioning System (GPS), radio-frequency identification (RFID) tags, wireless beacons or some other positioning method. The audio track stored on the user device is then played back based on the current user position.

3. Wireless audio transmission with manual user input: in these systems the user manually selects the audio track on the handheld device. Subsequently, the device sends a request to a nearby access point over an RF channel. The requested audio track which is stored on a server is then sent by the AP to the user over the RF channel.

4. Location-aware wireless audio transmission: these systems are the most advanced and include both a mechanism for user localization and wireless audio transmission over a radio-frequency (RF) channel.

LightTour can be classified as a location-aware optical wireless audio transmission system. This gives it several benefits over the first three categories. Firstly, being location-aware benefits the user experience as no longer need to spend time searching the relevant information on the user device [47]. For the same reason, it benefits the impaired. Secondly, storing the audio in the APs instead of on the user device allows for a more flexible system that support news future applications. For example, a possible application is sending museum announcements to the visitors in real-time. Lastly, it is possible to implement an uplink (UL) channel which can be used for various applications. One such application is providing user feedback on the museum exhibit in real time using a short voice message spoken by the user into a microphone at the UD.

Additionally, LightTour offers several advantages over the above four RF-based systems. These benefits generally overlap with the overall advantages of VLC over RF. The first and most important advantage is the reduced system complexity and cost of LightTour. Current systems require extra hardware installations of RF APs. LightTour, on the other hand, integrates the AP electronics into the existing lighting infrastructure. A second benefit is that by combining illumination and data transmission, overall power consumption is reduced.

Current VLC MAC protocols and standards provide high throughput and efficiency for a wide variety of network topologies. However, none of the existing protocols can offer both low system and processing complexity while also providing sufficient throughput for downlink audio transmission to a large number of users. For example, a first group of protocols requires a central coordinator or network wide synchronization which increases system complexity and cost [41]. A second solution uses an orthogonal frequency division multiplexing (OFDM) scheme for frequency based inter-cell interference mitigation [38]. This requires a relatively high processing power because computation of fast Fourier transforms (FFTs) is necessary. A third type of solution uses complex hardware circuits such as transceivers consisting of multiple LEDs and photodiodes [83], or frequency modulating circuits to provide frequency division multiple access (FDMA). Another well-known method is the Institute of Electrical and Electronics Engineers (IEEE) 802.15.7 carrier-sense multiple access with collision avoidance (CSMA/CA) which is only able to provide reliable unicast transmissions since it has no mechanism to handle broadcast collision [3]. Unicast transmission to many users is significantly less efficient than broadcast.

In addition, little research has been done on efficient uplink transmissions in full duplex VLC. In this work a novel high-efficiency full-duplex MAC protocol is designed and implemented. For the downlink, we design a MAC protocol that does not require a central coordinator, has low processing requirements and uses simple transmitter and receiver front ends.

1.2 Research Goals and Contributions

The primary goal of this thesis is to design and implement a VLC audio tour system with off-the-shelf electronic devices and the Arduino Due platform [7]. More specifically, we aim at developing a low-complexity system that can transmit audio to a large number of users simultaneously using the downlink (DL) channel, and to receive the feedback of users through the uplink (UL) channel. This above research goal has led to the following contributions of this thesis:

Contribution 1: Design the system architecture. In order to support both DL and UL communication with light and by considering the practical limitations (using visible light for UL is not user-friendly), we design a system architecture where we use a visible light DL channel and an infrared (IR) UL channel. The photodiodes (PDs) used at the AP and at the UD are all sensitive to both visible light and IR light. Our measurements show that the non-line-of-sight (NLOS) reflection of the DL signal can be detected by the AP that is transmitting the signal and also by other nearby APs. This phenomenon has led to the development of a low-complexity self-interference cancellation (SIC) mechanism and novel VLC DL and UL medium access control (MAC) protocols that can support the networking between multiple access points and many UDs.

Contribution 2: Design a low-complexity software clock recovery mechanism (PHY layer). We design, implement and test a mechanism for continuous clock recovery (CCR) during reception of a downlink frame at the user device. The CCR algorithm is based on 2-fold blind oversampling of the received data together with a novel low-complexity best phase picking method [37]. Under low SNR conditions, the CCR mechanism achieves close to the theoretical FER of a receiver with perfect clock recovery. Under high SNR conditions the FER is decreased from 7% without CCR to 0.05% with CCR.

Contribution 3: Propose a low complexity self-interference cancellation method (PHY layer). We propose and implement a low-complexity digital SIC mechanism based on a lookup table (LUT) trained by minimizing the mean square error of difference between the received signal and the expected signal. The LUT decreases the required CPU time by 90% compared to methods that uses the LMS algorithm to estimate the channel response [51]. The SIC mechanism is robust to changes in ambient light level and achieves up to 30dB of cancellation when the received signal strength is high. When the received signal strength is low, cancellation close to and sometimes below the noise floor is achieved. To our knowledge this is the first implementation of SIC in VLC. This is because most of the current research eliminates self-interference by using different wavelength transmitters and receivers for the downlink and the uplink or assumes the received self- interference signal strength is negligible compared to the received signal strength of the uplink signal. However, our measurements show that in a typical setup the received signal power of the reflected downlink signal is higher than the received uplink signal strength.

Contribution 4: Propose a novel DL VLC-CSMA/CD protocol (MAC layer). We develop a novel VLC MAC protocol based on carrier-sense multiple access with collision detection (CSMA/CD) [60, p.268,285] for DL audio transmissions. The protocol is specifically designed to provide broadcast transmissions with uncoordinated APs. Our VLC-CSMA/CD protocol uses software SIC to implement CD using energy detection [67]. Measurements on the hardware setup and using Monte Carlo simulations show that VLC-CSMA/CD achieves a channel throughput of 96% under high loads which is 10% greater than the current IEEE 802.15.7 standard [3]. Furthermore, Monte-Carlo simulations of a real-world setup with user density of 1.5 users/m^2 show a mean user wait time of 5s for broadcast VLC-CSMA/CD while 25s for 802.15.7. SIC allows APs to detect nearby users in a bandwidth and energy efficient way using a 'continue transmission' (CT) sequence transmitted by users when they detect a frame header for which they want to receive the corresponding payload. The CT sequence mechanism reduces wasted DL bandwidth occupation by up to 99.2% and by extension AP power consumption by 17%.

Contribution 5: Propose a novel UL busy slot multiple access method (MAC layer). We develop the second VLC MAC protocol for the UL inspired by busy tone multiple access (BTMA) [12]. UL transmissions happen during the transmission of DL frames. The novel MAC protocol uses 'busy slots' (BS) interleaved in time with the DL frame payload instead of a 'busy tone' in a dedicated frequency band in BTMA. Therefore, we call the new protocol busy slot multiple access (BSMA). BSMA solves the hidden node problem that originates from the fact that users are not able to sense each other's ongoing transmissions. BSMA does not increase transmitter (TX) and receiver (RX) hardware complexity like BTMA and trades this lower hardware complexity for 2.5% lower throughput at high loads. BSMA incurs a very low overhead of $5-7\%$ from UL transmissions on the DL throughput and Monte Carlo simulations show an uplink channel efficiency of 70% under high loads.

Contribution 6: Implement the whole LightTour system. A system consisting of three APs and two UD is built. All system design elements described in chapter 3 are implemented and functioning on the hardware setup. The VLC transceivers at each AP and user devices (UDs) are controlled by Arduino Due microcontroller boards. The link is full-duplex and physical layer data rates of 210kb/s on the DL and 100kb/s on the UL are achieved. Frame error rates of 0.01% for the DL and 2% for the UL are achieved. Extensive experiments are carried out. Results demonstrate that LightTour can provide good audio service to the end users and collect their feedback, and the proposed methods SIC, new MAC protocols, and the system implementation can together improve the throughput by one order of magnitude compared to other state-of-the-art systems that use Arduino.

1.3 Organization of this Thesis

In chapter 2 we give a brief introduction to VLC and provide an overview of the state-of-the-art research on VLC. Chapter 3 presents the system design consisting the topology and protocol stack. Chapter 4 presents the hardware and software implementation of the LightTour system. Modules for the AP and UD are built, together with a small-scale experimental implementation of the LightTour system. Next, in chapter 5, we evaluate the performance of the system both experimentally (small-scale) and with simulations (large-scale). Experimental evaluations can demonstrate the performance of the LightTour system for a small number of APs and users. The simulations are needed in order to study the system under a large number of APs and users. The thesis is concluded in chapter 6 by a vision of future improvements on the LightTour system.

Chapter 2

Visible Light Communication

This chapter gives a brief background on VLC. The VLC principles and system models are presented. Finally, we provide a survey on state-of-the-art research related to our work.

2.1 Background

Visible light communication is a subset of optical wireless communication in the visible spectrum. VLC is gaining a lot of traction over the decade from both the academia and the industry. The global market for VLC is expected to grow from USD 2.56 billion in 2018 to USD 75 billion in 2023 [55]. This growth is driven by a variety of reasons. Firstly, the solid-state lighting revolution has resulted in low-cost, long-lifespan, high-efficiency LEDs. In addition, LEDs are capable of switching at high frequencies (a few MHz) using simple and low-cost driving circuits [52]. Moreover, the visible spectrum provides a license-free bandwidth of several hundreds of THz. The RF spectrum, on the other hand, is limited to at most $\sim 300GHz$ where a large fraction of this bandwidth is occupied by licensed bands [65]. The RF spectrum congestion is only expected to grow in the future due to an explosive growth in mobile communications and internet of things (IoT) devices.

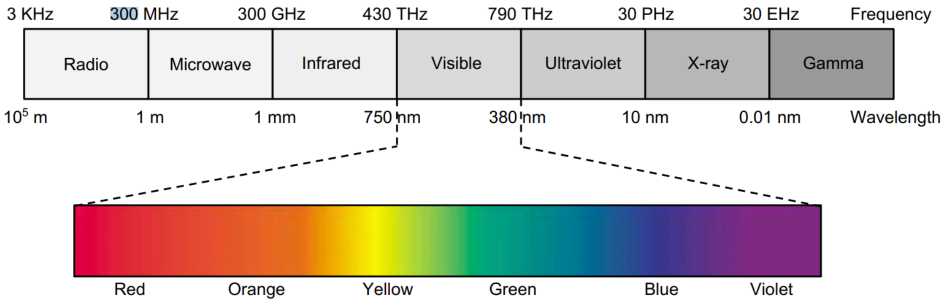


Figure 2.1: The electromagnetic spectrum, taken from [52].

Low cost LEDs and a wide, unlicensed spectrum are not the only causes for the emergence of this new multibillion-dollar industry. The high switching frequency of LEDs allows modulation of signals at a higher rate than the maximum frequency the human eye can perceive as flicker. As a result, data transmission and illumination can be combined, making VLC very power-efficient. The fact that visible light does not pass through opaque objects is one of the key distinctions which determines the possible secure applications for VLC compared to RF. The blocking of light also allows a high spatial reuse of the channel. Furthermore, the inability of VLC signals to penetrate walls provides an additional layer of security against eavesdropping. As a result, VLC is a promising candidate for the next-generation wireless indoor communications.

2.1.1 Brief History

Visible light communication has been used since ancient times to carry information over a distance. Many ancient civilizations used fire beacons or smoke signals to relay messages over a long distance. For example, the Chinese used fire beacons at night and smoke signal during the day along the Great Wall [34]. Similarly, the play ’Agamemnon’ by Aeschylus describes a relay fire that was used to announce the fall of Troy over hundreds of kilometres across continent borders [62]. A more widely known application of VLC are lighthouses which navigate ships in coastal waters by transmitting either a constant light level or modulated pulses. With the advent of electricity and telecommunication technology new ways were found to use visible light to carry messages. Signal lamps are used on ships since the late 19th century to relay messages by flashing signals in Morse code [77]. They are still used in modern military applications when radio communications need to be silent and in airports signal lamps provide a backup communication mechanism for air traffic control. In 1880, Alexander Graham Bell’s photophone was able to transmit wireless modulated voice messages over a distance of 213m [14]. The development of low-cost high-speed LEDs sparked a revolution in visible light communication. In 2000, the Keio research group in Japan demonstrated the viability of combining illumination and communication with white LEDs [59]. In 2003, the VLC Consortium (VLCC) was established [52]. The VLCC consists of mostly Japanese technology companies. VLCC promotes the adoption of VLC and specifies VLC standards. In 2011, the IEEE published the VLC Standard 802.15.7 [1], which defines a standard for the VLC physical (PHY) and MAC layer. The standard has seen its latest revision in 2018 [3].

2.1.2 Single Link System

A typical VLC link consists of an LED-based transmitter and a PD based RX. Figure 2.3 shows a schematic of such a link. The input signal $x(t)$ is used to modulate the instantaneous luminous flux of the LED using an intermediate driver circuit. The RX front-end converts the incident light intensity to a data signal using a photodiode followed by an amplifier with an optional filter. This type of communication

Figure 2.2: “US Navy sailor sending Morse code using a signal lamp,” taken from [77]

.

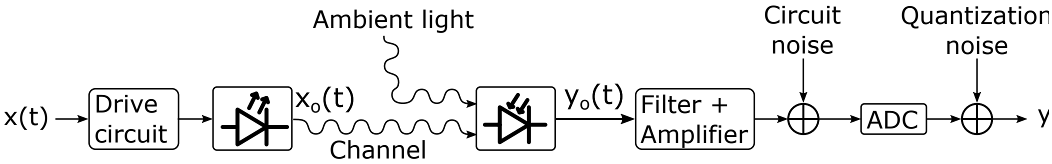


Figure 2.3: VLC system model with IM/DD and ADC demodulator, figure inspired by [31, p. 63].

mechanism is called intensity modulation with direct detection (IM/DD) [52] and allows simple and cheap TX and RX front-ends.

Types of LED There are two types of LEDs that can be used in an indoor VLC system which provide both illumination and communication: white phosphorescent LEDS and RGB (red, green and blue) LEDs. The bandwidth of a typical phosphorescent LED is limited to a few MHz because of the long relaxation time of the phosphor coating [19]. Techniques have been developed to increase the bandwidth of phosporescent white LEDs. For example, a bandwidth of 151MHz is achieved using a post-equalization circuit by Li et al. [43]. Phosphorescent LEDs are used in combination with simple modulation schemes like on-off-keying (OOK). RGB LEDs on the other hand do not need the bandwidth-limiting phosphor coating. As a result, they enjoy an increase in bandwidth without requiring additional equalization techniques. The disadvantage of using RGB LEDs for VLC is the tripling of hardware complexity of both the transmitter and the receiver front-end [32].

2.1.3 System Model

Referring to figure 2.3, the VLC channel with IM/DD can be modelled as follows [31, p. 79]:

$$y_o(t) = \gamma h_c(t) \otimes x_o(t) + n(t) \tag{2.1}$$

where $y_o(t)$ is the output signal of the receiver photodiode, γ is the photodiode responsivity, $h_c(t)$ is the channel impulse response, $x_o(t)$ is the transmitted optical signal (figure 2.3) and n is additive white Gaussian noise (AWGN). At high ambient light levels n is dominated by shot noise while at low ambient light levels thermal noise caused by resistive elements in the receiver dominates a VLC link [31, p. 67]. Shot noise arises due to the discrete nature of photons and electrons. Shot noise is greater for high ambient light levels. Thermal noise is mostly independent on the received signal and the ambient light level.

Next, we extend the system model in equation 2.1 to a model that finds an approximate relationship between the transmitted signal $x(t)$ and received signal $y(t)$ (figure 2.3). The extended model hence includes the effects of the drive circuit and LED at the TX and the filter, amplifier and ADC at the RX. The extended model is used by the SIC mechanism in section 3.4.2.

We use a phosphorescent LED and hence the modulation bandwidth is limited to a few MHz [19]. However, the overall bandwidth in our system is limited by the receiver bandwidth which contains a low pass filter with a cutoff frequency $f_{c,2} = 695kHz$ which is only a fraction of the modulation bandwidth of the LED. Hence, the limited modulation bandwidth of the LED can be ignored in our approximate model. Additionally, the LED luminous flux versus forward current relationship $\Phi(I)$ is nonlinear [21, p. 27]. For SIC, the nonlinearity of the LED can also be ignored since we use a binary modulation scheme (OOK) which only allows two discrete light levels to be transmitted. If another modulation scheme which uses multilevel symbol coding were used, the SIC mechanism has to solve the nonlinearity problem using e.g. pre-equalization at the transmitter [39].

Next, the signal passes through the optical channel modelled by equation 2.1. The RX circuit filters and amplifies the photodiode output signal $y_o(t)$. This process introduces nonlinear distortion and AWGN. After amplification and filtering a DC bias is added to the signal and the resulting signal is converted by an ADC which introduces quantization noise. The complete system model describing the relationship between $x[k]$ and $y[k]$ is thus:

$$y[k] = h_r[k] \otimes (\gamma \cdot h_c[k] \otimes x[k] + n_s[k]) + n_r[k] + n_q[k] + d_r[k] + V \tag{2.2}$$

where $h_r[k]$ is the impulse response of the receiver filter and amplifier, $h_c[k]$ is the channel impulse response, $n_s[k]$ is the photodiode additive white Gaussian shot noise, $n_r[k]$ is the RX circuit AWGN, $n_q[k]$ is the quantization noise, $d_r[k]$ is the nonlinear distortion of the RX circuit and V is the DC bias added before the ADC.

The SIC mechanism in section 3.4.2 uses a simplified version of the extended system model. In equation 2.2 all noise terms are grouped and the nonlinear receiver distortion is ignored. Furthermore, in a typical room setup the root mean square (RMS) delay spread of an NLOS link is around 10-20ns. On the downlink, our system uses an optical clock rate of $\Omega_{dl} = 500kHz$. An optical clock is defined as length of a symbol in seconds. Hence the length of one downlink symbol is $\frac{1}{\Omega_{dl}} = 2\mu s$ which is ~ 100 greater than the RMS delay spread of the channel. As a result, we can ignore inter-symbol interference caused by the channel and thus replace the impulse response of the channel $h_c[k]$ by its DC-gain $h_c[0]$ which we refer to as the channel path loss. We now obtain the simplified system model used by SIC:

$$y[k] = h_r[k] \otimes (h_c(0)x[k]) + n[k] + V \tag{2.3}$$

In section 3.4.2 we use this model to design the SIC mechanism.

2.1.4 Optical Path Loss

We briefly present the VLC line-of-sight optical path loss model as derived by [31]. Figure 2.4 shows an LOS link with the model parameters. The luminous intensity angular distribution of an LED light follows a Lambertian distribution:

$$I(\phi) = I(0)cos^m(\phi) \tag{2.4}$$

where ϕ is the angle of irradiance and m is the order of the Lambertian emission. The value of m is related to the LED's half power semi-angle $\phi_{1/2}$ usually provided in the datasheet:

$$m = \frac{ln(2)}{ln(cos(\phi_{1/2})}. \tag{2.5}$$

The direct current (DC) path loss in an optical LOS link is given by:

$$h_{los}(0) = \begin{cases} \frac{(m+1)A_r}{2\pi d^2} cos(\psi) cos^m(\phi) & 0 \leq \psi \leq \Psi_c \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

where A is the physical area of the PD detector, d is the distance between transmitter and RX, ψ is the angle of incidence, ϕ is the angle of irradiance, $T_s(\psi)$ is the gain of the optical filter, $g(\Psi)$ is the gain of the optical concentrator and Ψ_c is the FOV of the receiver.

2.1.5 Flicker Mitigation and Dimming

Visible light communication combined with illumination has two unique challenges not present in other communication technologies: flicker mitigation and dimming support. Flicker is the fluctuation in light intensity that can be perceived by the human eye. Flicker must be avoided as it is unpleasant and moreover can have harmful psychological effects on humans [54]. The IEEE 802.15.7 standards defines the maximum flickering time period (MFTP) as: “the maximum time period over

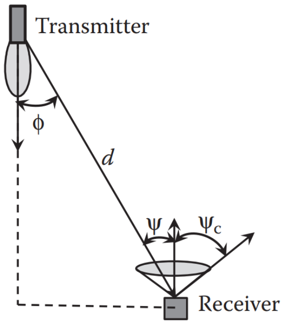


Figure 2.4: LOS propagation model geometry, taken from [31, p. 82]
.

which the light intensity can be changing, but for which the resulting flicker is not perceivable by the human eye" [3]. A value of MFTP=5ms is generally considered safe [54]. The 5ms period corresponds to a minimum modulation frequency of 200Hz.

A VLC transmitter must mitigate flicker both during transmission and while idle. During frame transmission long strings of consecutive low or high pulses reduce the effective frequency of the transmitted signal. In IM/DD schemes, a run-length limited (RLL) and DC-balanced line code is used for intraframe flicker mitigation [52]. An RLL line code prevents long strings of consecutive LOW or HIGH symbols. The DC balance of the code ensures that the average light output remains the same within a short time period. In IEEE 802.15.7 interframe flicker mitigation is achieved by transmitting a predefined 'idle pattern' with an average brightness equal to that during the data frame [3]. The idle pattern can either be an in-band or an out-of-band signal.

Light dimming is the ability to change the perceived brightness of the light according to the user's preference. Light dimming is outside of the scope of this work but the LightTour system can upgraded in the future with a PHY layer that implements dimming support using one of the described dimming methodologies in 802.15.7 [3]. In order to provide optimal lighting conditions, museum lighting is often a complex system of lights at varying intensities. Furthermore, an adaptable lighting system is needed to provide optimal lighting conditions if the exhibition changes over time. In conclusion, implementing light dimming could greatly enhance the viability of LightTour for actual museum use.

2.1.6 Modulation Schemes

Various IM/DD-based modulation schemes are suitable for VLC. We briefly present four of the most common schemes [52]:

- **On-off keying (OOK)**: each bit is transmitted as either a high or a low pulse to represent a binary 1 and 0 respectively. M-ary pulse-amplitude modulation

(MPAM) is a generalization of on-off keying which uses M light levels in each pulse to represent a symbol of $\log_2 M$ bits.

- **M-ary pulse position modulation (MPPM)**: it encodes M message bits by transmitting a single pulse in one of 2^M possible positions.
- **Color Shift Keying (CSK)**: it is a scheme designed for RGB LEDs. Data is represented as different color points on the space chromaticity diagram while overall perceived light is kept constant.
- **OFDM**: is a technique also used in wireless systems where inter symbol interference can greatly be reduced by transforming the signal to the frequency domain.

2.2 Related Work

2.2.1 IEEE 802.15.7

This section gives a brief description of the principle of operation and capabilities of the IEEE 802.15.7 standard. Special attention is given towards the MAC layer of the standard as a novel MAC protocol is one of key areas of research in this thesis. The standard defines both a PHY and MAC layer for short-range optical wireless communications. The base standard was published in 2011 [1] and the most recent revision was done in 2018 [3].

2.2.2 Topologies

Three possible topologies are defined: peer-to-peer, star and broadcast. In a peer-to-peer network each device can communicate directly with any other device within range. One of the peers acts as the coordinator. A star-topology consists of one central device acting as coordinator. All other devices can only communicate with the central device. In addition, all star-networks operate independently. This is achieved by associating a unique OWPAN identifier with each coordinator. The broadcast topology is a simpler version of the star-topology. The difference is that no device association is necessary and that messages are broadcast by the central device to all other devices in range.

2.2.3 Physical layer

The 2011 version of the 802.15.7 standard provides three types intended for different applications:

- **PHY I**: is intended for outdoor applications. This operating mode uses OOK or VPPM and supports data rates in the tens to hundreds of kb/s.
- **PHY II**: is intended for indoor applications with white (phosphorescent) LED's. This operating mode also uses OOK or VPPM but due to the shorter

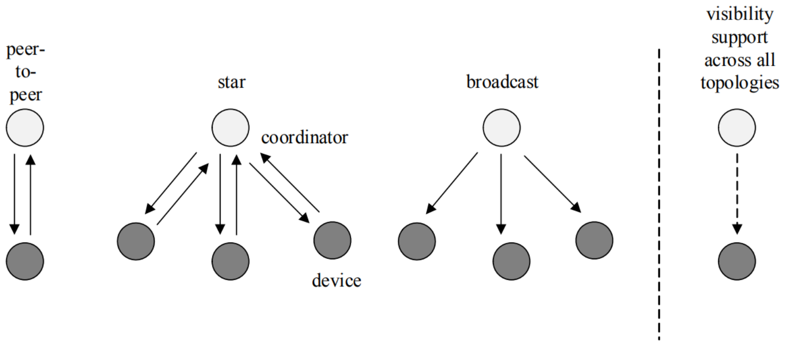


Figure 2.5: IEEE 802.15.7-2018 MAC topologies

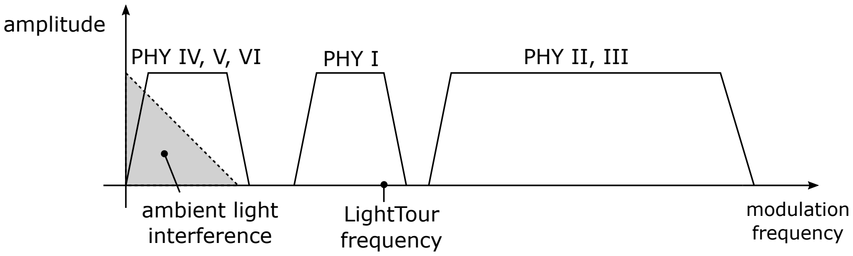


Figure 2.6: IEEE 802.15.7-2018 PHY types over the modulation domain spectrum with an indication of LightTour's current modulation frequency, figure reproduced from [3, p. 25].

symbol lengths (i.e. a higher optical clock rate (OCR)) is able to achieve data rates in the tens of Mb/s.

- **PHY III**: is intended for indoor applications with RGB LED's and uses CSK to achieve data rates in the tens of Mb/s.

In the 2018 standard three more PHY types are defined providing low data rate transmissions (maximum 22kb/s). PHY IV, PHY V and PHY VI are intended for discrete light sources, diffused surface light sources and video displays respectively. Each PHY type is further split into a large selection of operating modes. An operating mode is defined by its modulation type, OCR, the choice of RLL-code and the error correction code. A full list of operating modes can be found in tables 76 to 79 in [3]. The optional error correcting code described in the standard is Reed-Solomon (RS). Flicker mitigation and dimming support are a cross layer function between the PHY and MAC layer. The RLL and DC-balanced codes that are used in the standard are in increasing order of complexity and channel efficiency: Manchester, 4b6b and 8b10b.

2.2.4 MAC layer

The media access control protocol can be operated with or without beacon frames. In a beacon-enabled VPAN the coordinator broadcasts a beacon frame periodically in

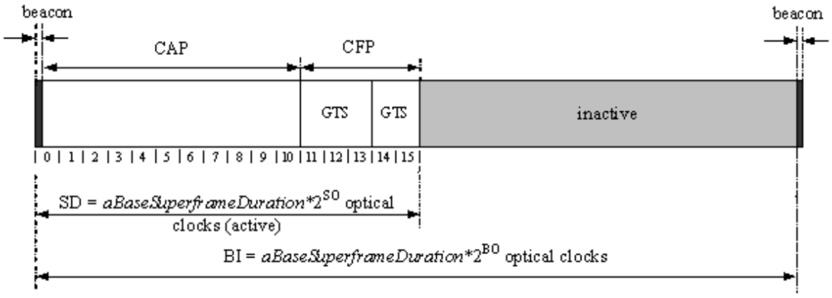

beacon
beacon
CAP
CFP
GTS
GTS
inactive
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
SD = aBaseSuperframeDuration*2^SO optical clocks (active)
BI = aBaseSuperframeDuration*2^BO optical clocks

Figure 2.7: IEEE 802.15.7-2018 superframe structure [3, p.41]

order to associate, synchronize and coordinate other nodes within the VPAN. Other functions provided by the MAC layer that are not already mentioned are mobility support and providing device security.

The MAC protocol of the IEEE standard uses an optional superframe structure in order to coordinate device transmissions in the VPAN. The superframe is bounded by beacon frames periodically broadcast by a central coordinator. All devices in the VPAN synchronize to the beacon frame. When the superframe structure (and by extension the beacon frames) is disabled, channel access is done using a unslotted CSMA/CA. Clear channel assesment (CCA) is optional and when it is disabled, the random-access algorithm is similar to unslotted Aloha. Frames with the Acknowledgement Request header field set to one require and acknowledgement frame (ACK) by the RX. The ACK is transmit without the random-access algorithm after a short interval after the last optical clock of the data frame.

A simplified version of the superframe structure is shown in figure 2.7. It consists of a contention access period (CAP) and a contention free period (CFP). In the CAP, channel access is done using a slotted version of CSMA/CA. The slots in the CFP period are assigned by the coordinator by request of the devices in the VPAN. Reservation of a slot for multiple superframes is possible. If a device is assigned a slot, only that device can transmit during that slot (guaranteed time slot, GTS). The reservation mechanism is useful for applications that require a fixed bandwidth and/or bounded access times such as real-time and multimedia applications.

Limitations The IEEE 802.15.7 MAC protocol is a flexible protocol with many modes of operation defined. However, in a large and dense system of APs, network-wide synchronization is difficult. As a result, the protocol then consists of unslotted CSMA/CA which can only provide reliable unicast transmissions.

2.2.5 Other State-of-the-art Work

This thesis discusses a SIC mechanism which provides support for both full duplex and collision detection SIC. These topics are well-studied in RF communications. For example, Ahmed et al. [4] present an all-digital SIC mechanism for RF systems. In their work the channel is estimated using a least square estimator with time-orthogonal (uncorrelated) training sequences at the start of each frame. The derivation of our VLC system model in equation 2.2 is partly inspired by their work. Collision detection based on SIC has been extensively studied by Vermeulen et al. [66].

The performance of full-duplex visible light communication networks is studied by Zhang et al.[82]. They propose two full duplex contention protocols: U-ALOHA and FD-CSMA. They state that in indoor VLC environments self-interference can be ignored because of the lower received power from the NLOS self-interference compared to the LOS uplink. Our research does not make this assumption. The first protocol, U-ALOHA, is a slotted version of the Aloha protocol for contention access on the uplink channel. In the second protocol, FD-CSMA, users sense the channel before transmission. If the channel is clear, the user starts transmitting a uplink frame. When the AP detects the uplink frame, the AP transmits 'busy symbols' for the entire duration of the uplink frame. Alternatively, if the AP has any queued downlink frames, a downlink frame is transmitted by the AP instead of the busy symbols. Nearby users sense that the downlink channel is busy and defer their transmissions to a later time. They study both protocols using theoretical analysis and Monte Carlo simulations.

Wang. et al [70] have developed a CSMA/CD protocol with hidden avoidance for a VLC system with LED-to-LED communication. Bidirectional communication is achieved using an OOK modulation scheme where the uplink channel is sensed during the transmission of LOW symbols. The bidirectional link furthermore allows collision detection. Little research has been done on combining this technique with a high data rate PHY layer. The classification accuracy of the collision detection mechanism under low SNR conditions is also not studied.

2.3 Conclusion

A brief introduction has been given on VLC. We have derived the system model to include distortions caused by the transmitter and receiver front-ends. We have provided a short summary on the IEEE standard and some state-of-the-art research on solutions comparable to our system. In the following chapter we design the LightTour system.

Chapter 3

System Design

This chapter specifies the design of LightTour. The design is implemented on a hardware setup which will be discussed in the next chapter. We first provide a high-level overview of the system. The remainder of the chapter presents the LightTour protocol stack in detail, starting with the MAC layer. The MAC layer consists of both a DL and an UL MAC protocol. Next, we present the PHY layer which provides the new SIC, collision detection and CCR schemes. Lastly, we present the application layer which provides DL audio transmission and UL text transmission.

3.1 General Overview

3.1.1 Topology

The topology of LightTour is shown in figure 3.1. The system consists of an even distribution of fixed access points (APs), each with a white LED transmitter and PD-based RX. The APs are mounted on the ceiling and provide both illumination and communication. The protocol does not require wired or wireless communication between APs which greatly reduces installation cost and system complexity. Like a normal audio tour system, each user is provided with a handheld user device (UD), through which the audio is played back. In LightTour the handheld device also contains a PD RX and IR LED transmitter for communication with APs.

3.1.2 Downlink and Uplink

Each AP broadcasts audio data using its white LED to the UDs within its communication range. There is an LOS path between the white LED at the AP and the PD at the UD. We call this link the DL. Likewise, the UD transmits data using its IR LED to the AP on an LOS path. We call this link the UL. A potential application for the UL channel is the ability for the user to provide feedback about the museum exhibition to the AP. This feedback can either be in the form of a like/dislike button, a text message or even a spoken audio message. In the current version of LightTour the feedback is implemented in the form of a text message. In a future revision, the

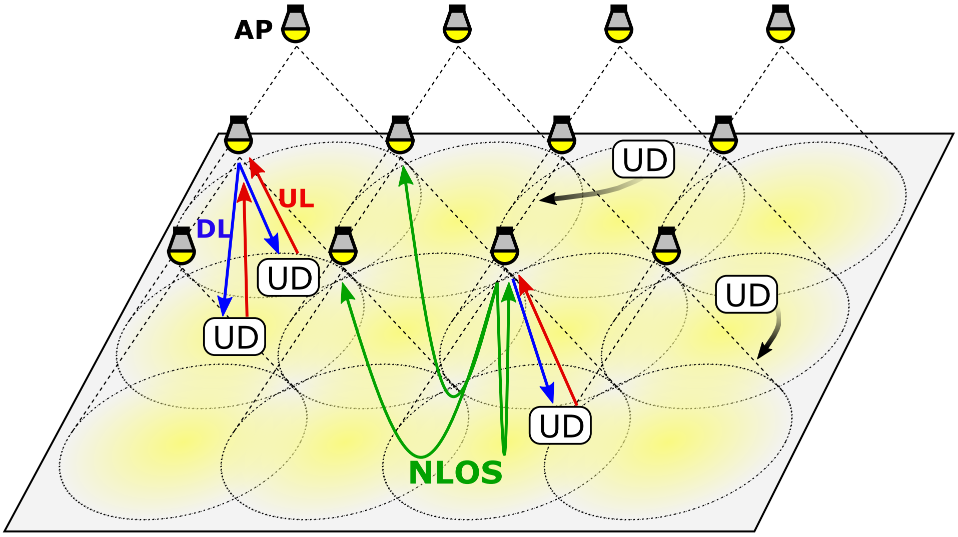


Figure 3.1: System topology

application layer can be extended to provide audio messages on the UL. Calculations show (section 3.4.1.2) that the UL protocol can provide the data rate required for future UL audio transmission.

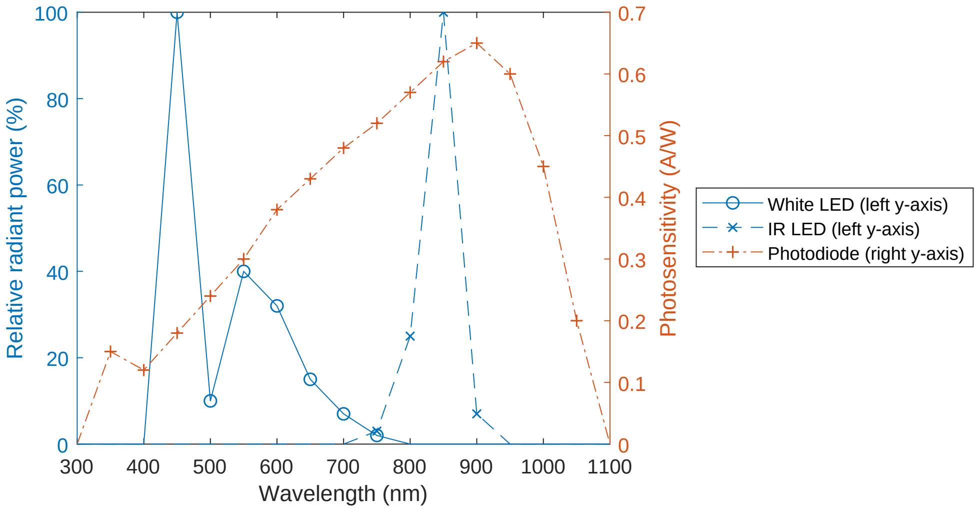


Figure 3.2: Relative radiant power spectrum of the white LED and IR LED on the left as well as the photosensitivity spectrum of the PD on the right [21, 68, 35]

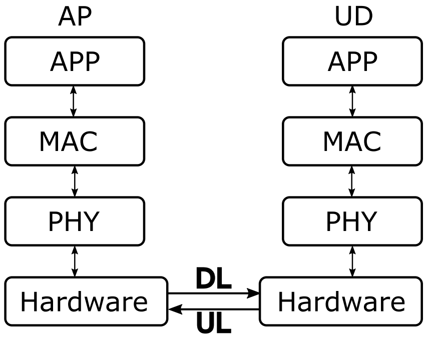


Figure 3.3: Protocol stack

3.1.3 Non-Line-Of-Sight

The same model of PD is used for both the APs and the UDs. Figure 3.2 shows the sensitivity of the PD over the light spectrum together with the relative optical output power of the white LED (AP) and IR LED (UD). The luminous flux of the white LED is high. The floor or other objects in the room reflect the visible light emitted by the white LED. This results in an NLOS link between each AP and its neighbouring APs (green arrow in figure 3.1) since the PD at each AP is sensitive to visible light. In section 5.1.2 we measure the received signal strength of the NLOS signal.

The NLOS signal offers both opportunities and challenges. The inter-AP NLOS signal offers the opportunity for neighbouring APs to sense each other's signal. The inter-AP NLOS signal is used by the MAC protocol to achieve an efficient channel utilization. A challenge is the intra-AP NLOS signal (self-interference) which limits the ability to perform collision detection and hinders full duplex transmission.

A second type of NLOS signal is present in this topology which is not shown in figure 3.1. The signal transmitted by the IR transmitter at each UD is reflected by the ceiling and reaches the PDs of nearby UDs. However, due to the low radiant power of the IR LEDs, the received signal strength for this second type of NLOS signal is negligible.

3.1.4 Protocol Stack and Frame Contents

The protocol stack consists of only three layers: the physical (PHY), medium-access control (MAC) layer and application (APP) layer as shown in figure 3.3. No network layer is required since all communication occurs in a single hop. For simplicity, some of the functionality of the OSI-model transport layer [60, p.41-45] is implemented at the application layer. We will present each of the three layers in detail the following sections.

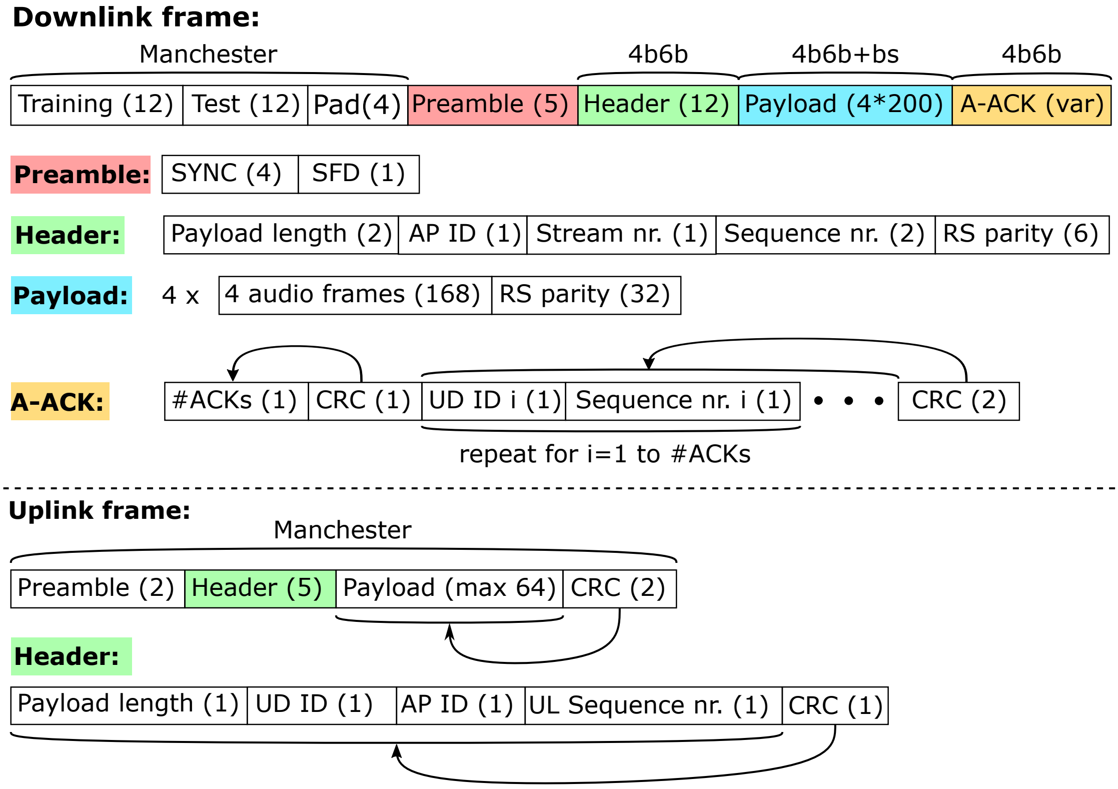


Figure 3.4: DL and UL frame contents and their length in bytes

The DL and UL frame contents are shown in figure 3.4. A DL frame starts with a training-test-pad (TTP) sequence which is used by the PHY layer's SIC and CD mechanisms (section 3.4.2 and 3.4.3.2). Next, it contains a short preamble for start of frame detection and synchronization (sections 3.4.3.4, 3.4.3.5). Next is the header which for simplicity is a unified header containing PHY, MAC and application related fields. The header is encoded with RS forward error correction (FEC) and thus contains RS parity bytes (RS presented in section 3.4.1.4). Next is the payload which contains four RS blocks. Each RS block carries four APP layer audio frames. An APP frame contains 20ms of compressed audio. To avoid confusion with the full DL and UL frames we hereafter refer to the APP layer audio frames (20ms) as 'mini-frames'. The number of RS blocks is variable but in the current implementation is set to four. Hence, each DL frame carries $4 \times 4 \times 20ms = 320ms$ of compressed audio. Finally, an aggregated acknowledgement (A-ACK) is appended to the end of the DL frame and contains acknowledgements for successful UL transmissions (section 3.3.3.2).

An UL frame is much shorter compared to a DL frame. The UL frame contains only a 2-byte-long preamble, a unified header with an 1-byte cyclic redundancy check (CRC) and a payload of up to 64 data bytes with a 2-byte payload CRC (CRC presented in section 3.4.1.5).

3.2 Application Layer

The application layer provides end-to-end audio transmission on the DL and text transmission on the UL. This section is divided into parts. The first describes the DL application layer at both the AP and UD while the second part describes the UL application layer at both the AP and the UD.

3.2.1 Downlink

The objective of the DL application layer is to transmit audio streams from the AP to users. An audio stream is defined as the transmission of chronological frames of the same audio track (audio file) as shown in figure 3.5. In the current implementation only one audio track per AP is supported. Future work can extend the functionality to allow multiple audio tracks per AP.

3.2.1.1 Flow/Congestion Control

Frames of an audio stream are transmitted using a flow control algorithm [60, p. 201]. The flow control algorithm limits the frame transmission rate of an audio stream at the AP. This contrasts with a mechanism where a frame is always sent to the MAC layer for transmission as soon as the previous frame finishes transmission.

The advantage of using a flow control mechanism is that it lowers the computational power and memory requirement of the receiver hardware. Flow control also limits the effective DL data rate of each AP and as a result helps to improve fairness. Hence, the flow control mechanism is also a congestion control mechanism [60, p. 398]. The disadvantage of a flow control mechanism is the greatly increased time to transmit a full audio track. Since DL frames are broadcast, Fa user can arrive at an AP while the AP is in the middle of transmitting an audio stream. This results in the user having to wait until the audio stream reaches the end of the audio track before useful audio can be received. Therefore, if an audio stream takes longer to transmit, the average user wait time increases. We solve this problem in the next section.

The flow control algorithm is implemented by a leaky-bucket algorithm [60, p. 407]. We refer to algorithm 4 in appendix A for pseudo code of the leaky-bucket algorithm implementation. Every time a frame is transmitted, the ‘leaky bucket’ which has a finite capacity gets filled by a fixed value. In addition, the bucket continuously leaks at a fixed rate. Once the bucket is full, the AP must wait until the bucket leaks enough of its capacity to prevent overflowing the bucket. There are three parameters that determine the operation of the leaky bucket: `bucketSize`, `IFS` (interframe space) and `flowRate`. The final values of these parameters used in the system can be found in appendix C. The `bucketSize` determines the amount of frames that can be transmitted in short, high-rate bursts. The `bucketSize` is therefore also equal to the required application layer buffer size at the receiver. The

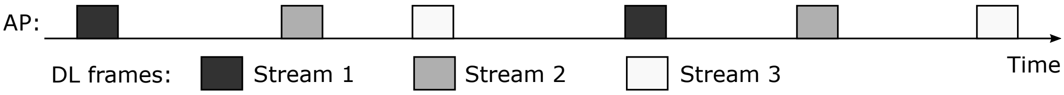


Figure 3.5: Timeline of AP transmitting three parallel audio streams

IFS is the minimum time between two successive DL frames. The IFS is required to allow UDs to process the received frame before arrival of the next frame. The flowRate is the constant rate at which the bucket is emptied. The flowRate is defined as a multiple of the audio playback rate at the RX. For example if flowRate=1 audio is transmitted at the same rate it is played back. In section 5.2.1.4 we study the effect of the value of flowRate. At the UD, audio starts being decoded and played back once the UD's buffer of received audio is at least 80% full. Audio stops playing when the buffer is empty.

3.2.1.2 Parallel Audio Streams

The solution to the increased user wait time caused by flow control is to give APs the ability to transmit multiple audio streams in parallel. Since each stream is transmitted at a lower rate due to flow control, multiple audio streams can be supported by the channel. The frames of the audio streams are interleaved in time. Figure 3.5 shows an example of an AP transmitting three audio streams. In section 5.2.1.3 we measure the average user wait time by simulation depending on the number of audio streams.

The flow control algorithm now has one leaky bucket for every audio stream. If multiple leaky buckets indicate that a frame can be transmitted, the application layer chooses to transmit a frame for the audio stream whose bucket is the emptiest. This effectively corresponds to earliest deadline first scheduling [74]. The deadline is the time point in the future at which the bucket would be empty if the AP stopped transmitting. If the UD does not receive a new frame from the stream before the deadline, audio playback at the UD is interrupted which should be avoided at all costs.

In the current implementation of LightTour the number of maximum parallel audio streams per AP is fixed. In future versions, a protocol can be derived which adaptively increases or decreases the number of parallel streams depending on the channel conditions. The advantage of such a protocol is the reduced user wait time.

3.2.1.3 Location Based Service

LightTour is a location-based service [74] which means that the audio track received by a UD is dependent on the location of the UD. The algorithm to determine which audio stream is received is similar to a technique used in cellular networks for handoff between cells [48]. At the UD, we define AP_{stream} as the AP from which the application layer at the UD is currently receiving an audio stream. The application layer can only receive one audio stream at a time. However, frame headers transmitted by multiple nearby APs still arrive at the UD. Every time a frame header from AP_{stream} is decoded, the application layer at the UD stores the RSSI of AP_{stream}: $RSSI_{stream}$. Every time a frame header from another AP, AP_{new}, is decoded, the application layer at the UD compares $RSSI_{stream}$ to $RSSI_{new}$. The application layer switches to AP_{new} if:

$$RSSI_{new} > \Theta_{loc} RSSI_{stream} \tag{3.1}$$

where Θ_{loc} is the location hysteresis threshold. Θ_{loc} is a fixed design parameter and on the hardware setup a value of $\Theta_{loc} = 2$ gives good performance. We refer to section 3.4.3.6 for a description of the way the value of RSSI is calculated.

3.2.1.4 Probing Frames

The application layer at the UD can determine which audio frames it wants to receive and decode using the method described in the previous section. When a DL frame header of AP_{stream} is decoded, a short CT sequence is transmitted on the UL channel. We describe the functionality of the CT sequence at the MAC layer in section 3.4.1.6 and its detection mechanism at the PHY layer in section 3.4.3.3 The CT sequence indicates to the AP that at least one UD wants the frame payload. An audio stream for which no CT sequences are received, transmits a 'probing' frame once every $t_{probe} = 1$ seconds. The probing frame consists of the header of the first frame of the stream. If a CT sequence is detected for this probing frame, the probing frame payload is transmitted, and the stream transmission is started.

The probing frame mechanism has three benefits compared to a system that is not able to detect users and as a result must continuously broadcasts all streams:

1. DL bandwidth reduced. Without user detection, the AP would transmit all parallel audio streams continuously. With user detection the AP only transmits the frame header of one audio stream every second. With three parallel streams per AP and using the final system parameters in appendix C this results in a reduction in the total transmission time of $\frac{3.125 frames/s \times 3 \times t_{dl} - 2ms}{3 \times t_{f,dl}} = 99.21\%$ where $t_{f,dl}$ is the DL frame length and each probing frame is $2ms$ long.

2. AP power consumption reduced. Authors in [64] show that the power consumption of the LED is higher when transmitting data than when the AP is in illumination mode. The TX front end in our hardware setup has a power consumption of $P_{w,ill} = 2.51W$ when idle (in illumination mode) and $P_{w,comm} = 3.04W$ when transmitting data. Our calculation above shows a reduction in transmission time of 99.21%. Consequently, the AP power consumption is reduced by $\frac{0.9921(P_{w,comm} - P_{w,ill})}{P_{w,comm}} = 17.3\%$.

3. User wait time reduced. When arriving at an AP, a user has to wait until an audio stream is available. The 'probing' mechanisms results in a wait time of at most t_{probe} seconds if an audio stream is available.

3.2.2 Uplink

The objective of the UL application layer is to transport text messages from the user to the AP. When necessary, a long text message is split into multiple UL frames. The UL application layer utilizes the UL MAC FIFO buffer explained in section 3.3.3.2. The FIFO buffer handles retransmissions and places the frames in the correct order at the AP receiver. For testing purposes, a text message consisting of a long string of text is transmitted by each UD to its AP_{stream}. The AP APP layer places the received frames in the correct order and sends the received text messages in over a Universal Serial Bus (USB) connection to a personal computer (PC) where the messages are printed in the correct order.

3.3 Medium Access Control Layer

The DL and UL MAC protocol control access to the DL channel and UL channel respectively. This section starts with a high-level overview of the operating principles of the two protocols. Next, we present a detailed description of each protocol.

3.3.1 Overview

The design goals of the DL and UL MAC protocols are:

- DL
  - Support for a high number of users: reliable and timely audio transmission should be guaranteed at high loads (many users).
  - Support for soft-deadline data such as audio
  - Fairness: no AP should be able to occupy a large portion of the channel.
- UL
  - Minimize overhead on DL throughput.
  - Reliable delivery of text data
- DL & UL
  - High bandwidth efficiency
  - High energy efficiency
  - Low complexity: the protocols need to function on a simple microcontroller without wired or RF connection between APs or UDs.

To achieve these requirements, random access protocols are designed for both the DL and the UL. We adapt CSMA/CD to VLC for the DL MAC protocol [60, p.268,285] and refer to this protocol as VLC-CSMA/CD. For the UL MAC protocol we adapt BTMA to VLC and we call the new protocol busy slot multiple access (BSMA) [12]. All DL audio transmissions are broadcast since a broadcast protocol

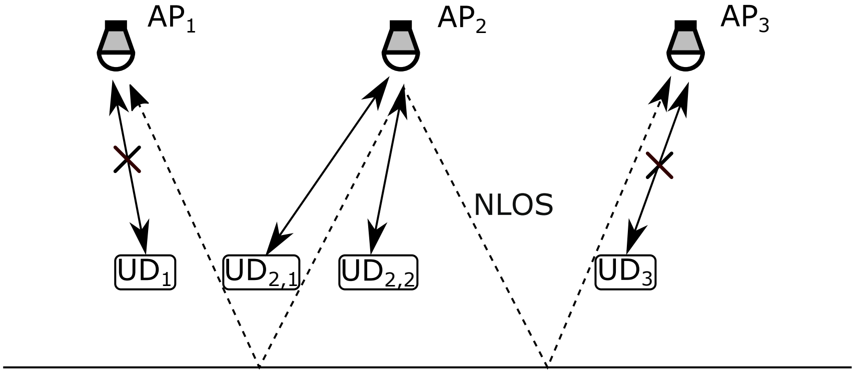


Figure 3.6: Example setup with three APs where AP2 is transmitting

shows greatly improved performance under high loads (section 5.2.1.4). Fairness and support for soft-deadline transmission on the DL is achieved using a leaky-bucket algorithm at the application layer which limits the data rate at each AP to provide both flow control and congestion control (section 3.2.1.1)[60, p.201, 392]. The UL protocol is designed to minimize overhead on the DL channel under high loads. For this reason, only full-duplex transmissions are allowed on the UL channel, i.e. UL transmissions can only occur during DL transmission. The full-duplex channel is provided by SIC (section 3.4.2). In section 5.2.2 we show that BSMA achieves the requirement of high UL bandwidth efficiency.

The principle of operation is presented based on an example setup shown in figure 3.6. In the figure AP$_2$ is broadcasting a downlink frame. UE$_1$ is located at AP$_1$, UE$_{2,1}$ and UD$_{2,2}$ are located at AP$_2$ and UD$_3$ at AP$_3$. Communication between AP and UD always happens with the nearest AP. The NLOS link allows APs to detect ongoing transmissions of neighbouring APs. The DL VLC-CSMA/CD protocol guarantees neighbouring APs are not able to simultaneously transmit a DL frame. As a result, since only full duplex UL transmissions are allowed, UDs at neighbouring APs can never transmit at the same time. Therefore, AP$_1$, UD$_1$, AP$_3$ and UD$_3$ are blocked from transmitting in this figure. As a result, the UL protocol operates on a network that can be modelled as a star topology where the AP is a central node and its UDs are the spokes (3.9).

Figure 3.7 shows an example communication timeline for all nodes in figure 3.6. For simplicity, we assume each AP can detect the transmissions of the other two APs. This is the case in figure 3.6 if the APs are arranged in a triangle formation and each AP can detect its neighbours. The timeline starts with AP$_1$ transmitting a DL frame. While AP$_1$ is transmitting, AP$_2$ and AP$_3$ sense the DL channel is busy and defer their transmissions to a later time. Consequently, UD$_{2,1}$, UD$_{2,2}$ and UD$_3$ are blocked from transmitting on the UL. UD$_1$ transmits a full-duplex CT sequence when it decodes the header of the DL frame. The CT sequence allows AP$_1$ to detect if there are users nearby who want to receive audio. AP$_1$ detects the CT sequence

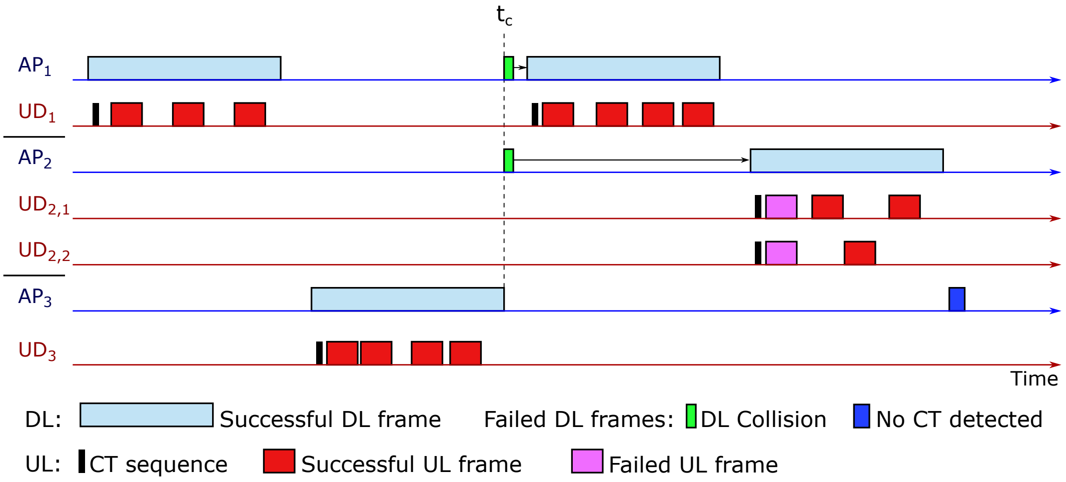


Figure 3.7: Example timeline of the setup in figure 3.6. Each AP is assumed to be able to detect the two other APs.

and transmits the full DL frame. During the DL payload, UD$_1$ transmits three UL frames using the random access BSMA protocol. When AP$_3$ senses a clear channel, it transmits a frame to the UDs in its service range (UD$_3$). At time t_c, AP$_1$ and AP$_2$ start transmitting simultaneously and detect a collision. After a random backoff AP$_1$ can transmit, followed by AP$_2$. AP$_2$ broadcasts a frame to both UD$_{2,1}$ and UD$_{2,2}$ and AP$_2$ detects the overlapping CT sequences transmitted by the two UDs. UD$_{2,1}$ and UD$_{2,2}$ use the random access BSMA protocol to transmit during the DL frame. The first two frames are lost due to collision. Finally, AP$_3$ transmits the start of a DL frame but fails to detect a CT sequence. The reason for this is that the application layer at UD$_3$ has decided, after decoding the header, it does not want the audio payload and no CT sequence is transmitted on the UL. The decision process by the application layer at the UD is presented in 3.2.1.3.

3.3.2 Downlink

The algorithm for the DL MAC protocol is shown in figure 3.8. The algorithm operates in a similar way to the ethernet CSMA/CD protocol [60, p.268,285]. The differences between VLC-CSMA/CD and ethernet CSMA/CD are the CD implementation at the PHY layer, the CT sequence added by our protocol and the nonpersistent backoff strategy instead of 1-persistent for ethernet. In section 5.2.1.1 we test several backoff strategies. Best performance is obtained with a nonpersistent strategy.

The algorithm starts when the MAC layer receives a transmission request from the application layer. The first step is MAC frame assembly. Assembly consists of calculating the header fields, line code encoding of the payload data and storing the raw encoded frame into a single data structure. Next, the AP initializes the variable for the number of backoffs (NB) and the variable for the backoff exponent (BE) to zero. The AP also stores the time, startT, of packet arrival from the application layer. The `micros()` function returns the time since the system startup in μs. Subsequently, a nonpersistent sensing of the channel is performed. The CCA mechanism is explained in section 3.4.3.1. If CCA indicates the channel is clear the AP transmits a random Manchester TTP sequence. The TTP sequence serves two purposes. Firstly, it allows the AP to estimate the channel response of the self-interference signal which are used by the SIC mechanism. Secondly, TTP allows the AP to perform CD. The SIC and CD mechanisms are provided by the PHY layer and explained in section 3.4.2 and 3.4.3.2.

If a collision is detected, the program checks if the time since startT is greater than the maximum allowed time, `macMaxT`, for DL transmission attempts. If this is the case, the transmission is aborted and the application layer is notified. Otherwise, NB and BE are updated using a binary exponential backoff algorithm [60, p.285]) and the AP waits a random period between 0 and $2^{\text{BE}} - 1$ unit backoff periods (UBP). The UBP is $t_{ubp,dl} = 2 \times t_{v,dl}$ which is the unit time period used in a CSMA/CD protocol [60, p.285]. The time $t_{v,dl}$ denotes the DL vulnerable period which is determined in section 3.4.3.1. The optimal values for `macMaxBE` and `maxMaxNB` are found experimentally in 5.2.1.1 .

If no collision is detected when transmitting the TTP sequence, the AP continues transmission of the frame header. All UDs in range detect and decode the frame header. Using the header, the application layer at each UD decides whether it wants to receive the corresponding payload (section 3.2.1.3). If the decision is positive, the PHY layer at the UD immediately transmits a full duplex CT sequence (section 3.4.1.6) on the UL channel. Otherwise, the transmission is ignored by the UD. When the AP detects a CT sequence before the CT deadline (t_{ct}), it continues transmitting the frame payload. Otherwise, the AP aborts the frame transmission and notifies the application layer.

Solving the Downlink Hidden Node Problem

The downside of a CSMA/CD protocol is that it does not solve the hidden node problem [60, p. 278]. The hidden node problem on the DL exists when APs have overlapping service range but are not able to sense each other's transmissions. However, by proper placement of APs the hidden node problem can be avoided. In a rectangular grid of AP (figure 3.1) with inter-AP distance d_{hor}, DL horizontal transmission range d_{tx} and horizontal inter-AP sensing range d_{sense} two conditions must be satisfied for correct operation of the system:

$$d_{tx} \geq \frac{\sqrt{2}}{2} d_{hor} \tag{3.2}$$

$$d_{sense} \geq \sqrt{2} d_{hor} \quad \text{if } d_{tx} = \frac{\sqrt{2}}{2} d_{hor}. \tag{3.3}$$

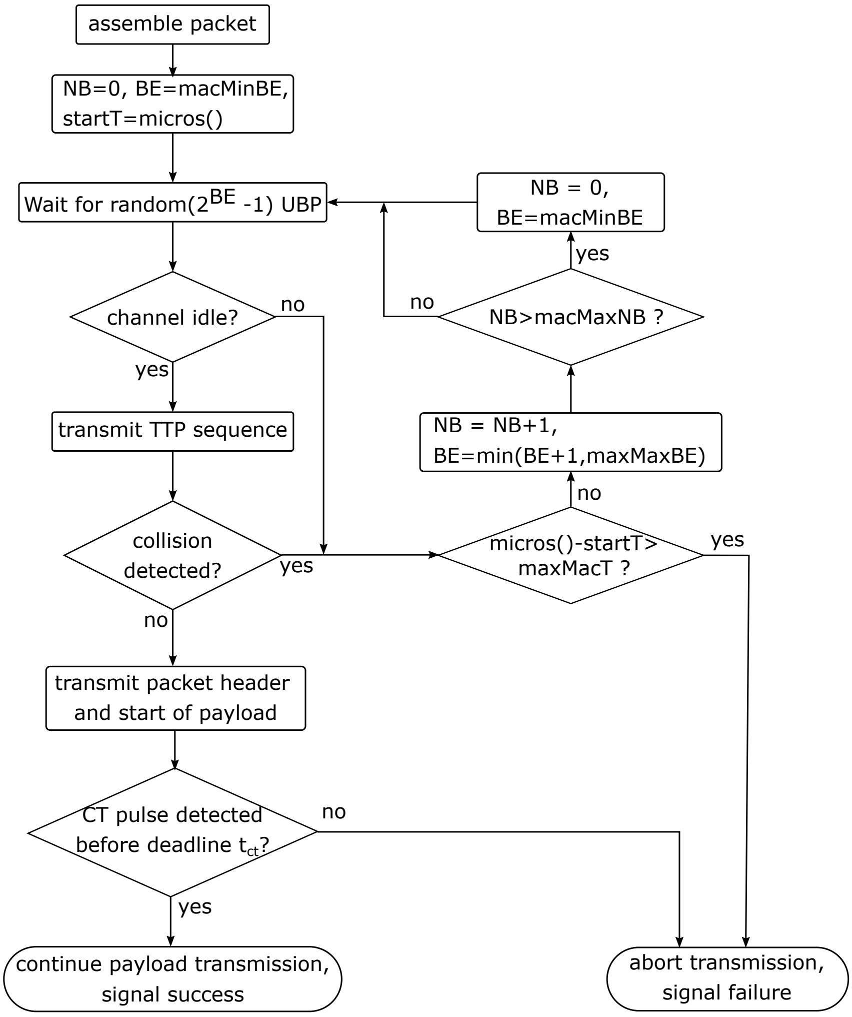

assemble packet
NB=0, BE=macMinBE, startT=micros()
Wait for random(2^BE -1) UBP
NB = 0, BE=macMinBE
yes
channel idle?
no
no
NB>macMaxNB ?
yes
transmit TTP sequence
NB = NB+1, BE=min(BE+1,maxMaxBE)
no
collision detected?
yes
micros()-startT> maxMacT ?
yes
no
transmit packet header and start of payload
CT pulse detected before deadline t_ct?
no
yes
continue payload transmission, signal success
abort transmission, signal failure

Figure 3.8: DL MAC protocol

The first condition ensures that the complete area of the room is serviceable by DL transmissions. The second condition eliminates the hidden node problem as it ensures that all APs that have overlapping service range are able to sense each other's signal. On the hardware setup measurements in section 5.1.2 we show that our implementation satisfies the condition for avoiding the hidden node problem.

Achieving High Reliability without ACK Frames

Unlike IEEE 802.15.7 [3], our DL protocol does not incorporate ACK frames, transmitted by UDs, confirming correct reception of a DL frame. Since our protocol uses broadcast transmissions, ACK frames would significantly increase protocol complexity. In a broadcast protocol with ACK frames, if at least one ACK is not received, a broadcast retransmission is necessary. This can significantly decrease overall throughput if one of the UDs is at a location at which the received SNR is low and thus the frame error rate (FER) is high.

LightTour uses two mechanisms to achieve reliable audio transmissions without ACK frames. The first mechanism is CD. In CSMA/CD, a frame is retransmitted if a collision is detected. If the CD mechanism has perfect classification accuracy, CD ensures that the frame will eventually be transmitted without collision.

Bit errors at the UD's RX can still result in part of the frame being lost. However, audio transmissions do not require a FER of 0% as the audio codec, Speex, is robust to some loss of its mini-frames ($20ms$ audio frames) [80]. The second mechanism to increase reliability is RS FEC which decreases the BER (and thus also the FER) for the same SNR [22]. The RS codes used are presented in section 3.4.1.4.

3.3.3 Uplink

The UL MAC protocol controls access to the UL channel. The UL channel is used to transmit text feedback messages from the UD to the AP. In future versions the functionality can be extended to short voice feedback messages. Below we present the UL MAC algorithm.

3.3.3.1 Star Topology

Only full duplex UL transmissions are allowed. This serves two purposes. Firstly, full duplex UL communication does not compete for channel resources with DL traffic. Hence, delivery of audio on the DL is not affected when a large number of UDs transmit feedback data on the UL. Secondly, because the DL protocol guarantees no two APs can transmit a DL frame simultaneously, full duplex communication ensures UL transmissions of UDs located at different neighbouring APs cannot collide. As a result, neighbouring APs can be ignored by the UL protocol.

The transmission power of the IR LEDs at the UDs is much lower than the white LEDs at the APs. On the DL APs can sense transmissions of neighbouring APs via the NLOS link. At the UD on the other hand, the low signal strength eliminates the possibility of detecting neighbouring UDs. As a result, UDs are hidden from each other [60, p. 278]. The network model for the UL is a star topology where the AP is the central node and the UDs in the AP's service range are edge nodes that are hidden from each other (figure 3.9).

3.3.3.2 Aggregated Acknowledgement and MAC Queue

The UL MAC layer at each UD keeps a first-in first-out (FIFO) queue Q_{ul} of pending transmissions. When no DL frames are being transmitted, the UL application layer can request to place a message into Q_{ul} for transmission during the next DL frame. When the UL MAC layer receives a frame from the application layer, the frame is

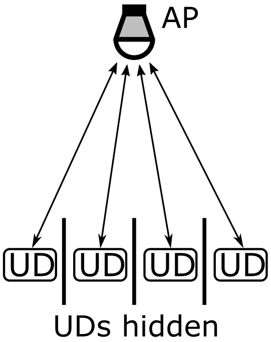


Figure 3.9: Uplink MAC protocol topology (star)

assembled and placed into Q_{ul}.

Multiple UL frames may be transmitted by the same UD during one DL frame. Likewise, multiple UDs may transmit during the same DL frame. All UL frames that are successfully received by the AP are acknowledged using an aggregated acknowledgement (A-ACK) appended by the AP to the end of the DL frame, as shown in figure 3.4. After the UD decodes the A-ACK, Q_{ul} is updated. All acknowledged frames are removed from Q_{ul}. The unacknowledged frames remain in Q_{ul} to attempt transmission during the next DL frame.

A similar queue is kept at the AP UL MAC layer. The AP buffers all frames received from the UD and delivers them in the correct order to the application layer. The sequence number in the header of the UL frame is used to determine the frame order.

The combination of A-ACK, Q_{ul} and UL frame retransmissions provides a reliable delivery of UL data. Additionally, it allows messages longer than the maximum UL frame length to be transmitted.

3.3.3.3 Busy Slot Multiple Access

Our solution to the hidden UD problem [60, p. 278] in the UL topology (figure 3.9) is a novel MAC protocol based on busy tone multiple access (BTMA) [12]. BTMA is a protocol intended for a wireless medium. BTMA relies on an out of band signal (i.e. the busy tone) that is transmitted continuously in a dedicated frequency band by a node that is receiving data. The busy tone indicates to other nodes that the channel is busy. We adapt the BTMA algorithm to VLC. In order to maintain the simple transmitter and RX hardware we don't implement a busy tone on a separate frequency band. Instead, the busy tone is replaced by dedicated 'busy slots' (BS) periodically interleaved in time with the DL frame payload as shown in figure 3.11. The BSes serve the same purpose as the busy tone in BTMA. We refer to our new protocol as BSMA. Interleaving and updating of BSes in the payload is performed at the PHY layer. We explain the BS mechanism at the PHY level in more detail in section 3.4.1.3.

3.3.3.4 Protocol

The UL MAC protocol (BSMA) controls the transmission of UL frames in Q_{ul} at the UD. The UL MAC protocol can be divided into two parts. The first part operates during DL payload transmission and is shown in figure 3.10. The UL BE at the UD is fixed for the full duration of the DL frame. The second part (algorithm 1) is executed after the UD decodes the A-ACK at the end of the DL frame and updates the UL BE. The BE update algorithm uses knowledge of the number of successful and unsuccessful UL transmissions by the UD during the DL frame.

At the start of the DL frame payload, the first step is a check if Q_{ul} contains any UL frames. In the next step the algorithm waits for the remaining wait time TW. Next, the algorithm waits until the channel is idle and a p-persistent backoff strategy is performed [60, p.267]. If, during the p-persistent backoff, the channel is sensed to be busy, the algorithm assigns a new random value between 0 and $2^{BE}-1$ to TW and goes back to the wait step. Otherwise, the next frame in Q_{ul} is transmitted. When transmission of the UL frame is finished, the Q_{ul} index is incremented and TW is set to a new random value between 0 and $2^{BE}-1$ and the entire process repeats.

The BE is updated after decoding of the A-ACK at the end of the DL frame using algorithm 1. The inputs to the algorithm are the current value of BE, the number of transmitted frames by the UD during the current DL frame (nbAttempts) and the number of UL frames of the UD that are acknowledged by the AP in the A-ACK. The new BE is a function of the percentage of successful UL frames of the UD during the current DL frame. No extensive study is performed on optimizing this function. In section 5.2.2 we measure the performance of the protocol and show that it can provide a throughput of 70% of the bandwidth under high loads and a fast convergence of data rate at each node to a stable equilibrium. In appendix B we derive the maximum overhead caused by UL transmissions on the DL channel which is found to be 7%. This satisfies the primary requirement of the UL MAC protocol.

3.4 Physical Layer

The PHY layer is the lowest layer of the protocol stack and its primary purpose is transmission of raw bits from one node to another. Transmission involves modulation, line coding, clock recovery, preamble detection, error detection and correction and SIC. Other services provided by the PHY layer are illumination, flicker mitigation, received signal strength indication (RSSI), BS interleaving, the CT sequence for UD detection, CCA and CD. This section is logically divided into three parts. The first part presents the PHY specification. The second part presents the SIC algorithm. The third part presents the remaining algorithms operating at the PHY layer.

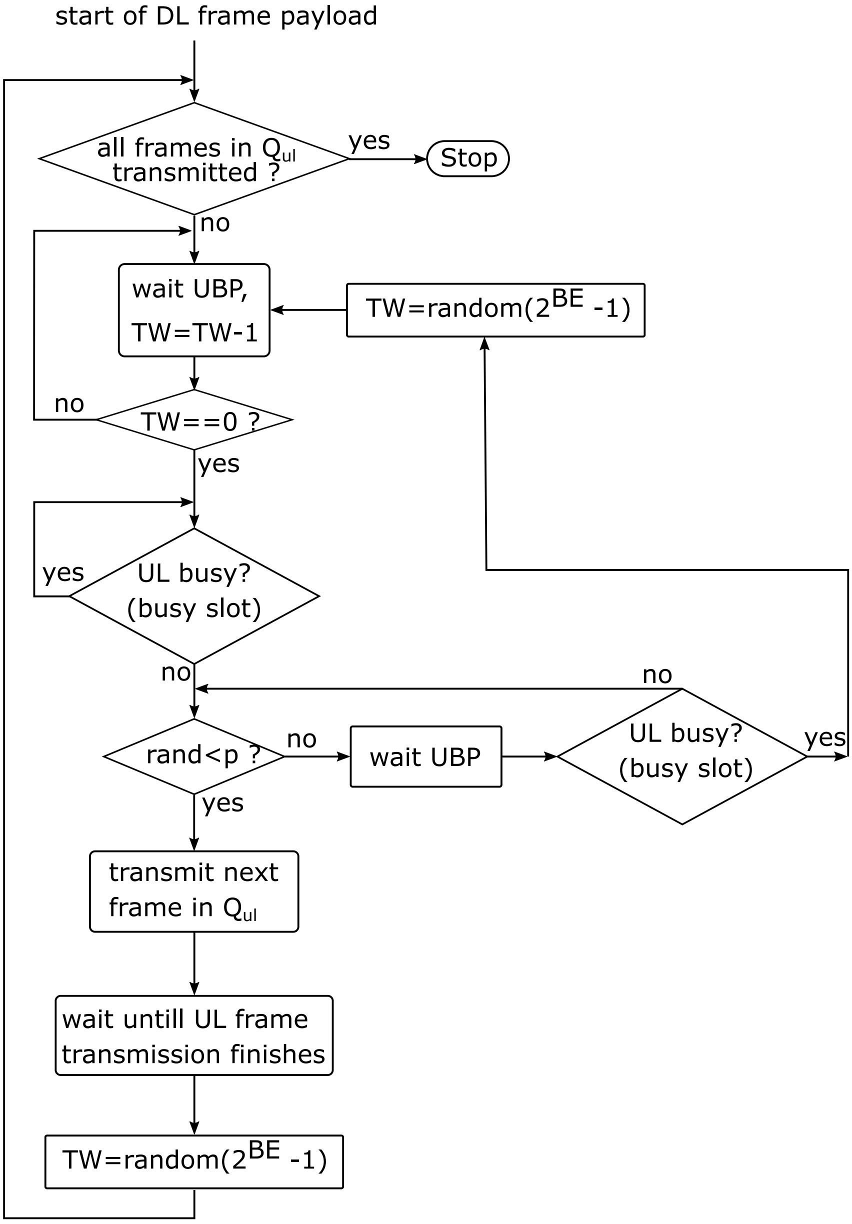


Figure 3.10: UL MAC protocol during DL frame transmission

Algorithm 1: UL BE update (after decoding of A-ACK)

```
Parameters: ulMacMaxBE, ulMacMinBE
Input     : BE, nbAttempts, nbACK
Result: BE
if nbAttempts != 0 then
    update=0;
    successPercentage=100*nbACK/nbAttempts;
    if successPercentage < 25 then
        update=3;
    else if successPercentage < 50 then
        update=2;
    else if successPercentage < 75 then
        update=1;
    else if successPercentage < 90 then
        update=-2;
    else
        update=-3;
    end
    BE=max(min(BE+update,ULmacMaxBE),ULmaxMinBE);
end
```

3.4.1 Specification

This section presents the specification of the PHY layer. The specification consists of the modulation, flicker mitigation, line code, BSes, CRC, FEC and PHY layer specific sequences like the preamble, TTP sequence and CT sequence. The LightTour PHY specification is built on a backbone that is very similar to the IEEE 802.15.7 standard. Some minor modifications are made to make the PHY specification more suitable to the hardware setup and to the upper layers. On top of this backbone, we extend the PHY specification with the TTP sequence, CT sequence and the BSes.

3.4.1.1 Illumination and Interframe Flicker Mitigation

The AP PHY controls both the illumination and modulation of the white LED. PHY provides interframe and intraframe flicker mitigation. At the UD, no illumination or flicker mitigation is needed since the human eye is not sensitive to IR light [31, p.46]. Chapter 4.1.3.1 describes how illumination and inter-frame flicker mitigation is supported by the AP front-end hardware.

3.4.1.2 Modulation, Line Code and Intraframe Flicker Mitigation

All signals in LightTour use OOK modulation. OOK allows for simple hardware TX and RX front-ends. At the transmitter, the signal is modulated using the USART hardware module on the Arduino Due (section 4.2.1).

As explained in section 2.1.5, OOK in VLC requires an RLL and DC-balanced line code in order to avoid intraframe flicker. In LightTour two line codes are used for different frame contents: 4b6b and Manchester. Table 3.1 and figure 3.4 show the DL and UL frame contents together with the line code used. The AP encodes its transmissions using 4b6b or 4b6b+bs (4b6b with interleaved busy symbols) except the TTP sequence which is encoded using Manchester. Manchester coding is used for the TTP sequence as this is required by the SIC mechanism and CD mechanism. The UD encodes all its transmissions using Manchester. This is because the AP must be able to decode the received data in real time to be able to encode the A-ACK before the end of the DL frame.

Even though flicker mitigation is not necessary for the UL channel, using an RLL and DC-balanced line code for the UL is still beneficial. The power spectrum of an RLL and DC-balanced code has zero power at DC and very low power at low frequencies. This property allows for a RX front-end with a bandpass filter. The bandpass filter attenuates the low frequency components of the received signal. Since most of the background noise in VLC is present at low frequencies as shown in figure 2.6, the received SNR is increased.

The blind oversampling method used in the clock recovery mechanism (section 3.4.3.4) requires that the downlink OCR $f_{o,dl}$ is (at most) half the ADC sampling rate $f_{adc} = 1MHz$. The SIC mechanism generates an output signal that at the DL OCR $f_{o,dl}$. As a result the sample rate of the full duplex UL channel is $f_{o,dl}$. The UL clock recovery uses the same mechanism as the DL clock recovery. Hence the UL OCR ($f_{o,ul}$) is half the DL OCR. In a future version of the system where the UD is synchronized with high accuracy to the DL transmission (e.g. using a hardware phase-locked loop (PLL)) and the phase difference between the UL and DL transmission can be controlled accurately, an UL OCR equal to the DL OCR of 500kHz is possible since no blind oversampling would then be necessary. As a result, the UL data rate can be doubled.

Section B.3 in appendix B contains a derivation of the DL and UL maximum PHY data rate in our implementation. The results are shown in figure 3.1. We also derive the maximum application layer UL data rate to a single AP: $R_{ul,ap} = 18kb/s$. We also find the maximum uplink data rate from a single UD: $R_{ul,ud} = 8kb/s$. In a future version, the UL application layer can be extended to provide support for short audio feedback messages. If we assume an UL audio bit rate of $8kb/s$ and each user generates spoken audio for on average 10% of the time each AP can on average support up to $\frac{R_{ulapp,ap}}{0.1 \times 8kb/s} = 12.4$ users. For a grid topology with $d_{hor} = 1.2m$ spacing between APs (figure 5.14), this corresponds to 8.7 users/m^2.

Since $R_{ulapp,ap}$ is more than what is required by the application layer, future work can extend the UL PHY specification with another line code that trades bandwidth

Spec	DL	UL
OCR	500kHz	250kHz
Line code	*Training and test sequence:* Manchester *Header and A-ACK:* 4b6b *Payload:* 4b6b+bs	Manchester
FEC	*Header:* RS(12,6) *Payload:* RS(200,168)	None
Data rate	210kb/s	100kb/s

Table 3.1: PHY specification

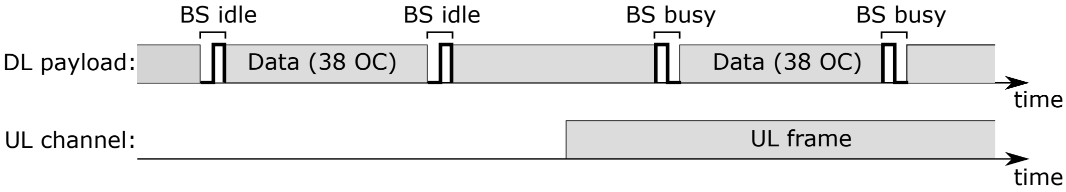


Figure 3.11: Busy slot (BS) mechanism, (OC = optical clocks)

efficiency for power efficiency. This way, the battery life of the UDs is increased. An example of such a line code is 4-PPM which has twice the power efficiency of Manchester [25].

3.4.1.3 Busy Slots

At the AP, BSes are interleaved with the frame payload at regular intervals of $l_{bsi} = 40$ symbols (or $t_{bsi} = \frac{l_{bsi}}{f_{o,dl}} = 80\mu s$) by the PHY layer. Each BS is two symbols long and contains a symbol indicating if the UL channel is busy. A 'HIGH-LOW' symbol is transmitted during the BS when the UL channel is busy and a 'LOW-HIGH' symbol when the UL channel is clear. The busy symbols are used in the BSMA UL MAC protocol explained in section 3.3.3.3.

The AP detects if the UL channel is busy by detecting UL preambles using the mechanism in section 3.4.3.5. Once a preamble is detected, the symbols in the next BSes are set to 'HIGH-LOW' indicating a busy UL. After decoding the 'payload length' field in the UL frame header, the AP sets the busy symbols after the UL frame back to 'LOW-HIGH'. Users perform CCA during the DL payload by decoding the last received busy symbol.

Compared to BTMA which uses a dedicated frequency band [12], the BS mechanism causes an additional inherent delay between the time of detection of the start of the UL frame and update of the BS. This delay is due to the fact that the interval spacing t_{bsi} of BSes is not zero. As a result, once the AP detects the start of an UL frame, it can take up to $t_{bsi} = 80\mu s$ before the next busy symbol is updated to indicate that the UL channel is busy. By contrast, the busy tone in BTMA is updated without delay once the UL frame is detected. As a result, in BSMA the MAC protocol vulnerable period is up to $t_{bsi} = 80\mu s$ larger compared to BTMA. In chapter 5.2.2 we study the effect of the increased vulnerable period on the performance of BSMA.

3.4.1.4 Reed-Solomon Forward Error Correction

Similarly to IEEE 802.15.7 [3], RS is used for FEC [22]. In LightTour, FEC is only used in the DL channel. The RS code we use has an 8-bit symbol length. RS codes can be classified by their block length $\ddot{n}$ and message length $\ddot{k}$ and are denoted by RS($\ddot{n}$,$\ddot{k}$).

Two separate RS codes are used for different parts of the DL frame. The first, RS(200,168), is used for the DL frame payload. This code can correct up to $(\ddot{n} - \ddot{k})/2 = 16$ erronous bytes. RS(200,168) has a code rate of $\frac{\ddot{k}}{\ddot{n}} = \frac{168}{200} = 0.84$ which is similar to the code rate of the PHY II operating mode of IEEE 802.15.7 which is $\frac{128}{160} = 0.8$ [3]. The value k=168 is chosen as this is a whole multiple of the number of bytes in one mini-frame which is 42 bytes and carries 20ms of audio compressed audio. Making $\ddot{k}$ a whole multiple of the application data lowers implementation complexity.

The second RS code, RS(12,6), is used exclusively for the DL frame header. By coding the header using a separate and short RS code, the RX can start decoding the header as soon as the 12 header bytes are received. This results in the header being decoded much earlier than if a long RS code were used for both header and payload. As a result, the UD can respond quickly with an UL CT sequence. Consequently, the deadline for UL CT detection at the AP (t_{ct}) can be shifted closer to the start of the DL frame. This results in less wasted bandwidth when no CT is detected. The UL CT mechanism is explained further in section 3.4.1.6.

3.4.1.5 Cyclic Redundancy Check

Compared to RS codes, CRC codes [22] have two important differences. Firstly, the CPU time required for encoding and decoding of CRC codes is much shorter than RS codes (figure 5.2). This is important since RS decoding in real time (at the rate of transmission) is not possible on the Arduino Due (figure 5.2). As a result, during decoding, buffering of received samples is necessary. The decoding will finish after some time delay after the frame is received. The second difference is that CRC codes can only detect errors and not correct them. Hence, using CRC codes results in higher FER (at the same SNR).

For the above two reasons, CRC codes are used where decoding is time-critical and/or where data loss is not unacceptable. Firstly, the CRC code is used for UL frames. UL frames need to be decoded fast since the AP must finish decoding all UL frames before the end of the DL frame payload in order to assemble and append the A-ACK (section 3.3.3.2). For the same reason, the A-ACK must be encoded fast. Therefore, each A-ACK is also encoded with a CRC. During the DL payload, central processing unit (CPU) time is also required for other operations such as SIC and UL preamble detection. This further limits the ability to implement a high time-complexity code such as RS for UL frames. As a final note, UL frames have an automatic retransmission mechanism (section 3.3.3.2) which solves the data loss problem.

3.4.1.6 Other Physical Layer Specific Sequences

We briefly present three other PHY layer specific sequences. All three sequences are used by one or more algorithms described in section 3.4.3.

Continue Transmission When the UD decodes a DL header, the application layer at the UD indicates if it wants to receive the corresponding payload. When the payload is desired by the application, the PHY layer immediately transmits a CT sequence on the UL channel. The CT sequence consists of four random Manchester-encoded bytes. The AP continues transmitting the payload until the CT pulse deadline t_{ct}. If no CT pulses are detected by the AP before the deadline, the AP stops transmitting the payload. The CT sequence principle of operation is shown in figure 3.7. In section 3.4.3.3 we discuss the CT sequence detection method at the AP.

Preamble Each DL and UL frame contains a preamble as shown in 3.4. The preamble is not encoded with a line code. The preamble allows the RX to identify the start of the frame. The preamble consists of two parts: SYNC and start frame delimiter (SFD). The SYNC sequence consists of an alternating pattern of high and low bits. The SYNC sequence is used for clock recovery at the RX. The SFD sequence consists of a predefined, fixed pattern of bits. The SFD allows the receiver to pinpoint the start of the frame. We discuss clock recovery and preamble detection in section 3.4.3.5 and 3.4.3.4.

Training-Test-Pad The TTP sequence is prepended to the start of each DL frame and consists of random Manchester encoded data. The TTP sequence is used to determine the channel response for the SIC mechanism (section 3.4.2) and to perform CD (section 3.4.3.2).

3.4.2 Self-interference Cancellation

In this section we design a low-complexity digital SIC mechanism. In figure 3.1 the self-interference is the NLOS arrow emerging from and arriving at the same AP. Self-interference cancellation enables full-duplex communication and CD using a receiver PD at the AP that is sensitive to both visible (DL) and IR light (UL). We first describe a simplified SIC algorithm which ignores the distortion introduced by the receiver front-end band pass filter. Next, the full SIC algorithm is presented.

3.4.2.1 Simplified Algorithm

The system is shown in figure 2.3. The approximate system response as derived in section 2.1.3 is:

$$y[k] = h_r[k] \otimes (h_c(0)x[k]) + n[k] + V \tag{3.4}$$

In the simplified SIC algorithm we ignore the band-pass filter $h_r[k]$ at the receiver. As a result, the system model in the above equation is further simplified to:

$$y[k] = H \cdot x[k] + V + n[k] \tag{3.5}$$

where H is the overall DC gain of the complete system. At the start of each DL frame a short random training sequence is transmitted. The training sequence consists of 12 random Manchester encoded bytes. The training sequence is random since multiple nearby APs can start transmitting at the same time. The sum of the transmissions is detected by the receiver. The cross correlation of random Manchester sequences is zero [53]. As a result, the training signal transmitted by another AP can be modelled as uncorrelated noise. We group the extra uncorrelated noise term with $n[k]$ in equation B.5.

The goal is to estimate H and V at the AP from the known transmitted training signal $x[k]$ and resulting received self-interference signal $y[k]$. Let N_{tr} denote the length of the training signal in number of symbols. Signal $x[k]$ has two possible values: -1 and 1. Because of the Manchester encoding of $x[k]$ the number of times $x[k] = -1$ is equal to the number of times $x[k] = 1$. The system model in equation B.5 is a simple linear regression with an additive white Gaussian noise (AWGN) term. As a result H and V are found from the following minimization problem which minimizes the sum of squared noise terms [78]:

$$\text{Find } \min_{H,V} U(H,V), \quad \text{for } U(H,V) = \sum_{k=1}^{N_{tr}} n^2[k] = \sum_{k=1}^{N_{tr}} (y[k] - V - H \cdot x[k])^2. \tag{3.6}$$

The solution to this problem is derived in appendix B and is as follows:

$$\hat{H} = \frac{\bar{y}^p - \bar{y}^n}{2} \tag{3.7}$$

$$\hat{V} = \bar{y} = \frac{\bar{y}^p + \bar{y}^n}{2} \tag{3.8}$$

where $\bar{y}$, $\bar{y}^p$ and $\bar{y}^n$ denote the mean of $y[k]$, $y^p[k]$ and $y^n[k]$ in the training signal and

$$y^n[k] = \begin{cases} y[k] & \text{when } x[k] = -1 \\ 0 & \text{when } x[k] = 1 \end{cases}$$

and

$$y^p[k] = \begin{cases} y[k] & \text{when } x[k] = 1 \\ 0 & \text{when } x[k] = -1. \end{cases}$$

Which allows computing the estimate of $y[k]$:

$$\hat{y}[k] = \begin{cases} \bar{y}^p & \text{when } x[k] = 1 \\ \bar{y}^n & \text{when } x[k] = -1. \end{cases}$$

During the transmission of the frame payload the AP estimates the clean channel. This result is the residual self-interference after cancellation (RSIC):

$$\rho[k] = y[k] - \hat{y}[k]. \tag{3.9}$$

3.4.2.2 Full Algorithm

The full SIC algorithm includes the receiver bandpass filter $h_[k]$ and thus the system model in equation 3.4 is used. Now the AP has to estimate $h_r[k]$, H and V. In the literature the filter $h_r[k]$ is usually estimated by solving a system of linear equations using a least squares method or adaptively using a recursive algorithm such as the LMS algorithm [51]. The LMS approach requires 2N multiplications and 2N additions per iteration of the least mean square (LMS) algorithm where N is the length of the filter $h_r[k]$. If the FIR filter is not updated but only its output is calculated, N additions and N multiplications are needed per sample.

During DL payload transmission, the AP detects the presence of UL transmissions using preamble detection (section 3.4.3.5). A real-time detection of the UL preamble is necessary in order to update the symbol in the BS (see section 3.4.1.3). Therefore, it is paramount that $\rho[k]$ is computed in real time. Experimentally, it is found that an FIR filter of at least 11-taps is needed to get good SIC performance in 4b6b (section 5.1.3.1). In section 5.1.1 we measure the execution time of the LMS algorithm and FIR filter on the Arduino. For an 11-tap filter, LMS and FIR require respectively $3.5\mu s$ and $1.5\mu s$ per sample (figure 5.1). The SIC function has some extra overheads. The time to calculate SIC using and FIR filter including overheads is $2.2\mu s$ per sample. The DL OCR is $f_{o,dl} = 500kHz$. Therefore, if each bit is to be processed in real time, there is $2\mu s$ of time available to process each bit. This means, the SIC mechanism with FIR filter alone is too slow to be computed in real time. In addition, other operations like CD, UL CT sequence detection, UL preamble detection and UL line code decoding also require CPU time during the DL payload.

A lower time-complexity SIC mechanism is needed. The solution is using a LUT to evaluate the filter. The equation to compute the estimate of the output signal now becomes:

$$\hat{y}[k] = \Lambda_\iota \cdot \hat{H} \cdot x[k] + \hat{V}. \tag{3.10}$$

where Λ is a LUT consisting of 2^Γ elements and Γ denotes the LUT order. The index i denotes the index in the LUT. The scalar Λ_ι replaces the vector $h_r[k]$ in equation

B.5. The index ι is determined by keeping track of the Γ previously transmitted bits in $x[k]$ using the following formula:

$$\iota[k] = \sum_{j=0}^{\Gamma-1} (\frac{(x[k-j]+1)}{2} \cdot 2^j) \tag{3.11}$$

which can be implemented more efficiently using logical shift operations:

$$\iota[k] = ((\iota[k-1] << 1) \mid x_{01}[k]) \&\& 2048 \tag{3.12}$$

where $x_{01}[k]$ denotes a representation of $x[k]$ with values of 0 and 1 instead of -1 and 1 for the LOW and HIGH symbols (note: $2048 = 2^\Gamma$ and $\Gamma = 11$).

The system path loss H and the DC offset V are estimated in the same way as in the simple SIC algorithm (equation B.12 and B.13). They are both estimated at the beginning of each DL frame by transmitting a training sequence of 12 Manchester encoded bytes. Since H is re-estimated for each DL frame the full SIC algorithm can handle changes in channel conditions. The receiver filter $h_r[k]$ and the LUT are only dependent on the receiver circuit. We assume that the properties of the receiver circuit are constant in time. Therefore $h_r[k]$ and Λ need only be estimated once. Factors that can influence the receiver filter are temperature and ageing of the filter circuit or nearby radiation sources.

We estimate the LUT in Matlab using a large training set of transmitted data $x[k]$ and received data $y[k]$ collected on the hardware setup. The training set is collected by transmitting random Manchester and 4b6b encoded data. The AP is pointed downwards at a height of 0.65m from the floor. This distance is chosen as it increases the SNR in the received signal compared to the normal distance of 1.7m and thus improves training. The training set is partitioned into 2^Γ subsets x_ι and y_ι. The subset x_ι contains all $x[k]$ where ι is equal to equation 3.11. Each element in x_ι is thus preceded by the same Γ transmitted bits. LUT element ι, Λ_ι, is estimated by minimizing the squared error:

$$\Lambda_\iota = \underset{\Lambda_\iota}{\operatorname{argmin}} U_\iota = \underset{\Lambda_\iota}{\operatorname{argmin}} \sum_{j=1}^{M_\iota} (y_{\iota,j} - L_i \hat{H} x_{\iota,j} - \hat{V})^2 \tag{3.13}$$

where j is the sample index in set ι and M_ι is the number of samples in set i. For each ι, all $x_{\iota,j}$ are either -1 or 1. We derive the solution for $x_{\iota,j} = 1$. To minimize U_ι w.r.t. Λ_ι we calculate the least squares solution. We set the partial derivative of U_ι

w.r.t. L_ι to zero and solve for Λ_ι:

$$\begin{aligned}\frac{\partial \Lambda_\iota}{\partial U} &= \sum_{j=1}^{M_\iota}(2\hat{H}(y_{\iota,j} - \Lambda_\iota \hat{H} + \hat{V})) = 0\\ \Leftrightarrow & \sum_{j=1}^{M_\iota}((y_{\iota,j} - \Lambda_\iota \hat{H} - \hat{V})) = 0\\ \Leftrightarrow & \sum_{j=1}^{M_\iota} y_{\iota,j} = \sum_{j=1}^{M_\iota}(\Lambda_\iota \hat{H} + \hat{V}) = M_\iota(\Lambda_\iota \hat{H} + \hat{V})\\ \Leftrightarrow & \Lambda_\iota = \frac{\bar{y}_\iota - \hat{V}}{\hat{H}}. \end{aligned} \tag{3.14}$$

$\hat{H}$ and $\hat{V}$ are estimated from the total training set using equation B.12 and B.13

To minimize the memory requirements of the LUT and time complexity of the SIC algorithm, the elements of Λ are converted into 8-bit fixed-point representation.

The full SIC algorithm on the Arduino consists of calculating the estimates $\hat{H}$ and $\hat{V}$ from the training data at the beginning of each frame. Next, equation 3.10 is used for each sample k to get the estimate $\hat{y}[k]$. For each sample the LUT index, $\iota[k]$, is updated using equation 3.11. The LUT is stored as a 2^Γ =2kB array in memory. Evaluating equation B.15 with 3.10 requires only 2 multiplications and 2 additions per sample. In addition, the update of the LUT index ι requires only 2 logical operations (equation 3.11). We measure the execution time and performance of the SIC mechanism in section 5.1.

3.4.3 Other Algorithms

After SIC we present the most important remaining algorithms used at the PHY layer. Each algorithm has the requirement that it should be very low complexity in order to operate in real time (process data at the same rate of transmission) on a microcontroller like the Arduino.

3.4.3.1 Clear Channel Assessment at the AP

CCA provides indicates the state (busy or clear) of the DL channel to the MAC layer. We implement CCA with the energy detection method because of its low complexity. CCA with energy detection is also defined in the IEEE 802.15.7 standard [3, p. 238]. The binary hypothesis test for CCA is formulated as [36]:

$$y[k] = \begin{cases} n[k], & H_0(\text{channel idle}) \\ s[k] + n[k] & H_1(\text{channel busy}) \end{cases} \tag{3.15}$$

where $s[k]$,$n[k]$ and $y[k]$ denote the detected signal from another AP, the noise and the received signal respectively. The test statistic is the variance of $y[k]$ over a window of the past N_{cca} samples:

$$\sigma_{cca}^2 = \mathrm{Var}\Big[y[k] \;\mid\; 0 \le k \le N_{cca}\Big] = \frac{1}{N_{cca}} \sum_{k=0}^{N_{cca}} \big(y[k] - \bar{y}\big)^2 \tag{3.16}$$

$$\bar{y} = \frac{1}{N_{cca}} \sum_{k=0}^{N_{cca}} y[k] \tag{3.17}$$

The AP decides if the channel is clear or busy by comparing σ_{cca}^2 with a threshold Θ_{cca}. The null hypothesis H_0 is rejected (channel sensed busy) if $\sigma_{cca}^2 > \Theta_{cca}$.

The window size N_{cca} affects the classification accuracy of CCA mechanisms based on energy detection [36]. Furthermore, the window size also affects the vulnerable period $t_{vul,dl}$ of the DL MAC protocol. We call this delay the DL vulnerable period $t_{vul,dl}$. The maximum value of $t_{vul,dl}$ in seconds is:

$$\max t_{vul,dl} = \frac{N_{cca} + N_{adc}}{f_{adc}} + t_{prop,link} \tag{3.18}$$

where $N_{adc} = 20$ is the ADC buffer size and $t_{prop,link} \approx 0$ is the delay caused by the link which is negligible compared to the other delays. The classification accuracy is measured on the hardware setup in section 5.1.5 and we determine a suitable value for N_{cca}.

3.4.3.2 Collision Detection

CD happens at the start of each DL frame during the test sequence which is the second part of the TTP sequence. Figure 3.12 depicts a timeline of the start of transmission of two DL frames by two APs. While transmitting the training sequence, the AP uses the received signal $y[k]$ to estimate the channel parameters used for SIC in equation $B.12$ and $B.13$. SIC is then applied on $y_{te}[k]$ which is the received signal during the transmission of the test sequence. The output of the SIC mechanism is the residual self-interference after cancellation (RSCI): $\rho_{te}[k]$.

The subsequent steps are similar to the CCA mechanism. The hypothesis problem for CD is:

$$\rho_{te}[k] = \begin{cases} n_{te}[k] & H_0(\text{no collision}) \\ s_{te}[k] + n_{te}[k] & H_1(\text{collision}) \end{cases} \tag{3.19}$$

where $n_{te}[k]$ is noise due to AWGN and due to imperfect cancellation of the self-interference and $s_{te}[k]$ is the received signal of interest transmitted by a nearby AP. The AP measures the variance σ_{cd}^2 in ρ_{te}:

$$\sigma_{cd}^2 = \frac{1}{N_{te}} \sum_{k=0}^{N_{te}} |\rho_{te}[k] - \overline{\rho}_{te}|^2 \tag{3.20}$$

$$\overline{\rho}_{te} = \frac{1}{N_{te}} \sum_{k=0}^{N_{te}} \rho_{te}[k]. \tag{3.21}$$

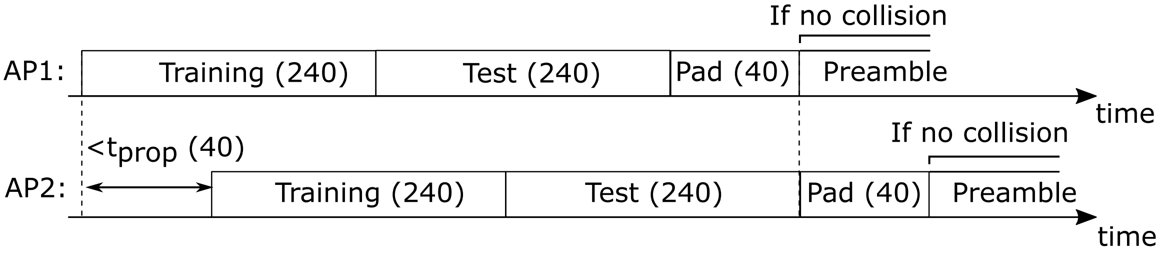


Figure 3.12: Timeline of start of frame transmission; the numbers (n) denote the length in number of symbols (optical clock periods).

The null hypothesis is rejected if $\sigma_{cd}^2 > \Theta_{cd}$ and a collision is reported to the MAC layer. Similarly to the CCA mechanism, the accuracy of CD is dependent on N_{te}. In chapter 5.1.5 we measure the accuracy of CD and find suitable values for N_{tr} and N_{te}.

If a collision occurs without a pad sequence, the vulnerable period $t_{v,dl}$ would result in part of the transmitted test sequences not overlapping. As a result, σ_{cd}^2 would be lower which has a negative impact on the classification accuracy. To solve this problem, a pad sequence with an equal length as $t_{v,dl}$ is used as shown in figure 3.12.

3.4.3.3 Continue Transmission Sequence Detection

Like CCA and CD, the CT sequence detector at the AP also uses energy detection. After transmission of the payload and before the CT detection deadline t_{ct} two measures of energy are calculated:

$$\sigma_h^2 = \mathrm{Var}\Big[\rho_h[k] \quad | \quad 0 \le k \le N_h\Big] \tag{3.22}$$

$$\sigma_p^2 = \max_j \mathrm{Var}\Big[\rho_p[k] \ | \ jN_B \le k \le (j+3)N_B\Big] \quad 0 \le j \le L_{ct} - 4. \tag{3.23}$$

where ρ_h and ρ_p are the RSIC signal during transmission of the header and payload respectively. $N_h = 180$ and $N_B = 10$ are the number of symbols in the DL header and in a single byte. $L_{ct} = 24$ is the number of payload bytes before the CT detection deadline t_{ct} which is $480\mu s$ after the end of the header transmission.

At time t_{ct} the two energies are compared. If $\sigma_p^2 > \Theta_{ct} \times \sigma_h^2$ a CT sequence is detected. The threshold factor $\Theta_{ct} = 2.5$ is found experimentally and provides a good balance between false positives and false negatives.

The reason why CT sequences are detected by measuring the received signal strength and not by looking for some predefined sequence such as a preamble is as follows. Since the DL protocol is broadcast, multiple UDs can transmit an UL CT sequence at the same time. The CT sequence transmissions of the UDs are not synchronized. As a result, the AP receives multiple, overlapping, time-shifted copies of the CT sequence. Hence, the received signal is garbled preventing the AP from decoding the CT sequences successfully. Instead, the AP measures the variance. To guarantee multiple CT sequences do not destructively interfere at the AP, each CT sequence is a random sequence of bytes. Random Manchester encoded signals are uncorrelated. As a result, the power of the sum of the CT sequences is the sum of the powers [53].

3.4.3.4 Clock Recovery

No hardware PLL is used for clock and data recovery. Instead we design and implement a novel low-complexity software-based solution based on blind oversampling of the received signal [37]. In blind oversampling the received signal is sampled using a free-running clock at some multiple of the OCR. This way the receiver obtains multiple phases of the received signal. The receiver then picks the best phase. The best phase is the phase which samples the data closer to the center of each bit.

At the start of the frame, the best phase is selected using the SYNC field of the preamble. During preamble detection, the following cross correlations are calculated:

$$\tau_i = y[Wk+i] \star x_{SYNC}[k], \quad \text{for } 0 \leq i \leq W, \quad 0 \leq k < N_{sync} \tag{3.24}$$

In the above formula, y is the received signal, x_{sync} is a polar transmission template (high=1, low=-1) of the SYNC sequence, N_{sync} is the length of x_{sync} in bits and, W is the oversampling rate, k is the sample index and i is the phase. The best phase i^* corresponds to:

$$i^* = \underset{i}{\operatorname{argmax}}(\tau_i). \tag{3.25}$$

The system uses an oversampling rate of $W = 2$. This is the lowest oversampling rate in a blind oversampling algorithm. A higher bit error rate (BER) can be obtained by using a higher W [37]. However, an increase in W either requires a receiver ADC with a higher sampling rate to achieve the same data rate or results in a lower data rate for the same ADC sample rate. Our measurements in 5.1.4 show that we can achieve low FER (2%) with $W = 2$.

The initial phase error of i^*, denoted by $\delta(i^*)$, is the time difference between the center of the bit and i^* after preamble clock recovery. Assuming the initial sample is always at least $0.1L_b$ removed from the center of a bit edge, the maximum initial phase error is:

$$\max \delta(i^*) = 0.4t_b \tag{3.26}$$

where t_b is the bit length in seconds. For the DL and UL the maximum initial phase error is:

$$\max \delta(i^*)_{dl} = 0.4/f_{o,dl} = 0.8\mu s; \quad \max \delta(i^*)_{ul} = 0.4/f_{o,ul} = 1.6\mu s. \tag{3.27}$$

Due to clock drift, i^* drifts away from the optimal sampling point when time increases. For this reason, a method is required to detect this drift during transmission and dynamically pick the best phase i^* based on the detected drift [13]. If two clocks

C1 and C2 with accuracy $Acc(C)$ have and initial error of $\max(\delta_{i^*})$ at time $t=0$, the maximum possible difference at time t is:

$$\max\delta(C(t)) = 2\times Acc(C)\times t + \max\delta(i^*) \tag{3.28}$$

Since, the Arduino hardware uses a crystal oscillator the frequency accuracy of the oscillator determines $Acc(C)$. The Arduino Due's oscillator is not found. However, its series number (12MHz KX-7) has models with accuracies from 10ppm to 50ppm (parts-per-million) [30]. We use the most likely value of 30ppm in our calculations. From equation 3.28 we obtain:

$$\max\delta C(t_{f,dl})_{dl} = 2\times 30\times 10^{-6}\times 25000\mu s + 0.8\mu s = 2.3\mu s = 1.15 t_{b,dl} \tag{3.29}$$

$$\max\delta C(t_{f,ul})_{max,ul} = 30\times 10^{-6}\times 3200\mu s + 1.6\mu s = 1.696\mu s = 0.425 t_{b,ul}. \tag{3.30}$$

where $t_{f,dl}[s]$ are $t_{f,ul}[s]$ the DL and UL frame lengths If we assume the sample point has to be at least $0.1T_b$ away from an edge, a value of $\delta C > 0.4L_b$ will lead to bit errors. As a result, continuous clock recovery (CCR) is necessary for the DL (i.e. at the receiver of the UD). For the UL we do not implement CCR since our calculations show that UL bit errors due to clock drift are unlikely (equation 3.30. Furthermore, the UL channel has an automatic frame retransmission mechanism which makes solves the problem of increased frame loss due to no UL CCR. FER measurements in section 5.1.4 show that an UL FER of 2% is achieved without CCR.

For the DL, we implement a novel low-complexity software CCR mechanism. As explained in section 4.2.1, USART is used for data modulation. Consequently, all transmissions have a USART start bit and a stop bit interleaved with every eight data bits as shown in figure 3.13. At the k'th transition between the stop bit and the start bit, the amplitude difference $A_\delta[k,i] = A_{stop}[k,i] - A_{start}[k,i]$ is calculated for the phases $i = i^* - 1$, $i = i^*$ and $i = i^* + 1$. An exponential moving average (EMA) and variance (EMV) of $A_\delta[k,i]$ are maintained for each phase i. Algorithm 2 shows the EMA and EMV recursive update step ([26]). α is a weight parameter, a higher α places more weight on more recent samples.

Next, algorithm 3 selects the new best phase i^* depending on the obtained EMA and EMV values. Θ_a and Θ_v are the hysteresis thresholds for the EMA and EMV respectively [73]. The algorithm compares the EMA and EMV of A_δ at i^*-1 and i^*+1 to the current best phase i^*. A phase is better if it is further from an edge. Hence, the phase for which EMA is greater is a better phase. Over time, the old best phase (i^*) drift towards the left or the right edge. Concurrently, either i^*-1 or i^*+1 drifts closer to the center of the bit (optimal sampling point). As a result, the EMA for the old phase (i^*) decreases and the EMA of the new phase (either i^*-1 or i^*+1) increases.

Likewise, the EMV of the optimal phase is lower due to two reasons. Firstly, the slope of the signal is steeper closer to an edge. Due to clock jitter, small variations on the sample time result in large variations on the amplitude (resulting in a greater

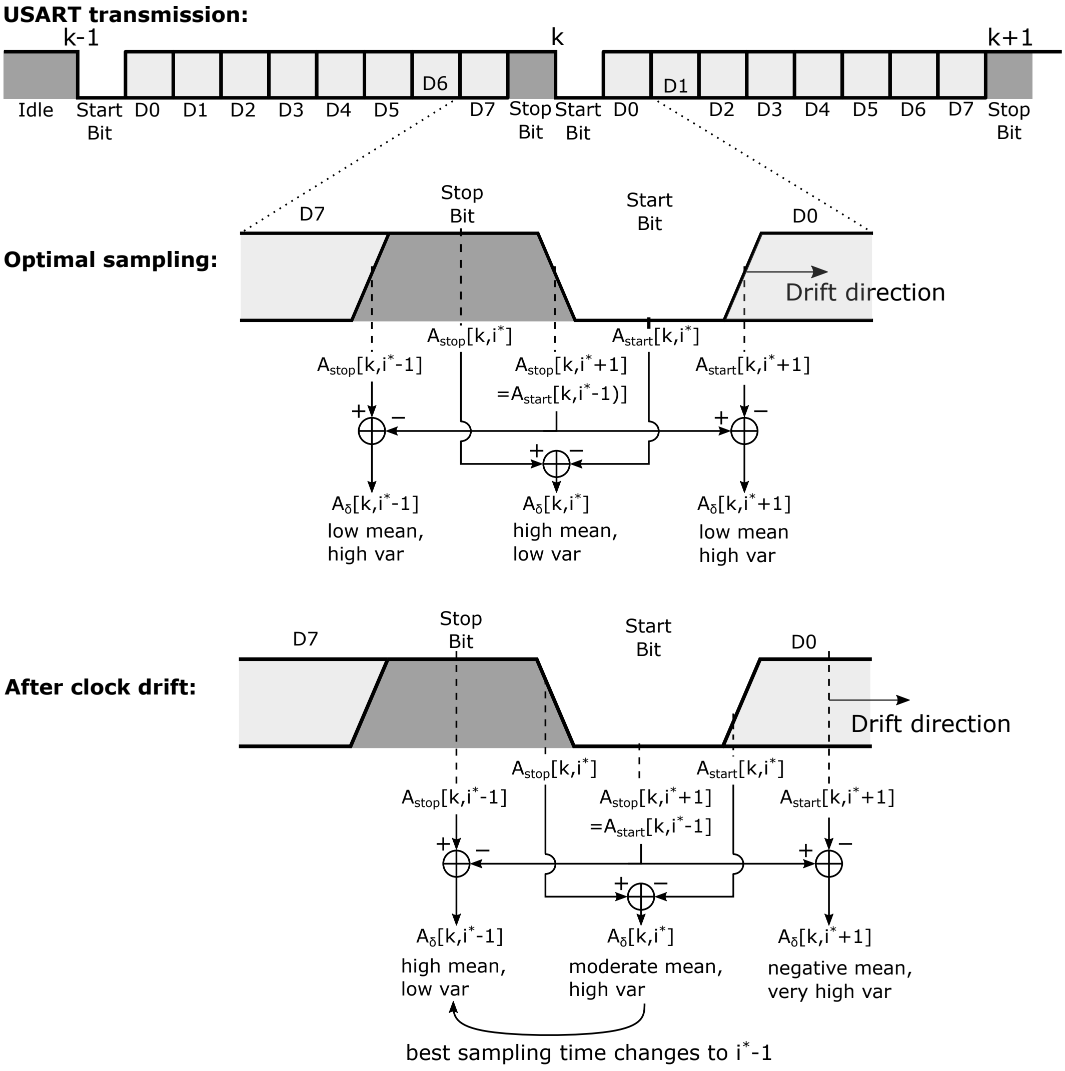


Figure 3.13: CCR principle of operation

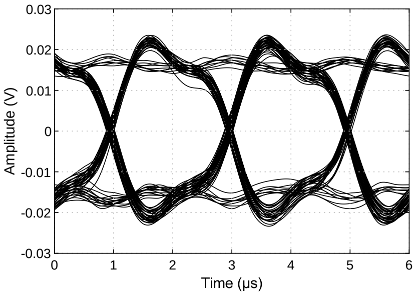


Figure 3.14: Measured eye diagram of the received signal on the hardware setup when transmitting random Manchester encoded data

EMV). Secondly, there are data bits before the stop bit and after the start bit. Inter-symbol-interference causes these variable data bits to interfere with the samples in the stop and start bit. The ISI in the start and stop is greater for samples closer to the left edge of the stop bit or right edge of the start bit.

The reason why EMA and EMV are used instead of only the values of $A_\delta[k,i]$ at time k, is the improved performance under low SNR conditions. The EMA and EMV are effectively a low pass filter on $A_\delta[k,i]$, smoothing out the additive white Gaussian noise and the clock jitter. As a result, fewer false positive phase switches occur. The downside of the low pass filter is the increased switching delay. However, since the clock drifts at a constant speed and jitter is low, very fast reaction time is not necessary. In conclusion, the function, advantages and disadvantages of the EMA and EMV are similar to the low-pass filter in a hardware PLL [33].

The values for α, Θ_a and Θ_v are found by trial-and-error. The values giving the best FER performance under low SNR conditions are the final parameters and are shown in appendix C. In section 5.1.4 we measure the FER with and without the CCR.

Algorithm 2: Recursive EMA and EMV update of $A_\delta[k,i]$ [26]

```
Result: $\bar{A}_\delta[k,i]$, $Var(A_\delta[k,i])$
Input : $A_\delta[k,i]$, $\bar{A}_\delta[k-1,i]$, $Var(A_\delta[k-1,i])$
diff $= A_\delta[k,i] - \bar{A}_\delta[k-1,i]$;
incr $= \alpha \times$diff ;
$\bar{A}_\delta[k,i] = \bar{A}_\delta[k-1]$+incr;
$Var(A_\delta[k,i]) = (1-\alpha) \times (Var(A_\delta)[k-1,i]+$ diff $\times$ incr ) ;
```

3.4.3.5 Preamble Detection

To detect the preamble, we use the samples $y[Wk+i^*]$ with $i^* = \text{argmax}_i(\tau_i)$ obtained from equation 3.24. Two checks are performed on $Y[Ok+i^*]$ to determine if it contains a preamble:

1. Compare τ_i with a fixed threshold Θ_{pr}. The check succeeds if $\tau_i > \Theta_{pr}$.

2. Decode the bits in $y[Wk+i^*]$ for $10 \leq k < (N_{sync} + N_{sfd})$ and compare the decoded bits to the template $x_{sync+sfd}$. We choose to skip decoding the first 10 bits of the preamble as the transition from an idle to busy channel can cause a transient at the receiver front-end. The transient makes the first bits of the preamble unreliable.

When the two checks succeed frame decoding starts at the first bit after the SFD.

Algorithm 3: CCR new best phase (i^*) selection procedure.

```
Result: $i*$, $\bar{A}_\delta[k,i]$,   $Var(A_\delta[k,i])$   for $i=(i^*-1)$ to $(i^*+1)$
parameter : $\Theta_a$, $\Theta_v$
Input       : $\bar{A}_\delta[k,i]$, $Var(A_\delta[k,i])$ for $i=(i^*-1)$ to $(i^*+1)$
if $\bar{A}_\delta[k,i^*-1] > \bar{A}_\delta[k,i^*]+\Theta_a$ and
  $Var(A_\delta[k,i^*-1]) < \Theta_v \times Var(A_\delta[k,i^*])$ then
    for $i=(i^*-1)$ to $(i^*+1)$ do
        $\bar{A}_\delta[k,i]=0$;
        $Var(A_\delta[k,i])=0$;
    end
    $i^*=i^*-1$;
else if $\bar{A}_\delta[k,i^*+1] > \bar{A}_\delta[k,i^*]+\Theta_a$ and
  $Var(A)_\delta[k,i^*+1] < \Theta_v \times Var(A_\delta[k,i^*])$ then
    for $i=i^*-1$ to $i^*+1$ do
        $\bar{A}_\delta[k,i]=0$;
        $Var(A)_\delta[k,i]=0$;
    end
    $i^*=i^*+1$;
end
```

3.4.3.6 Received Signal Strength Indication

The RSSI is calculated by the UD at the start of each DL frame. By also decoding the frame header the UD can determine the RSSI for all nearby APs. The application layer uses this information to only decode audio from the nearest AP. We explain this mechanism in chapter 3.2.1.3

Since LightTour only uses the RSSI to determine the closest AP, a very simple metric of the RSSI is sufficient. In more complex VLC localization algorithms such as trilateration [24] the RSSI is obtained by calculating the energy (i.e. variance) of the received signal. The metric used in LightTour is:

$$RSSI = \max_i \tau_i = \tau_{i^*} \tag{3.31}$$

which is the cross correlation calculated on the preamble during preamble detection in equation 3.24. Hence, no additional computation is required to obtain the RSSI.

3.5 Conclusion

The goal of this work is to provide audio delivery to a large number of users using a low-complexity system. In the previous two chapters we stated that current state-of-the-art solutions cannot achieve all these goals. This chapter has presented a new system that solves these limitations.

The design of the LightTour MAC protocols is based on four principles:

1. Each AP can sense nearby downlink and uplink transmissions using a photodiode sensitive to a wide wavelength.
2. Each AP can detect collisions with other nearby downlink transmissions.
3. The downlink hidden node problem is avoided by a careful placement of APs.
4. Uplink transmissions can only occur during downlink transmissions.

The full duplex channel and collision detection are enabled by a low-complexity SIC algorithm at each AP. The resulting VLC-CSMA/CD downlink MAC protocol enables broadcast audio transmissions in a reliable and bandwidth-efficient manner. The uplink hidden UD problem is solved using the novel BSMA protocol which indicates uplink channel status in dedicated busy slots during transmission of the downlink payload.

Downlink and uplink application layers are developed on top of the new MAC protocols. The downlink application layer is based on the transmission of parallel audio streams in combination with a simple flow/congestion control algorithm. The uplink application layer uses an automatic retransmission scheme to reliably deliver text messages.

A PHY layer is implemented providing a downlink and uplink data rate of 210kb and 100kb/s. Finally, it is found the downlink requires a CCR mechanism at the UD to provide low BER transmissions. A novel low-complexity software-based solution is designed based on blind oversampling. Clock drift is detected at the edge between USART stop and start bits which are periodically interleaved with data bits.

The subsequent chapter discuss the implementation of the system design on a hardware setup. The chapter after that evaluates the performance of the system design by means of hardware measurements and simulations.

Chapter 4

Implementation

We present our implementation of the system designed in the previous chapter on a test bench setup which is based on Arduino Due microcontroller boards. The test bench setup allows for validation of the system design in chapter 5. We first describe the hardware setup and the individual components such as the white LED transmitter front-end, the IR LED transmitter front-end and the PD receiver front-end. Next, we present the software implementation on the Arduino Due.

4.1 Hardware

Hardware modules for both the AP and the UD are built. Each AP and UD hardware module consists of a transmitter, receiver and Arduino board. In addition, each UD also contains a speaker and audio amplifier for audio playback. Figure 4.1 shows a picture of an AP module and figure 4.2 shows a UD module. We build three AP modules and two UD modules. With this number of APs and UDs most of the functionality of the LightTour system can be validated. The hardware setup is complemented by software simulation for validation with a high number of APs and users (chapter 5).

4.1.1 Setup

Figure 4.3 shows a diagram of the setup. The three APs are mounted at a vertical distance of $d_{ver} = 1.7m$ from the floor forming a triangle with $d_{hor} = 1.2m$ side length. The UDs are mobile. The APs and UDs are connected to a central USB hub attached to a computer. The USB connections provide power to the AP and UD modules as well as to a PC serial connection for programming of the Arduino boards and monitoring the system during runtime. The UDs can be powered using onboard batteries which the UDs to move unhindered without the wired USB connection. Additionally, the white LED front-ends at each AP are powered by a secondary 5V rail. The 5V rail is provided by a 35W AC/DC wall adapter.

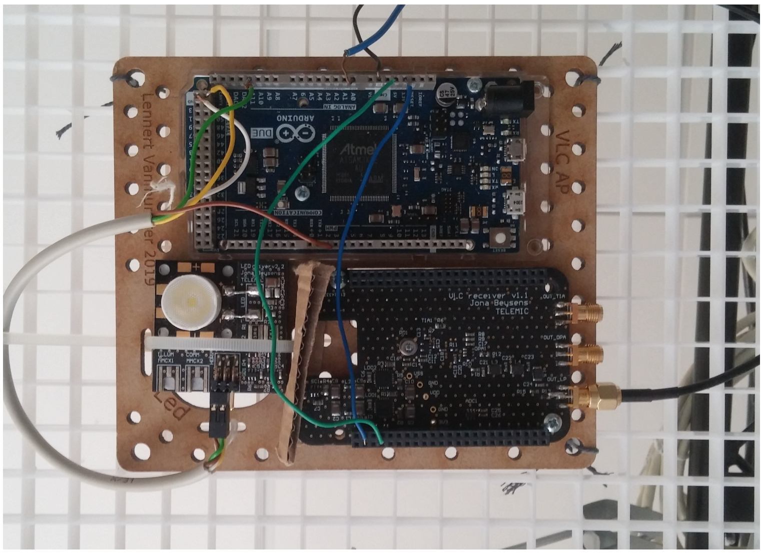


Figure 4.1: Hardware AP module

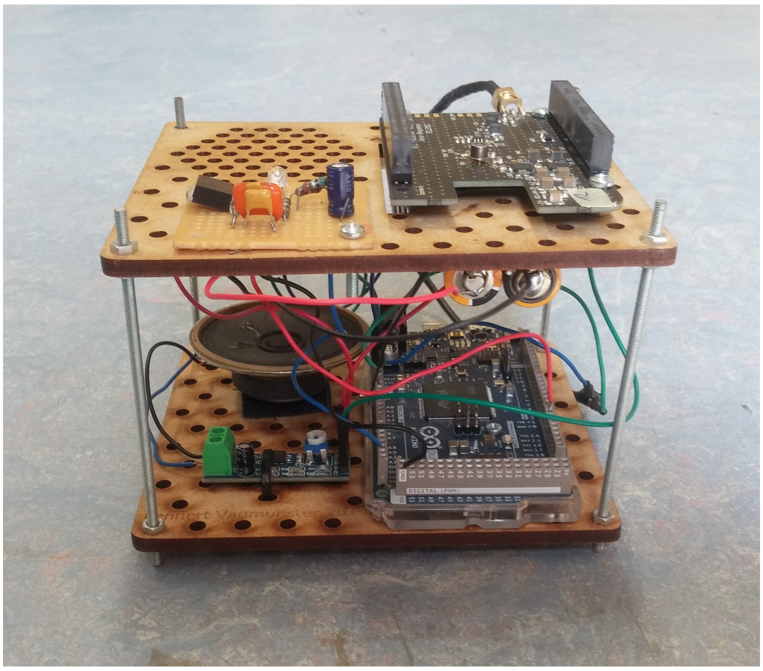


Figure 4.2: Hardware UD module

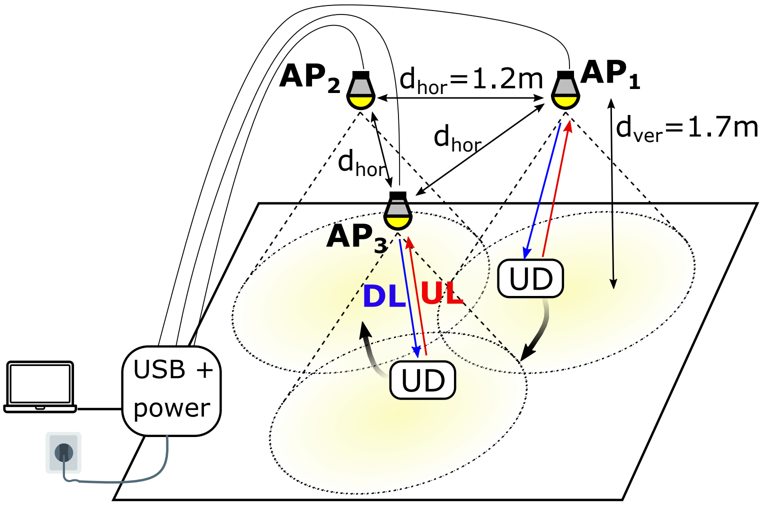


Figure 4.3: Experimental setup; icon sources: laptop [28], wall plug [58]

4.1.2 Arduino Due

The Arduino Due is an open-source prototyping board. Its low cost, sufficient performance and large availability of web resources [9] makes it an ideal choice to develop the system. The Arduino is typically programmed in C or C++. The Due is based on the AT91**SAM3X**8E microcontroller [7]. The specifications of the SAM3X that are relevant to this work are [11]:

- System
  - ARM Cortex M3. 32-bit, 84 MHz general purpose processor (i.e. the CPU)
  - 96kB embedded static random access memory (SRAM) + 512kB embedded flash memory
  - Powerful nested vector interrupt controller (NVIC, part of Cortex M3). The NVIC allows up to 30 interrupts with 16 priority levels for each. interrupt. Interrupt preemption is supported. Entry and exit of an interrupt service routine (ISR) has low latency (12 clock cycles [10]). Performance is improved further by interrupt tail-chaining.
  - Peripheral direct memory access (DMA) controller. This module allows offloading of peripheral control and data transfers from the central processor
- Peripherals
  - ADC with up to 1MHz sampling rate and up to 12-bit resolution
  - DAC with up to 1MHz sampling rate and up to 12-bit resolution
  - Hardware USART controller

- Timer counter (TC) module providing programmable timer channels. Each channel can be programmed to generate interrupts.
- 103 input/output (I/O) lines. The Arduino Due board has 54 physical general purpose input/output pins (GPIO) pins.
- USB connection for programming, monitoring and debugging

4.1.3 Transmitter and Receiver Front-Ends

We use the white LED transmitter and PD receiver front-ends from the work by Beysens et al. [16]. In his work, TX and RX front-ends are designed for a VLC system similar to LightTour. Similarities include the system topology and the utilization of the inter-AP NLOS signal in the MAC protocol. These similarities make the transmitter and receiver design compatible with the LightTour system. Additionally, we design a simple IR front-end. The most important specifications and components of the front-ends are shown in table 4.1 and 4.2 respectively.

4.1.3.1 Access Point Transmitter Front-End

Figure 4.4 shows a picture and schematic of the AP TX front-end designed by Beysens et al. [16]. The AP TX front-end uses a high performance white led (CREE XT-E), covered by a 30° full width at half maximum (FWHM) lens. The driver circuit consists of two parallel branches each consisting of a power transistor and resistor in series. The driver circuit is controlled by two inputs as shown in the schematic. Each input controls one of the branches. This setup allows the LED to be driven at three discrete luminous intensity levels (ξ_v). The three levels are no light ($\xi_{v,off}$) when both inputs are low, half intensity ($\xi_{v,ill}$) when only the illumination input is high and full intensity ($I_{v,high}$) when only the communication input is high. The $\xi_{v,off}$ and $\xi_{v,ill}$ levels are used for data transmission, respectively representing a LOW and HIGH symbol. The $\xi_{v,ill}$ level is used for illumination when the transmitter is idle. The resistors in the driver circuit are tuned so the average luminous flux at 50% duty cycled data transmission and during $\xi_{v,ill}$ are equal.

The average electrical power consumption of the AP front-end, is measured by Beysens et al. [16]. During communication at 50% duty cycle the average power consumption is $P_{w,comm} = 3.04W$ and during illumination $P_{w,ill} = 2.51W$. The reason for this difference are the physical characteristics of the LED. There is a convex relationship $P(\Phi)$ between an LED's power consumption P and its luminous flux Φ [64]. The power consumption with a fixed light level μ_Φ (in the illumination mode) is $P_{ill} = P(E[\Phi])$. In communication mode, on the other hand, $E[\Phi]$ is constant while Φ is modulated. This leads to a power consumption in communication mode of $P_{w,comm} = E[P(\Phi)]$. Consequently, through Jensen's inequality [50]: $P_{w,ill} < P_{w,comm}$.

Symbol	*Value*	*Unit*	*Description*
AP TX front-end			
$\phi_{w,1/2}$	15	°	White LED half power semi-angle (including lens)
$I_{w,high}$	900	mA	Current draw when transmitting high symbol
$P_{w,comm}$	3.04	W	Average power consumption in communication mode
$P_{w,illum}$	2.51	W	Power consumption in illumination mode
$\Phi_{w,high}$	$\sim$300	lumen	Luminous flux at current $I_{w,high}$

(a) AP TX front-end

Symbol	*Value*	*Unit*	*Description*
UD TX front-end			
$\phi_{ir,1/2}$	19	°	IR LED half power semi-angle
$I_{ir,high}$	150	mA	Current draw when transmitting high symbol
$P_{ir,comm}$	0.225	W	Average power consumption in communication mode
B_{ir}	7	MHz	Bandwidth
$\tilde{P}_{ir}$	$\sim$75	mW	Radiant power at $I_{ir,high}$

(b) UD TX front-end

Symbol	*Value*	*Unit*	*Description*
RX front-end			
$f_{c,1}$	7.2	kHz	High pass filter cut-off frequency
$f_{c,2}$	695	kHz	Low pass filter cut-off frequency
λ_{pd}	900	nm	PD peak sensitivity wavelength
S_{pd}	0.64	A/W	PD peak sensitivity

(c) RX front-end

Table 4.1: Hardware specifications

LED	CREE XT-E
Lens	TINA FA10645
Mosfet	NTR4501

(a) AP TX front-end

PD	S5971
TIA	OPA659
AC Amp	OPA355

(b) RX front-end

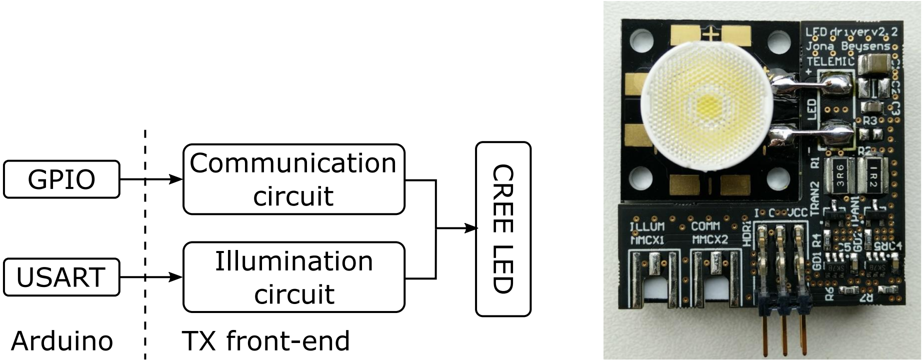
IR LED	TSHG5510
Mosfet	PSMN017-30EL

(c) UD TX front-end

Speaker	52-130
Amplifier	LM386

(d) UD audio

Table 4.2: Hardware components



(a) Schematic

(b) Picture

Figure 4.4: AP TX front-end by Beysens et al. [16]

4.1.3.2 User Device Transmitter Front-End

The circuit of the UD's IR TX front-end is shown in figure 4.5. The n-channel metal-oxide-semiconductor field-effect transistor (NMOS) controls the the IR LED forward current depending on the voltage level at its gate which is controlled by the Arduino USART output pin. The resistor R1 is designed for a forward IR LED current of $I_{ir,high} = 150mA$ when the microcontroller outputs a high signal. From the IR LED's datasheet this corresponds to a radiant power of $\tilde{P}_{ir} \approx 75mW$. Resistor R2 and R3 limit the constant current drawn from the Arduino's USART pin to $I_{usart} = \frac{V_{usart}}{R2+R3} = \frac{3.3V}{5.22k\Omega} = 0.63mA$. The nonzero gate capacitance of the NMOS results in a current flowing into and out of the gate when switching the NMOS. The maximum USART current when switching is limited to $\max I_{usart} = \frac{V_{usart}}{R2} = \frac{3.3V}{220\Omega} = 15mA$. The maximum constant current the USART pin can source is also $15mA$ [11].

The IR front-end is also designed for a high modulation bandwidth. The inherent rise and fall time of the LED are both 15ns. When driving the circuit with a 1MHz square wave at the NMOS gate we measure a rise time of $t_r = 50ns$ and fall time of $30ns$ of the voltage across the LED using an oscilloscope. The increase in switching time over the IR LED data sheet value is because the switching speed of the NMOS is limited by R2 which limits the current flowing to the gate. Hence, the charging time of the gate capacitance is increased. Using the rule-of-thumb $B[GHz] = \frac{0.35}{t_r[ns]}$ [17] the bandwidth of the circuit is $B = 7MHz$. This is 10 times higher than the bandwidth of the RX front-end.

Ignoring the power consumption of switching the MOSFET, the approximate power consumption of the circuit when transmitting a high symbol is:

$$P_{ir,high} = Vcc \times I_{ir,high} = 3V \times 150mA = 0.45W \tag{4.1}$$

When transmitting a 50% duty cycle signal (during communication) the average power consumption is $P_{ir,comm} = P_{ir,high}/2 = 0.225W$.

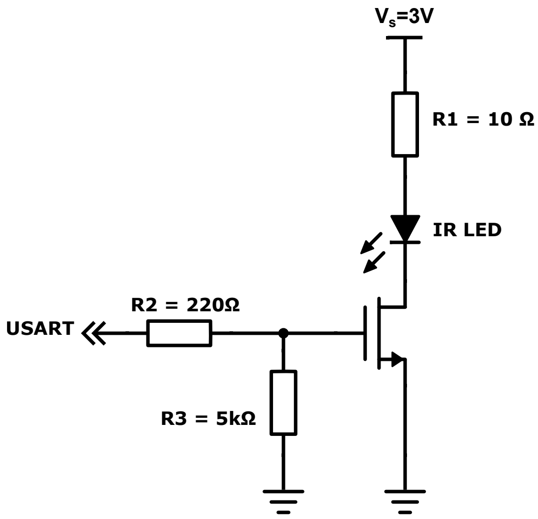
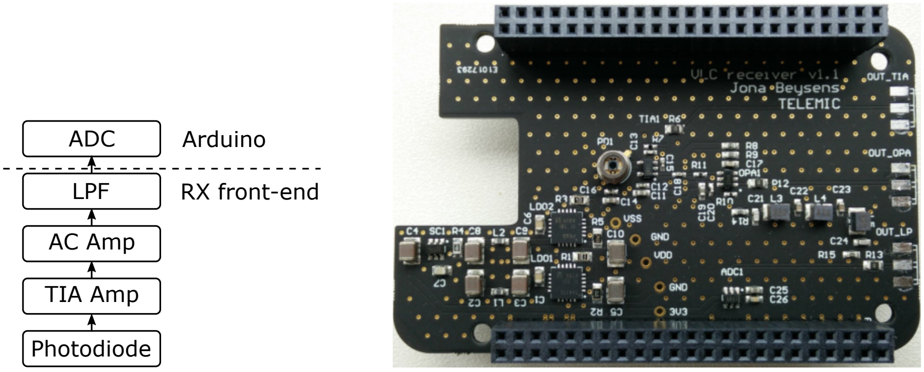


Figure 4.5: IR TX front-end circuit diagram



(a) Schematic

(b) Picture

Figure 4.6: RX front-end by Beysens et al. [16]

4.1.3.3 Receiver Front-End

The same PD (and RX front-end) is used for both the receiver at the AP and at the UD. Figure 4.6 shows a picture and block diagram for the RX front-end designed by Beysens et al. [16]. The RX front-end circuit amplifies and filters the output signal from the PD. The circuit consists of three stages. The first stage is a low-noise transimpedance amplifier (TIA) which amplifies and converts the PD current to a voltage at the amplifier output. The second stage is an AC coupled amplifier which removes the low-frequency ambient light and amplifies the signal further. The third stage is a 7th order passive low-pass Butterworth filter, to avoid aliasing in the subsequent ADC at the Arduino.

4.1.4 Speaker and Amplifier

Each UD is equipped with a 0.25W speaker and an audio amplifier for audio playback. This allows for easy validation and demonstration of the system functionality. The speaker can be substituted for headphones if desired. The audio amplifier is an of the shelf module based on the LMS386 IC.

The DAC of the Arduino is used to generate the audio waveform. The voltage range of the DAC output is 0.55V to 2.75V [11]. We center the audio signal around the center of the output voltage range: $\frac{2.75V-0.55V}{2} = 1.1V$ The LM386 module expects an AC-coupled input. We AC-couple the input of the audio amplifier using a $10\mu F$ capacitor in series with the input of the amplifier module. This makes the complete amplifier circuit similar to the "Simple Audio Player" Arduino tutorial [8] .

4.2 Software

The software running on the Arduino Due's is written in C. We describe the software functionality the AP and on the UD.

Optimization

Many functions require real time processing of the inputs. These functions must process samples at the sample arrival rate. Hence, for these functions the execution time per sample must be smaller than the time interval between arrival of samples. Without optimization, real time processing is not possible. Consequently, optimizing the code to minimize execution time is necessary.

The three methods that have the biggest impact on performance are the compiler settings, fixed-point arithmetic and using DMA to offload I/O operations from the central processor. Firstly, the C code is compiled using the '-O3' optimization flag which aggressively optimizes the code to minimize execution time at the cost of some extra memory and compile time. Secondly, since the Arduino Due does not have a hardware floating point unit, floating point operations have to be implemented using software by the compiler. Software implementation of floating point operations require can take hundreds of times more instructions to perform the same operation as the integer counterpart [81]. For this reason, we implement all variables and operations using fixed-point arithmetic. Thirdly, we explain the usage of DMA further during the next two sections.

Libraries

Some functionality of the AP and UD programs is implemented using external libraries. Firstly, LightTour uses the Speex audio codec [80]. Speex is an open-source speech codec based on code-excited linear prediction (CELP) [56]. The reasons why this codec is chosen are the detailed documentation, relatively simple implementation, low memory and time complexity and the sufficient coding efficiency. In future versions of LightTour, Speex can be substituted by another codec that supports music such as Opus [79] providing it meets the time and memory complexity requirements.

The second external library is ‘FastCRC’ by ‘FrankBoesing’ [27] which is written specifically for Arduino and provides fast CRC coding using LUTs. The last external library is Arduino-FEC by ‘simonyipeter’ [57] which provides an Arduino implementation of an RS encoder and decoder based on the Bose-Chaudhuri-Hocquenghem (BCH) view. Encoding and decoding with the BCH view has lower time and memory complexity than the original RS view [76]. The RS block length and message length can be chosen by the user.

4.2.1 Access Point

Figure 4.7 shows a schematic of the operation of the AP software. The hardware TX front-end is controlled by two output lines. The first is the USART hardware peripheral on the SAM3X which is connected to the communication input pin on the AP TX front-end to modulate the LED. The second is a GPIO line connected to the illumination input pin on the AP TX front-end. The output analog voltage signal of the RX front-end is connected to the SAM3X’s onboard ADC and is sampled at 1 MHz.

To decrease modulation and ADC sampling overhead on the central processor, the ADC and USART peripherals are controlled by the SAM3X’s peripheral DMA controller. By using DMA, the transmission of a frame only requires storing the encoded frame at some memory location and sending the transmit command to the DMA. When the transmission finishes an interrupt is triggered. The central processor is not involved during transmission. Similarly, the ADC samples are written to a buffer by the DMA. An interrupt is triggered every $N_{adc} = 20$ samples allowing the samples to be processed by the CPU. The greater N_{adc}, the greater the delay between a sample arriving and the time at which it is processed. This increases the DL and UL MAC protocol vulnerable period ($t_{v,dl}$ and $t_{v,ul}$) and hence lowers performance [69]. The smaller N_{adc}, the greater the context switching overhead caused by the ADC interrupt.

The software consists of four routines. The ‘handlers’ in figure 4.7 are ISRs and are called automatically when their corresponding interrupt is triggered. We enable interrupt preemption for all ISRs on the SAM3X’s NVIC and configure the priority of each ISR. The routines in order of highest to lowest priority are the TC0 (timer counter 0) handler, ADC handler, TC8 handler and the main loop. Communication between routines is implemented by shared global state variables and thread-safe queues [5]. A high priority routine can alter a shared global variable that is being read/modified by a low priority routine. As a result, data corruption occurs. In order to avoid this problem, high priority ISRs are briefly disabled whenever a shared variable is read/modified. When the high priority ISR is enabled again, any pending interrupts are processed by the ISR.

The most important routine is the ADC handler. It is triggered every $N_{adc} = 20$ samples ($20 \mu s$) by the DMA controller. Consequently, all functions in the ADC handler have to process finish processing their inputs before the next interrupt is triggered. Functions that do not require real-time processing and/or require longer than $20 \mu s$ processing time in the ADC handler are therefore implemented in the lower-priority TC8 handler. When no DL frame is being transmitted, the ADC handler performs CCA. Otherwise, the ADC handler implements SIC, CD, UL preamble detection, UL line code decoding and UL header decoding. The PHY DL transmission is also started in the ADC handler. This allows synchronization of PHY transmission and ADC sampling time. Since UL preamble detection is implemented in the ADC handler, the busy tone is also updated here.

The TC8 handler is a low priority routine implemented using the TC peripheral. The routine is triggered by the ADC handler when an UL message is received and when the DL transmission finishes. The functions in the TC8 handler require more than $20 \mu s$ to finish and can therefore not be implemented in the ADC handler. When the TC8 handler receives the line code, decoded UL messages from the ADC handler, the TC8 handler decodes the payload CRC and stores the information required for assembly of the A-ACK and for further processing by the UL application layer. At $320 \mu s$ before the end of the DL payload transmission the handler is called again to assemble and append the A-ACK.

The TC0 handler is the highest priority ISR. The TC0 handler switches the TX front-end back to illumination mode when DL transmission finishes by driving the USART line low and the GPIO line high. At the start of DL frame transmission in the ADC handler, the PHY sets TC0 to expire when the frame transmission finishes. As a result, the TC0 handler is called automatically when the transmission finishes.

Finally, the main loop handles the application and MAC layer of the DL transmissions. The main loop implements flow/congestion control, parallel audio streams and the DL MAC protocol. When a frame arrives from the DL application layer at the DL MAC layer, the DL MAC sends a command to the PHY layer to start transmission in the next execution of the ADC handler. This way, the USART transmission and ADC sampling can be synchronized which is necessary for SIC. The second function in the main loop is the UL application layer. The UL application processes the received messages in the UL MAC buffer. The messages are transmitted to a PC in the correct order over the USB connection. The USB connection is also used by the other routines for monitoring and debugging (not shown in figure 4.7).

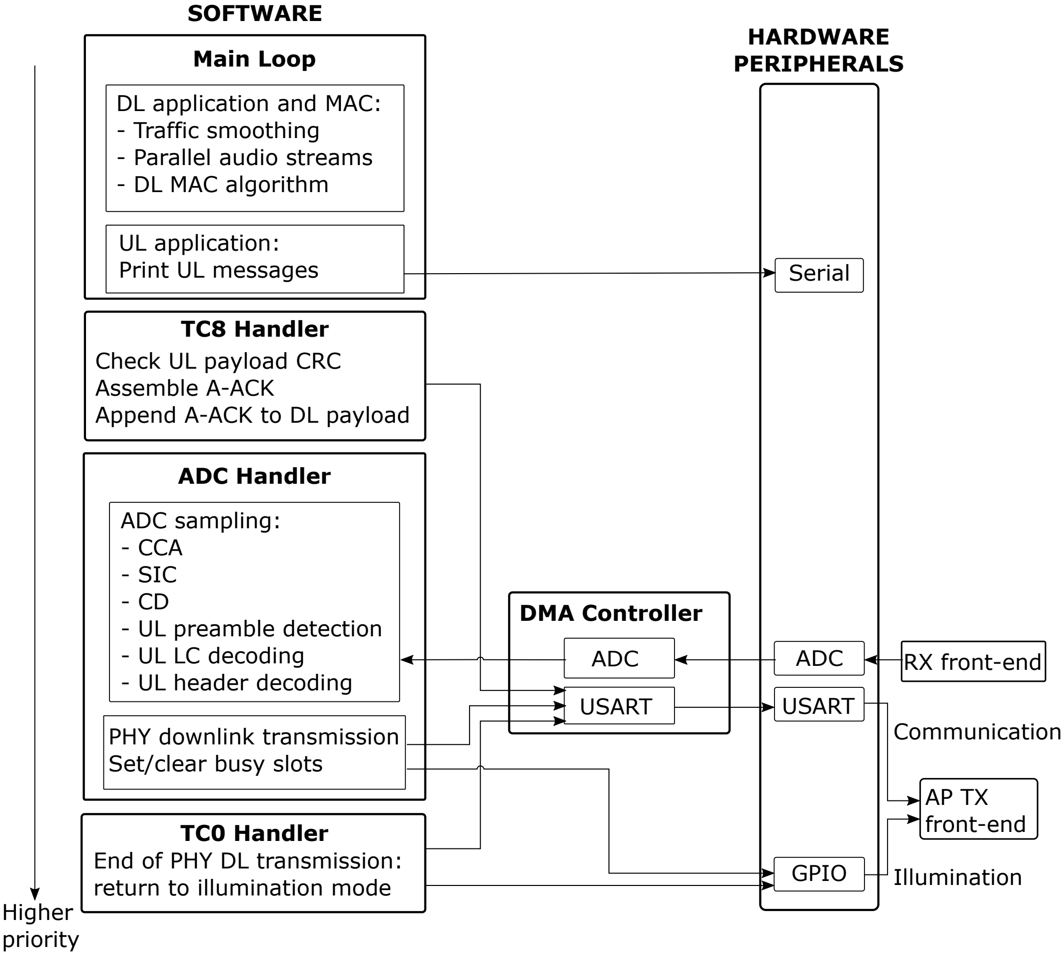


Figure 4.7: AP Arduino operation block diagram

4.2.2 User Device

The UD program structure is shown in figure 4.8. It follows a similar structure to the AP program. ADC sampling happens in the same way as in the AP program. The ADC handler now controls DL preamble detection, DL CCR, enqueueing of DL samples for further decoding by the TC8 handler and the decoding of the BS in the DL payload. Additionally, the PHY UL transmission is also started in the ADC handler. The IR LED is controlled by one physical pin on the Arduino instead of two pins for the AP TX front-end. The output pin controlling the IR LED is multiplexed between the USART hardware peripheral and the GPIO. In the idle state the output of the USART module is high as shown in figure 3.13. In order to save energy and not damage the IR LED during the idle state, control of the IR LED is given to the GPIO peripheral and the IR LED is driven low by the GPIO. The TC7 handler controls the switching between USART and GPIO at the end of the UL transmission.

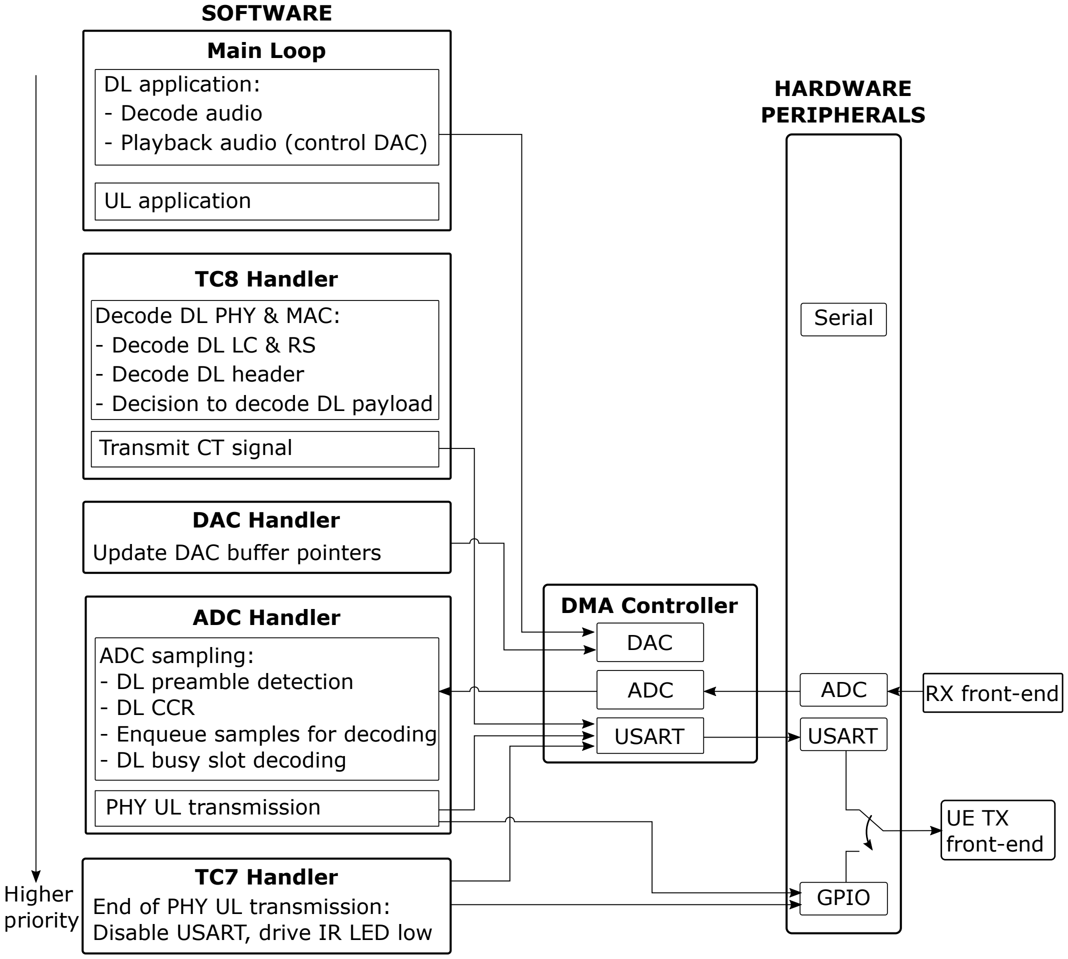


Figure 4.8: UD Arduino operation block diagram

Similarly to the AP program, the TC8 handler processes all functions that require longer than $20\mu s$ to finish. Those functions include decoding of the DL line code, RS and frame header. Based on the decoded frame header a decision is made by the application layer to decode the DL payload and the CT signal is transmitted if required.

The main loop decodes the received audio frames using the Speex library. The result is a 16kHz digital audio waveform. The samples are written to the Arduino onboard DAC using the DMA controller. The DAC handler is called periodically to update the DAC buffer. The main loop also controls the UL application and MAC FIFO buffer Q_{ul}. The serial module is used by almost all routines for printing monitoring messages (not shown in figure 4.8).

4.2.3 Conclusion

This chapter has presented the implementation of a small-scale system which enables evaluation of the proof of concept of the system designed in the previous chapter. Hardware modules based on Arduino Due microcontrollers are built for both the AP and UD. Data modulation is offloaded from the central processor using USART. The powerful NVIC on the Arduino Due provides interrupt priority levels with preemption. This principle is used in the software implementation to provide multiple subroutines with different priority levels which have a very low context switching overhead. In the next chapter we will analyse the performance of the system based on measurements on the hardware setup and on simulations.

Chapter 5

Performance Evaluation

In this chapter we analyse the performance of the system designed in chapter 3 and implemented in chapter 4. This chapter is divided into two parts. The first part studies the performance of the implementation and PHY layer. We measure the execution time of the most important algorithms, the received signal strength, the SIC performance, the FER and the CCA and CD performance. The second part measures the performance of the MAC and application layer. Hardware measurements are complemented by Matlab Monte Carlo simulations in order to study the performance for a large number of APs and UDs. We measure the throughput and delay of the DL and UL MAC protocols and compare them to existing protocols. At the application layer we simulate the average user wait time until the start of an audio stream for several application layer strategies. Finally, a simulation combining the MAC and application layer is performed measuring the performance of the system in a grid topology of APs.

5.1 Physical Layer

5.1.1 Arduino Code Timing

We measure the execution time of some key functions on the Arduino. As stated in section 4.2, the software is written in C and compiled with the '-O3' flag to optimize the code for shorter a execution time. The first experiment compares SIC-related functions such as LMS, FIR and our LUT method (section 3.4.2.2). Figure 5.1 shows the execution time per 10 DL symbols (i.e. per $20\mu s$) of those functions. The time is measured by executing each function for 10^4 DL symbols and normalizing the result. As explained in section 3.4.2.2, each received sample should be processed in one downlink symbol period ($2\mu s$). The first function on the figure is the LMS algorithm which recursively updates and computes the output of an 11-tap FIR filter. The second, third and fourth functions are fixed 11-tap FIR filters which only compute the filtered signal. These three functions differ in the way the FIR filter output is computed:

FIR mla: the output is computed using multiplications:

$$\texttt{y[i]+= h[j]*(x[i-j]);} \tag{5.1}$$

In assembly this line is converted to multiply-accumulate (mla) instructions.

FIR if: the output is computed using a conditional statement:

$$\texttt{y[i]+= (x[i-j]==-1) ? h[j] : -h[j];} \tag{5.2}$$

This is possible since `x` can take only two values: -1 and 1.

FIR logic: the output is computed using a branchless logic statement [6]:

$$\texttt{y[i]+= (h[j] \^{} -x[i-j]) + x[i-j];} \tag{5.3}$$

In this equation `x` is stored as a binary value that is either 0 or 1. The '^'-symbol denotes the exclusive or (XOR) operation .

The LUT algorithm, which uses equation 3.10 and 3.11 to compute the output and update the LUT index, requires $\sim$75% less computation time than the FIR filters and $\sim$90% less than the LMS algorithm. The cost of the LUT algorithm is an extra $2^{\Gamma} = 2^{11} = 2048$ bytes of memory where Γ is the LUT order.

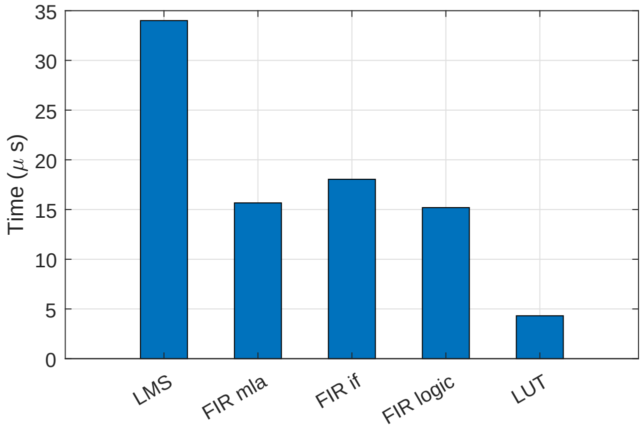


Figure 5.1: Execution time of SIC-related algorithms per 10 received downlink symbols ($20\mu s$)

Figure 5.2 shows the execution time of some other functions running at the AP and UD. The CPU usage (percentage of CPU time) of a function is the time shown divided by $20\mu s$. Some observations can be made from the figure. Firstly, the most costly function at the AP is SIC. There is an overhead of $11\mu s - 4\mu s = 7\mu s$ in the SIC function due to fixed point calculations, which require extra scaling operations, and state variable updates. In Figure 5.2, the times are measured without this overhead.

Secondly, our CCR mechanism is one of the cheapest functions requiring a CPU usage of 20% during reception. Thirdly, 4b6b decoding is slow (70% CPU usage during reception) and does not require computation in real time. Therefore, 4b6b decoding is implemented in the lower priority TC8 handler (section 4.2.2). Fourthly, RS and Speex decoding together require up to 185% CPU usage. They can thus not be computed in real time. Instead, RS and Speex are processed in low-priority routines and operate on large buffers of line code or RS decoded data. Downlink IFS and traffic smoothing are necessary to prevent overflow of the line code and RS buffers. Lastly, the CRC time shown is normalized to the UL payload length. Calculating the CRC requires only $0.18\mu s$ per $20\mu s$ of payload length. This is only $\frac{0.18\mu s}{11.05\mu s} = 1.6\%$ of the time required for RS decoding (in the best case for RS).

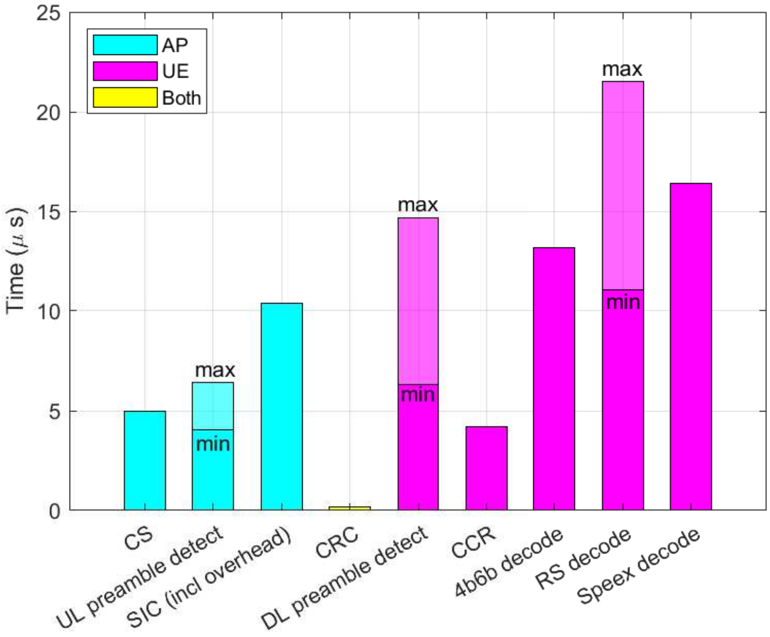


Figure 5.2: Execution time of common functions at AP and UD per $20\mu s$

5.1.2 Signal and Noise Power

We measure the received signal and noise power on the hardware for different relative positions of the APs and UDs. Having a measurement of the SNR is useful as it enables us to estimate the theoretical BER and the classification accuracy of the CCA mechanism.

All signal and noise power measurements are performed on the Arduino after the signal is converted by the ADC. The average power of a discrete signal $y[k]$ in the

	AP_1	AP_2	AP_3	Average
$P_n(LSB^2/sample)$	1.24	1.48	1.45	1.39

Table 5.1: Measured noise power on the hardware setup

interval $k=[0,K]$ is [44]:

$$P_{y,theory} = \frac{1}{K}\sum_{k=0}^{K}|y[k]|^2. \tag{5.4}$$

We are not interested in the DC offset of the received signal added by the hardware RX front-end (section 4.1.3.3). The received signal power is measured on the Arduino using:

$$P_y = \frac{1}{K}\sum_{k=0}^{K}|x[k]-\bar{y}|^2 \tag{5.5}$$

where $\bar{y}$ is the sample mean of y in the interval $k=[0,K]$. Equation 5.5 is the same as calculating the population variance of y [23, p.38,39]. The unit of measurement of y is the quantization resolution of the ADC. From the datasheet of the SAM3X , the value of the quantization resolution is $LSB = 732\mu V$ [11]. The unit of measurement for P_y is thus $LSB^2/sample = (732\mu V)^2/sample$. Firstly, the noise power P_n is measured for the three APs. The noise measurement is done on the hardware setup in figure 4.3 on one AP while all three APs are in illumination mode. Table 5.1 lists the results.

Next, we measure the received power P_y for three different transmitted signals:

- **Signal 1:** the visible light LOS DL signal. This is the signal transmitted by the AP and detected by the UD.
- **Signal 2:** the visible light NLOS signal. This is the signal transmitted by one AP and detected by another nearby AP.
- **Signal 3:** the infrared light LOS UL signal. This is the signal transmitted by the UD and detected by the AP.

The UDs are placed on the floor. Hence, the vertical distance between AP and UD is 1.7m. For signal 1 and 3 we vary the horizontal distance between AP and UD while for signal 2 we vary the horizontal distance between the two APs. The received signal $y[k]$ is the sum of the of the desired signal $s[k]$ and the noise noise $n[k]$. Hence, if we assume signal and noise are uncorrelated: $P_y = P_s + P_n$ [53]. We can now calculate the SNR as follows:

$$SNR = \frac{P_s}{P_n} = \frac{P_y - P_n}{P_n} \qquad SNR_{dB} = 10\log_{10} SNR \tag{5.6}$$

The measurement results of P_s and the SNR for signal 1, 2 and 3 are shown in Figure 5.3. The power of the DL signal is up to 1000 times greater than the UL signal. The current of the PD is proportional to the incident light power [31, p.

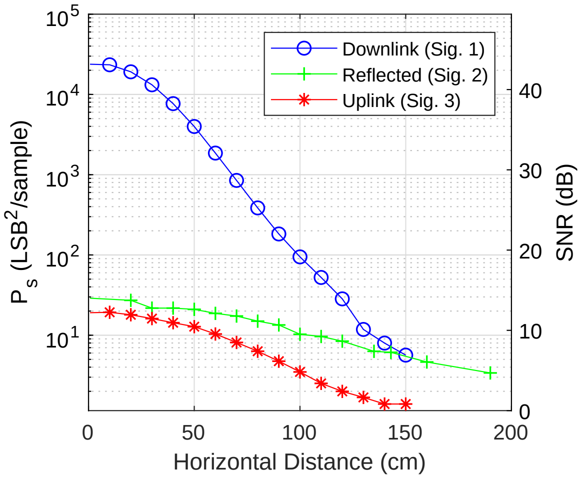


Figure 5.3: Power P_s and SNR of the DL, reflected and UL signal versus the horizontal distance d_{hor}

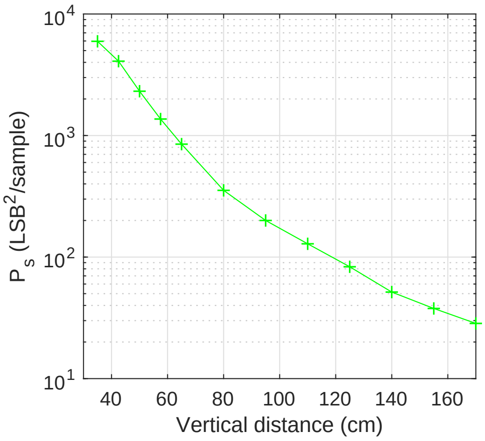


Figure 5.4: Power P_s of the self-interference signal versus the vertical distance d_{ver}

57]. As a result, the measured signal power P_s is proportional to the square of the incident light power.

Finally, we measure the received signal power depending on the vertical distance between the AP and the floor. The result is shown in figure 5.4. This measurement will be used to quantize the performance of the SIC mechanism in the following section.

5.1.3 Self Interference Cancellation Performance

The performance of the SIC mechanism is measured. We study the dependency of SIC on the line code, LUT order, the RX front-end and nearby light sources.

5.1.3.1 Line Code Dependency

We study the dependency of SIC on the LUT order and the line code. The LUT order is $\Gamma = \log_2(N_{lut})$ where N_{lut} is the number of elements in the LUT. The LUT order is also the number of previous samples that influence the value of the current estimate $\hat{y}[k]$ (equation 3.10 and 3.11). The LUT is effectively a replacement for an FIR filter and hence the LUT order is similar to the number of filter taps of an FIR filter. The memory requirement grows exponentially with Γ. The Arduino Due has 96kB of RAM. An order of $\Gamma = 16$ requires 2^{16} bytes=64kB of RAM and is thus the upper limit.

For this experiment we follow the mechanism described in chapter 3.4.2.2. The full procedure for this experiment is as follows:

1. Collect a large amount of transmitted and received signal samples from the hardware setup by transmitting an audio stream encoded in the chosen line code. The signal data is collected while the AP is at a vertical distance of 65cm from the floor.
2. Import the transmitted signal data and corresponding received ADC samples in Matlab. The data is split into two subsets: a training set (90%) and a test set (10%).
3. Using the training set in Matlab, estimate the LUT according to equation 3.11. The LUT order Γ is variable.
4. Cancel the self-interference on the test set using the LUT and equations 3.10 and 3.11. The result is the RSIC signal: $\rho[k]$.
5. Calculate the power in the RSIC signal: P_ρ.
6. The signal to RSIC power ratio is [46]:

   $$\chi = \frac{P_s}{P_\rho} \tag{5.7}$$

   where $P_s = P_y - P_n$ is the power of the received signal of interest calculated from the test set.

The experiment is repeated for several line codes: Manchester, modified 4b5b (m4b5b) [40], 4b6b and 4b6b with BSes inserted into the data stream (4b6b+bs). Figure 5.5 shows the result of the experiment. The SIC mechanism performs best on Manchester encoded data, especially at low LUT orders. The next best performing linecode is 4b6b closely followed by 4b6b+bs. The SIC performance of 4b6b and 4b6b+bs approaches that of the Manchester line code when $\Gamma > 11$.

Lastly, the SIC mechanism performs worst on m4b5b encoded data. The m4b5b line code has the lowest bandwidth overhead of the discussed line codes but unlike Manchester and 4b6b it is not DC-balanced. In m4b5b a group of four consecutive data bits is encoded to five consecutive bits. Since five is an odd number the group of encoded bits always has an unequal number of 1's and 0's. As a result the spectrum of an m4b5b encoded signal has a higher power at low frequencies than 4b6b and Manchester. The bandpass filter in the RX front-end ($r[k]$) is modelled by the LUT. The hardware bandpass filter strongly attenuates the low-frequency components. Since m4b5b has more power at low frequencies, a higher attenuation by the LUT filter model is required to simulate the hardware filter. A higher attenuation can only be achieved using a larger filter order [15]. At high filter orders, χ for m4b5b approaches χ for the other line codes.

The coding overhead for Manchester, m4b5b and 4b6b is 50%, 20% and 33% respectively. Hence, there is a trade-off between coding overhead and SIC performance. The m4b5b line code was used in an earlier version of LightTour in which full duplex

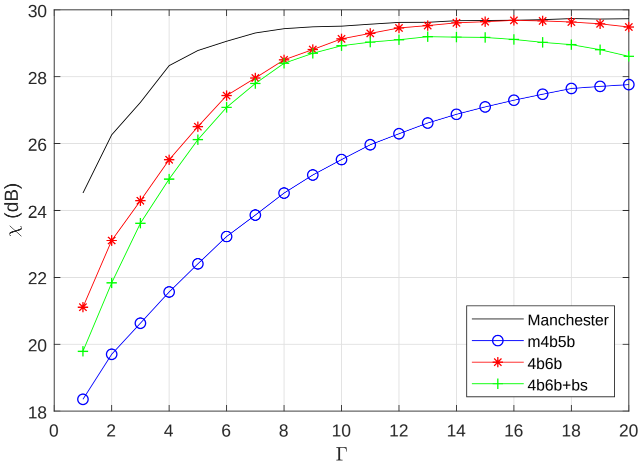


Figure 5.5: Amount of SIC suppression χ of various line codes versus the LUT order Γ at a vertical distance of $d_{ver} = 65cm$

and SIC was not implemented yet. However, due to the poor performance of m4b5b in SIC, the final version of LightTour uses 4b6b. The 4b6b line code provides a good balance between LUT size, SIC performance and coding overhead.

5.1.3.2 Hardware Dependency

We investigate the dependency of the SIC mechanism on the hardware RX front-end. We study if LUT training is required for each AP separately or if one LUT can be shared among multiple APs. For each AP, the methodology is as follows. Data collected from the AP is used to train the LUT as explained in section 3.4.2.2. LUT_x denotes the LUT obtained when using a training set collected from AP_x. Next, LUT_x is stored on AP_x for all three APs. Next, each AP transmits a 4b6b+bs encoded audio stream. In order to measure P_ρ for different values of P_y, we vary the vertical distance of each AP. For AP_2 and AP_3 we also measure P_ρ and P_y when using LUT_1 and LUT_2.

Figure 5.6a shows P_ρ versus P_y for all the tests. Likewise, Figure 5.6b shows χ versus P_y. The relationship between P_y and the vertical distance is shown in Figure 5.4. Some observations from the results are:

- P_ρ increases superlinearly with P_y. At $P_y = 2000$, which corresponds to a vertical distance of 55cm, P_ρ starts increasing sharply. When P_ρ increases, functions that rely on SIC like CD and full duplex communication suffer a performance penalty. It is hence not advised to position an AP closer than 55cm to a horizontal object. Similarly, users walking underneath the AP cause P_ρ to increase. Reflective clothing also causes P_ρ to increase and should be avoided. In a normal scenario where there is sufficient distance between AP and users, the increase in P_ρ caused by users is negligible.

		P_ρ		
Light Interference	P_n	LUT_{low}	LUT_{mid}	LUT_{high}
Low	1.18	1.83	2.06	1.66
Medium	1.43	2.04	2.19	2.42
High	4.91	7.37	7.41	6.38

Table 5.2: Measured SIC residue power P_{sic} under different light interference levels. LUT_X denotes the LUT trained for light level X.

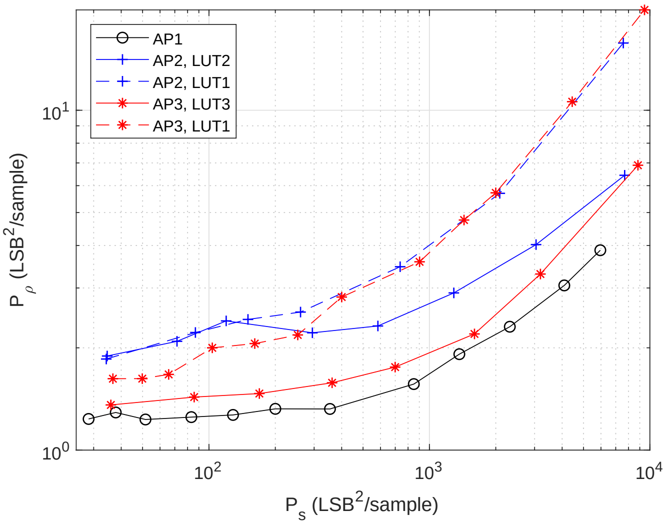
- When each AP uses its own LUT, χ for AP$_2$ and AP$_3$ is about $0.7dB$ and $2dB$ lower than AP$_1$ respectively. This is probably caused by variations in the hardware RX front-end and Arduino ADC.
- When AP$_2$ and AP$_3$ use LUT$_1$, χ increases by $0-1dB$ at low P_y and up to $3.5dB$ at high P_y. Hence, training a LUT for each AP improves the robustness of the SIC mechanism against poor placement of the AP (close to objects). Future work is possible on training the LUT on the Arduino as part of the operation of the protocol.

5.1.3.3 Light Interference

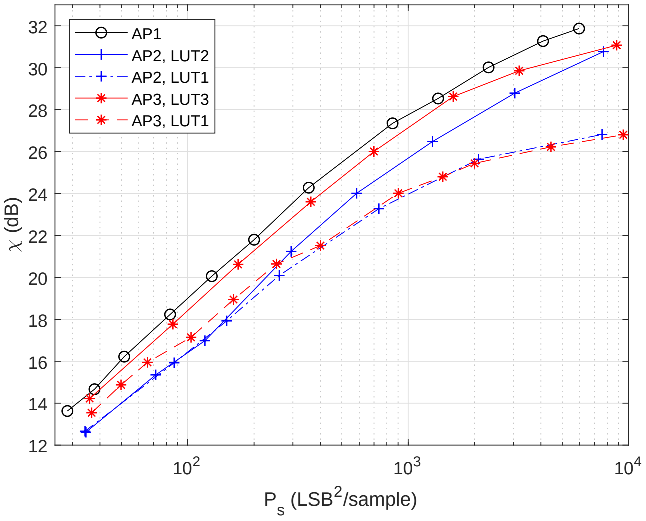
We study the effect of a nearby light source on the SIC mechanism. A change in ambient light level may affect the SIC performance. This is because of two reasons. Firstly, a high ambient light level introduces a high amount of shot noise [31], hence the noise floor is increased. Secondly, the RX front end is not a perfectly linear system. Both the PD and the subsequent circuits introduce some non-linearity [39]. Changing the ambient light level results in a different operating point on the output voltage versus illumination curve. If the new operating point is at a location on where the curvature is different, the SIC performance may be affected.

The measurements are collected on AP$_1$. We use the experimental setup shown in figure 5.7. AP$_1$ is placed at a vertical distance of 65cm from the floor. Three LUTs are obtained. $\text{LUT}_{1,\text{low}}$ is obtained from a training set collected on AP$_1$ when the ambient light is low (other light in room turned off, AP far from window). $\text{LUT}_{1,\text{med}}$ is obtained under the same ambient light but with another AP, in illumination mode and shining upward, placed at $d_{hor} = 30cm$ horizontal distance on the floor. $\text{LUT}_{1,\text{high}}$ is obtained when the AP on the floor is directly underneath the AP1.

P_y and P_ρ are measured for all nine combinations of light level and LUT. The results are shown in table 5.2. A measurement of the noise power P_n for the three light levels is also included. From the results it is clear that the SIC mechanism always achieves a P_ρ close to the noise floor. Training a LUT for a specific light level only has a noticeable benefit for a high light interference.



(a) Power in the residual self-interference after SIC P_ρ



(b) Signal to RSIC ratio: $\chi = \frac{P_y}{P_\rho}$ (amount of cancellation)

Figure 5.6: Measured SIC performance versus the received signal strength P_s when transmitting a 4b6b+bs encoded signal.

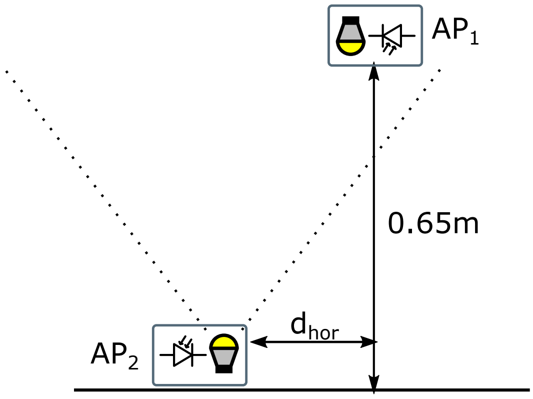


Figure 5.7: Test setup for experiment measuring SIC performance under different light interference levels

5.1.4 Frame Error Rate

5.1.4.1 Theoretical Estimate

We calculate an estimate of the FER for the DL and the UL depending on the SNR. Our assumptions are an AWGN channel with matched filter at the receiver and independent bit errors. This model does not serve as an accurate calculation of the FER for VLC. Rather, it allows comparison of our FER measurements in the next section with a simple FER estimate. In an AWGN channel with a matched filter at the receiver the BER for OOK is [20, p. 497]:

$$BER_{ook} = Q(\frac{s_{01} - s_{02}}{2\sigma_n}) \tag{5.8}$$

where s_{01} is the mean receiver output when a binary 1 is sent and s_{02} when a binary 0 is sent. Q denotes the Q-function [20, p.700,701]. The SNR obtained in section 5.1.2 can be rewritten as:

$$SNR = \frac{P_s}{P_n} = \frac{Var(s[k])}{\sigma_n^2} = \frac{(\frac{s_{01}-s_{02}}{2})^2}{\sigma_n^2} = (\frac{s_{01} - s_{02}}{2\sigma_n})^2. \tag{5.9}$$

Hence, assuming an AWGN channel with matched filter, we can calculate the theoretical BER from our SNR measurements:

$$BER_{ook} = Q(\sqrt{SNR}). \tag{5.10}$$

For the UL transmissions, Manchester coding is used. Manchester achieves the same BER but at a 3dB lower SNR [20, p. 502]:

$$BER_{man} = Q(\sqrt{2SNR}). \tag{5.11}$$

The UL signal is decoded using the RSIC signal $\rho[k]$. Therefore, the noise term in the SNR is the power of $\rho[k]$:

$$BER_{man,ul} = Q(\sqrt{\frac{2P_s}{P_\rho}}. \tag{5.12}$$

The FER is the probability of at least one bit error in one frame. Assuming bit errors are independent, the FER is calculated from the BER:

$$FER_{ul} = 1 - F(0, 8 \times L_{f,ul}, BER_{man}). \tag{5.13}$$

where $F(\dot{k}, \dot{n}, \dot{p})$ denotes binomial cumulative distribution function and $N_{f,ul}$ the length of the UL frame in bytes. In $F(\dot{k}, \dot{n}, \dot{p})$, $\dot{k}$ the maximum number of successes (bit errors), $\dot{n}$ the number of trials and $\dot{p}$ probability of success (bit error). $F(\dot{k}, \dot{n}, \dot{p})$ returns the probability of at most $\dot{k}$ successes in $\dot{n}$ trials with success probability $\dot{p}$ [23, p.114-120].

The FER calculation for the DL requires some additional steps since it involves a more complex PHY specification. We first calculate the byte error rate $ByteER_{4b6b}$ at the receiver after decoding of a 4b6b signal. The probability of a byte error is the probability of at least one bit error in the byte. The 4b6b decoding process is as follows. First, each sample in the received signal is individually compared to a threshold. The result is one bit for each sample. The first step has the same error rate as BER_{ook}. In the second step a group of six bits is decoded to four bits using a LUT mapping 6-bit words to 4-bit words. If at least one of the six individual threshold comparisons in the first step is wrong, the group of four decoded bits is wrong. A byte error occurs if at least one group of four bits in the byte is wrong. Hence, a byte error occurs if at least one of twelve individual binary threshold comparisons is wrong. Hence the byte error rate for a 4b6b signal is:

$$ByteER_{4b6b} = 1 - F(0, 12, BER_{ook}). \tag{5.14}$$

The RS FEC operates on symbols of GF(2^8) (see section 3.4.1.4). If more than one bit error occurs in a symbol (byte), the symbol is wrong. The payload RS(168,200) code can correct 16 bytes in every block of 200 transmitted bytes. For the DL we redefine the FER as the probability of a payload block error. Without header errors the block error rate is:

$$BlockER = 1 - F(16, 200, ByteER) \tag{5.15}$$

which is the probability of at least 17 byte errors in a block of 200 received bytes. The header uses a RS(6,12) code which is able to correct 3 bytes. Hence, the header error rate (HER) is:

$$HER = 1 - F(3, 12, ByteER). \tag{5.16}$$

The total DL FER (probability of payload block error) is:

$$FER_{dl} = HER + (1 - HER)BlockER. \tag{5.17}$$

In reality, the hardware receiver does not have a matched filter as is assumed in the above derivation. The bandpass filter in the receiver front-end distorts the received signal resulting higher BER. To get a more accurate estimate of the DL FER, we simulate the BER in Matlab using the training set used for LUT training in SIC (section 3.4.2.2). Simulated white, Gaussian noise is added to the received signal set to obtain the desired SNR. The BER is measured by decoding the received signal and checking if it matches with the transmitted signal in the training set. The number of bit errors are counted. We calculate the simulated DL FER ($FER_{dl,sim}$) from the simulated DL BER using equation 5.14 to 5.17.

5.1.4.2 Hardware Measurement

We measure the FER on the hardware setup for the DL and the UL channel depending on the horizontal distance d_{hor} between AP and user. The vertical distance between AP and user is 1.7m. The experiment consists of transmitting an audio stream on the DL. At the same time, during each DL packet the UD transmits four UL packets consisting of 32 payload bytes each using the full-duplex mechanism. The FER is measured in the Arduino software. For the DL channel we define the FER as $FER_{dl,meas} = \frac{\#RS_{fail}}{\#RS_{tot}}$. $\#RS_{fail}$ is the number of payload RS blocks unable to be decoded by the RS decoder. $\#RS_{tot}$ is the total number of RS blocks transmitted. In the experiment each DL packet contains four RS blocks. Per measurement at least 2000 DL packets (8000 RS blocks, 8000 UL packets) are transmitted.

For the DL channel we study three cases. In the first case we assume regular operation. In the second case we disable the CT sequence mechanism and the AP always transmits the full DL frame. In the third case we also disable the CCR mechanism (section 3.4.3.4) at the UD. For the UL, no CCR or CT sequence are used. Hence, we only study the UL under normal operation.

5.1.4.3 Results and Discussion

Figure 5.8 shows the measured FER versus the horizontal distance. On the DL failed frames are not retransmitted. Hence, an increase in the FER on the DL results in dropped audio frames and thus a decrease in audio quality. A FER above 5% makes the audio stream difficult to understand. As a result, the useful horizontal transmission range is 80cm with the CT mechanism and 130cm without the CT mechanism. In a future revision of the system the range with the CT mechanism can be increased by using an IR LED with a wider angle of half intensity ϕ_{ir} and/or higher radiant power $\tilde{P}_{ir}$. Enabling CCR on the DL lowers the FER from 7% to 0.05% at $d_{hor} < 1$m. Hence, CCR is essential for a satisfactory listening experience.

The UL has the ability to retransmit the same frame if a frame error occurs. Hence, a higher FER means more retransmissions which lowers the effective throughput. Best UL performance is obtained at $d_{hor} < 0.6$m. At high SNR (low horizontal distance), the UL FER is 2%. We can speculate about the possible causes for this relatively high FER. The first possibility is an unreliable UL preamble detection. The UL preamble is only two bytes long in order to reduce the UL frame detection time $t_{v,ul}$. A second possibility is that the UL also requires a CCR like the DL. In section 3.4.3.4 we calculate the maximum phase error due to clock drift. We conclude that bit errors due to clock drift on the UL are unlikely but nevertheless possible.

The relationship between horizontal distance and SNR is shown in figure 5.3. Figure 5.9 shows the theoretical, simulated and measured FER versus the SNR. For the DL, in an AWGN channel with no matched filter (simulation) a 0.8dB higher SNR is needed to achieve the same BER as in an AWGN with a matched filter (theory). For both the DL and the UL, our estimation closely matches the theoretical estimation and simulation results at low SNR levels ($<$11dB). The most probable reason why the hardware FER is not zero for high FER is the CCR mechanism. The CCR threshold parameters (Θ_a, Θ_v) are chosen for a minimal FER in a low SNR channel. As a result, when the SNR is high, a false positive may occur when the CCR switches to a new phase that is not better than the current one. Another possible cause is one or more errors during preamble detection.

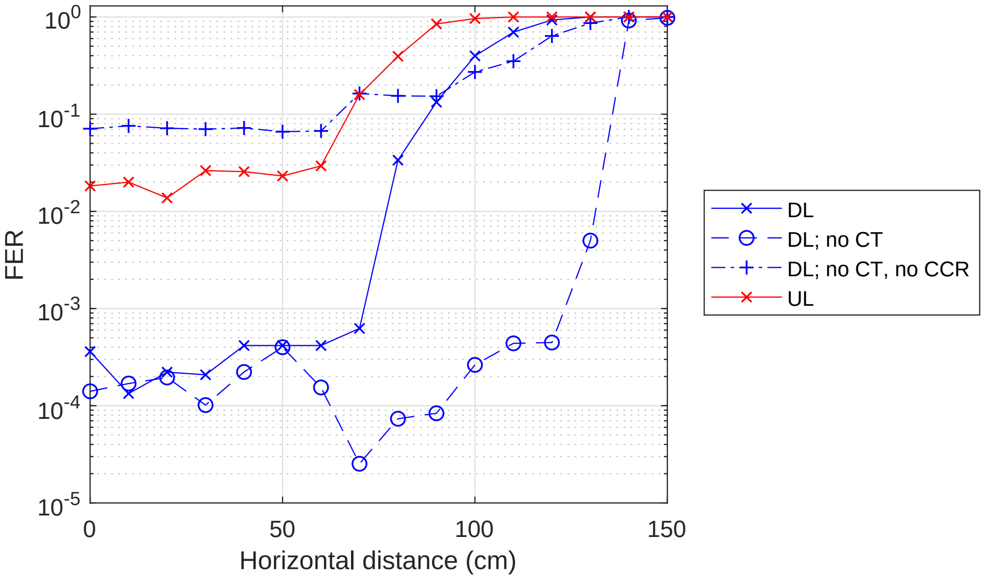


Figure 5.8: Measured frame error rate (FER) versus horizontal distance for the DL and UL channel.

5.1.5 Clear Channel Assessment and Collision Detection

As explained in chapter 3.4.3.1, CCA is performed by calculating the moving variance on the received signal (equation 3.16). We study the influence of the window length N_{cca} on the CCA classification accuracy. A low value of N_{cca} is preferred since the vulnerable period for the DL MAC protocol $t_{v,dl}$ is linearly dependent on N_{cca} (equation 3.18). Furthermore, an increase in the vulnerable period causes a decrease

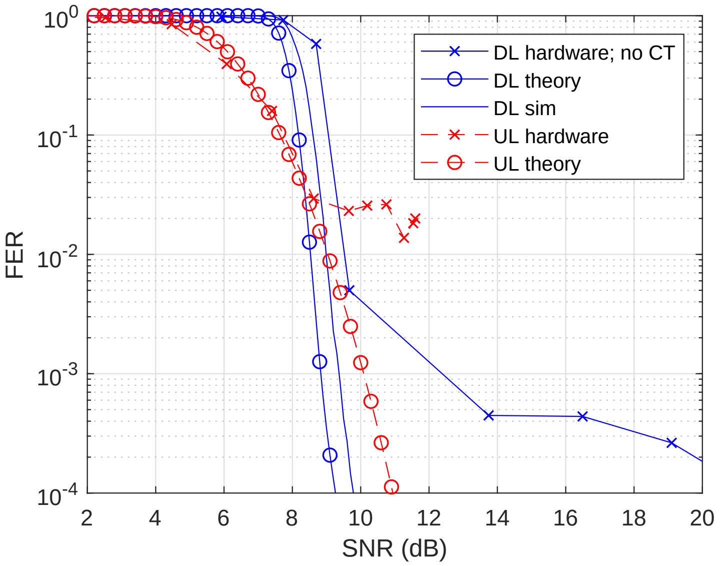


Figure 5.9: Measured, theoretical and simulated (only DL) FER versus SNR for the DL and UL channel.

in the throughput in a random access protocol (i.e. CSMA/CD) [61].

When $s[k]$ and $n[k]$ are both assumed to follow a Gaussian distribution, the PDF of the test statistic σ^2_{cca} at large N_{cca} can be approximated by Gaussian distribution [36]:

$$f_{\sigma^2_{cca}} = \begin{cases} \tilde{N}(\sigma^2_n, \frac{2\sigma^4_n}{N_{cca}}), & H_0 \\ \tilde{N}(\sigma^2_y, \frac{2\sigma^4_y}{N_{cca}}), & H_1 \end{cases} \tag{5.18}$$

where $\tilde{N}(\mu,\sigma^2)$ denotes a Gaussian distribution with mean μ and variance σ^2 and $\sigma^2_y = \sigma^2_s + \sigma^2_n = (1+SNR)\sigma^2_n$. We can conclude there is an inverse linear relationship between the variance of σ^2_{cca} and the length of the window N_{cca}.

For a detection threshold Θ_{cca} and assuming a large N_{cca}, the false positive probability p_{fp} and false negative probability p_{fn} are approximately:

$$p_{fp,cca} \approx \text{prob}[\sigma^2_{cca} > \Theta_{cca}|H_0] = Q(\sqrt{\frac{N_{cca}}{2}}\frac{\Theta_{cca}-\sigma^2_n}{\sigma^2_n}) \tag{5.19}$$

$$p_{fn,cca} \approx \text{prob}[\sigma^2_{cca} <= \Theta_{cca}|H_1] = 1 - Q(\sqrt{\frac{N_{cca}}{2}}\frac{\Theta_{cca}-\sigma^2_y}{\sigma^2_y}) \tag{5.20}$$

where Q is the Q-function [20, p.700,701].

The experiment consists of collecting a large amount of measurements of σ^2_{cca} on the Arduino. All measurements are done on AP_2 since AP_2 has the highest measured noise floor σ^2_n (table 5.1). From equation 5.19, p_{fp} is larger when σ^2_n is larger. Hence,

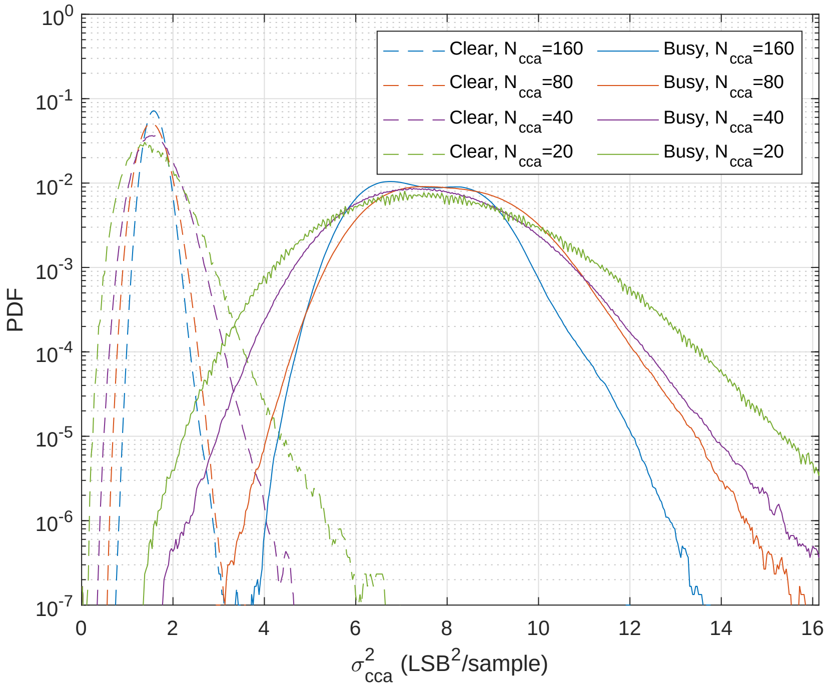


Figure 5.10: Log-PDF of the measured variance σ^2_{cca} in the CS window for different window lengths N_{cca}

if the CCA mechanism performs well on AP$_2$, it will also perform well on the other APs. The measurement proceeds on the setup in figure 4.3 and is performed first when the channel is clear. Next, the measurement is performed a second time when a neighbouring AP is continuously transmitting audio frames to simulate a busy channel. From figure 5.4, σ^2_y is approximately 8 in this setup.

Figure 5.11 shows the measurement results. It shows the log-PDF of the measured variance σ^2_{cca} when the channel is clear (dashed line) and when the channel is busy (solid line) for various values of N_{cca}. Classification errors occur if the PDFs of the clear channel measurements and busy channel measurements overlap. From the measurements this is the case when $N_{cca} = 20$ and when $N_{cca} = 40$. As expected, the variance of σ^2_{cca} decreases with increasing N_{cca}. As a result, when $N_{cca} = 80$ and when $N_{cca} = 160$ nearly no overlap occurs.

We choose a value of $N_{cca} = 80$ and set the classification threshold to $\Theta_{cca} = 2.625$. $N_{cca} = 80$ shows a good balance between classification accuracy and the DL CCA vulnerable period $t_{v,dl}$. For CCA, the cost of a false positive classification is lower than a false negative. A false positive results in an increase in the frame transmission delay. A false negative results in the AP transmitting while the channel is busy. This leads to a partial loss of the frame that was already being transmitted by another AP.

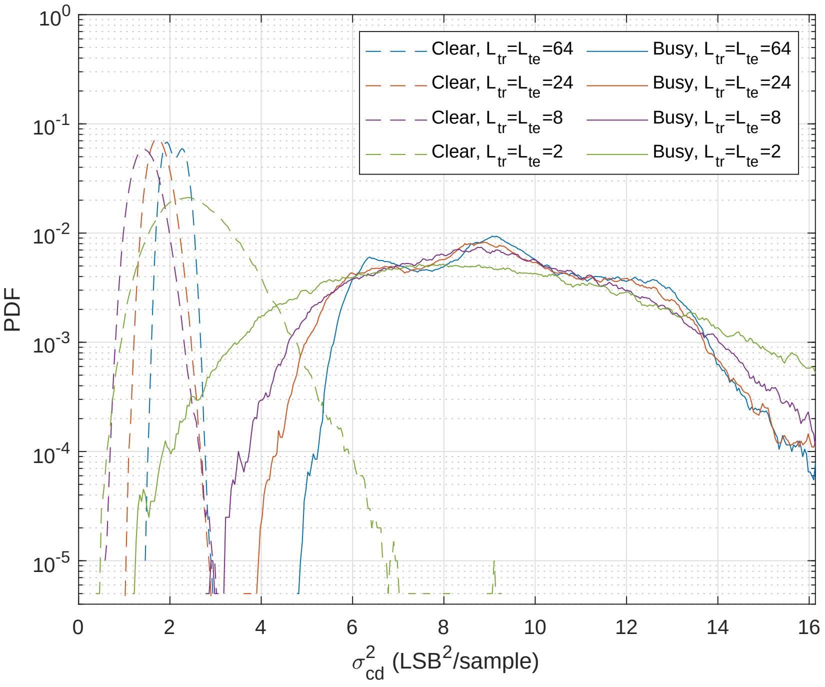


Figure 5.11: Log-PDF of the measured variance σ^2_{cd} in the CD test sequence for different values of L_{tr} and L_{te}

A threshold $\Theta_{cca} = 2.625$ results in a false positive rate of 0.01% and false negative rate of 0%.

Next, we study the classification accuracy of the CD mechanism (section 3.4.3.2) depending on the length of the TTP sequence. Briefly summarized, CD also relies on a variance measurement. The AP calculates the variance σ^2_{cd} in $\rho[k]$ while transmitting the test sequence at the start of the frame (equation 3.20). Following the same assumptions as CCA, the theoretical false positive and false negative probability for a detection threshold of Θ_{cd} are:

$$p_{fp} \approx \text{prob}[\sigma^2_{cd} > \Theta_{cd}|H_0] = Q(\sqrt{\frac{N_{te}}{2}}\frac{\Theta_{cd} - \sigma^2_\rho}{\sigma^2_\rho}) \tag{5.21}$$

$$p_{fn} \approx \text{prob}[\sigma^2_{cd} <= \Theta_{cd}|H_1] = 1 - Q(\sqrt{\frac{N_{te}}{2}}\frac{\Theta_{cd} - \sigma^2_y}{\sigma^2_y}) \tag{5.22}$$

where $\sigma^2_y = \sigma^2_s + \sigma^2_\rho = (1+\chi)\sigma^2_\rho$ and $s[k]$ is the received signal component from the transmission of a nearby AP. Choosing low N_{tr} and N_{te} reduces bandwidth overhead. The overhead caused by the TTP sequence is the primary reason why in Figure 5.13a the throughput S does not reach one.

In the CD experiment, AP$_2$ transmits a large number of frames. AP$_2$ is chosen since it has the highest P_ρ (see figure 5.6a). The calculated variance in each test sequence of each transmitted frame is logged. The experiment is done first when the channel is clear. Next, the experiment is repeated when a neighbouring AP is continuously transmitting a random Manchester encoded signal to simulate a collision.

Figure 5.10 shows the results of the experiment. Using these results we choose $L_{tr} = L_{te} = 12\text{byte} = N_{tr} = N_{te} = 240$ downlink symbols and a threshold of $\Theta_{cd} = 3.5$. No false classifications occur. The CD scheme requires three times the amount of samples compared to the CCA scheme to achieve high classification accuracy. This is probably because the sample rate of $\rho[k]$ is equal to the downlink OCR $\Omega_{dl} = 500kHz$ which is the same as the bit rate of the TTP sequence. Sampling at the same rate allows for simple software implementation. In future work, performance could be improved by using a test sequence at 250kHz. A similar technique to the preamble detection can then be used to select the best phase in the received test sequence. Solving this issue is left for future work.

5.2 MAC and Application Layer

We measure the performance of the DL and UL MAC layer and DL application layer. Since the hardware setup consists of only three APs and two users most of the experiments consist of simulations for a larger number of APs and users.

5.2.1 Downlink

5.2.1.1 MAC Protocol

We simulate the throughput and frame delay versus the offered load for the DL MAC protocol in Matlab. Our methodology and assumptions are the same as the analysis by Voulgaris et al. [69]. There is one difference though. Voulgaris et al. assume that if there is a frame in the buffer, it gets transmitted with a fixed probability. Our simulation on the other hand, implements the binary exponential backoff algorithm of the MAC protocol. This way we are able to study the influence of the binary exponential backoff parameters on the performance.

Time is slotted into slots of length $t_{slot} = 20\mu s$ or 10 DL symbols. At each transmitter, the user arrival rate follows a Bernoulli process [71]: in each slot a frame arrives from the application layer with probability g. We assume a buffer length of one frame at the transmitter's MAC layer. We define the offered load, G, as the average number of frame arrivals per frame time for all the APs. The relationship between G and g is:

$$G = \frac{ML_{dl}g}{t_{slot}\Omega_{dl}} \tag{5.23}$$

where M is the number of APs, N_{dl} is the frame length in number of optical clocks and Ω_{dl} is the DL OCR. The throughput S is defined as the fraction of time the channel is used for successful frame transmissions. The frame delay Δ is the time between frame arrival at the transmitter's MAC layer and start of successful frame transmission. The simulation's downlink vulnerable time period, frame length and TTP sequence length are identical to the hardware parameters listed in appendix C. Other assumptions for this simulation are:

- All APs are in range of each other. Hence, each AP can detect ongoing transmissions of all other APs.
- CCA with perfect accuracy unless stated otherwise.
- CD with perfect accuracy unless stated otherwise.
- The FER equals zero.

The first experiment aims to find optimal values of the DL MAC parameters `dlP`, `dlMacMinBE`, `dlMacMaxBE` and `dlMacMaxNB`. The parameter `dlP` is the probability of transmission in a p-persistent backoff strategy [60, p.267]. In the simulation the number of APs is nine as this is the expected maximum number of APs that can sense each other in a real world system. A situation with 9 APs colliding exists in a rectangular grid of APs where each AP can sense its immediate neighbours. In the simulation each AP is assigned a random CS and CD false positive rate up to 10% to simulate nearby noise sources or reflective objects. To find the optimal MAC parameters we perform a multi-objective optimization using the weighted sum method [45]. The objective function we aim to minimize is:

$$U(\vec{x}) = F_S^{norm}(\vec{x}) + 20 \times F_\delta^{norm}(\vec{x}) + F_J^{norm}(\vec{x}) \tag{5.24}$$

where

$$F_S(\vec{x}) = \sum_G (1 - S(G,\vec{x})) \tag{5.25}$$

$$F_\Delta(\vec{x}) = \sum_G \Delta(G,\vec{x}) \tag{5.26}$$

$$F_J(\vec{x}) = \sum_G (1 - J(G,\vec{x})) \tag{5.27}$$

$$F_i^{norm}(\vec{x}) = \frac{F_i(\vec{x}) - min_{\vec{x}}\{F_i(\vec{x})\}}{max_{\vec{x}}\{F_i(x)\} - min_{\vec{x}}\{F_i(\vec{x})\}}. \tag{5.28}$$

The vector $\vec{x} \in X$ is one vector from the set X of all possible design variable vectors. The design variables are `dlP`, `dlMacMinBE`, `dlMacMaxBE` and `dlMacMaxNB`. The objective function U consists of three cost functions. The first cost function, $F_S(\vec{x})$, increases when the throughput S is lower. S is simulated for a large of number of loads G and design variables, $\vec{x} \in X$. $F_S(\vec{x})$ is the sum of the costs for S over all the tested G. The second term $F_\Delta(\vec{x})$ is similar but applies a cost for a higher delay. The third term $F_J(\vec{x})$ applies a cost for lower fairness where J is the Jain's fairness index [72]. The Jain's index is a value between 0 and 1, and is higher the smaller

Protocol	dlP	macMinBE	macMaxBE	macMaxNB
IEEE CSMA/CA default	0	3	5	4
IEEE CSMA/CA optimal	0	4	9	5
non-IEEE CSMA/CA optimal	0	2	9	9
VLC-CSMA/CD optimal	0	0	4	9

Table 5.3: Best MAC parameters found from simulation according to the objective function in equation 5.24

the difference in throughput of the individual APs. Each term in U is normalized using the upper-lower-bound approach [45] in equation 5.28. $F_{\Delta}^{norm}(\vec{x})$ requires an additional normalizing weight of 20 in equation 5.24. To find the optimal design variables $\vec{x}^* = \text{argmin}_{\vec{x}} U(\vec{x})$, we repeat the simulation for a wide range of $\vec{x}$ and search for the minimum $U(\vec{x})$.

Additionally, using the same methodology we simulate the IEEE 802.15.7 CSMA/CA protocol [3]. We assume ACK messages in the IEEE protocol are instant and don't cause any bandwidth overhead. Furthermore, we assume an unslotted version of the protocol. We simulate the IEEE protocol first for the default protocol parameters [3, p. 190,191] shown in table 5.3. Additionally, we use the same objective function (equation 5.24) and find optimal MAC parameters $\vec{x}^*_{csma/ca}$.

The resulting optimal parameters, $\vec{x}^*_{csma/cd}$ and $\vec{x}^*_{csma/ca}$, are shown in table 5.3. The table shows that the weights of the three cost terms are chosen so the three costs are approximately equal for VLC-CSMA/CD. For the CSMA/CA protocol two parameter vectors are shown. The first parameter vector is allowed by the IEEE standard while for the second some of the parameters are outside of the allowed ranges in the IEEE standard. Table 5.4 shows the resulting values of each of the terms in the objective function for each protocol. For the CSMA/CA IEEE protocol there is a 54% reduction in the objective function when using the optimal allowed parameters instead of the default parameters. When using parameters outside the allowed range, a further reduction of 19% is achieved compared to the allowed optimal parameters. The optimal non-IEEE protocol trades better delay performance for slightly worse throughput compared the optimal IEEE-protocol. VLC-CSMA/CD achieves another reduction of 40% compared to the best CSMA/CA protocol. The reduction is mostly attributed to the increased throughput of VLC-CSMA/CD.

For further illustration, we show $S(G,x^*)$, and $\Delta(G,x^*)$ in figure 5.13a for both the VLC-CSMA/CD and CSMA/CA protocol for a simulation with 9 APs. The results show that VLC-CSMA/CD achieves a throughput improvement of 10% over the best CSMA/CA protocol at high loads. The CSMA/CA protocol with default parameters has a low throughput due to the high number of APs. The low default `dlMacMaxBE` and `dlMacMaxNB` values result in a high number of collisions at high loads. In CSMA/CA, if a frame collision occurs, the channel is wasted for the entire

Protocol	$F_S^{norm}(\vec{x})$	$20F_\Delta^{norm}(\vec{x})$	$F_J^{norm}(\vec{x})$	U
IEEE CSMA/CA default	2.8192	0.2985	0.2498	3.3675
IEEE CSMA/CA IEEE optimal	0.7627	0.6237	0.1600	1.5464
non-IEEE CSMA/CA optimal	0.9065	0.1749	0.1713	1.2527
VCSMA/CD optimal	0.3113	0.2412	0.2172	0.7692

Table 5.4: Results of objective function U obtained from simulation (lower is better)

length of the frame. In VLC-CSMA/CD on the other hand, collisions are detected at the beginning of the frame resulting in a more efficient channel occupation. VLC-CSMA/CD achieves similar performance results to wireless CSMA/CD studied by Voulgaris et al. [69].

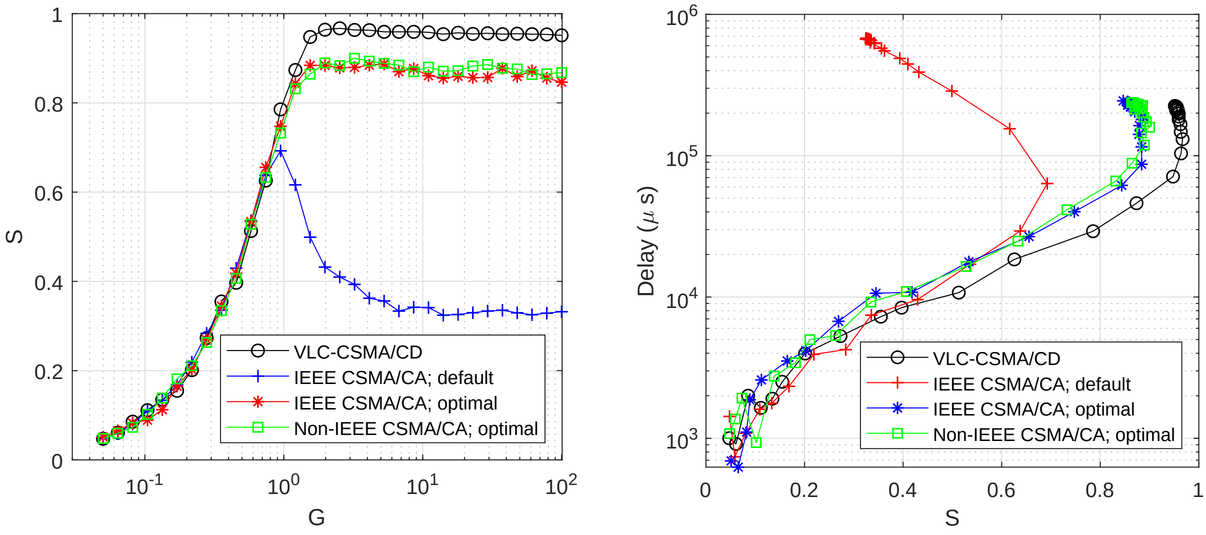
Next, we measure $S(G)$ and $\Delta(G)$ for a setup with 3 APs on the hardware setup (figure 4.3) and using the simulation. On the hardware setup we replicate the conditions of the simulation. Again, for each AP there is a probability of g that a new frame arrives per slot of 20μs. The throughput and frame delay are logged. During operation an oscilloscope is attached to the transmission pin of each AP. The oscilloscope is checked for correct operation of the MAC protocol. Correct operation is achieved when no two APs are transmitting a frame payload at the same time.

Figure 5.13 shows the result of the simulation and the hardware measurement for 3 APs. The throughput and delay curves are similar for both simulation and measurement. At high loads a throughput S of 96% is achieved in both the simulation and the hardware setup. The mean delay is never greater than 60ms which is sufficient for the LightTour application since DL frames of the same stream have to be transmitted every 320ms.

5.2.1.2 Throughput Simulation for Optimal Random Access MAC protocols

In this experiment we compare the throughput of some theoretically optimal random access protocols through simulation in a square grid of 16 APs as shown in figure 5.14. The horizontal distance between APs is $d_{hor} = 1.2$m as in the hardware setup. The transmission range d_{tx} is the maximum range for communication between AP and user. It is also the maximum range an AP can sense the transmission of a user in the CSMA/CA protocol. The sense range d_{sense} is the maximum distance between APs at which they are able to sense each other. We assume theoretical, simplified versions of RA MAC protocols that approach optimal versions of the protocols. In the simulation, time is slotted into slots of one frame length. In each slot a central coordinator randomly selects a set X of APs for transmission. The set X is selected subject to two constraints:

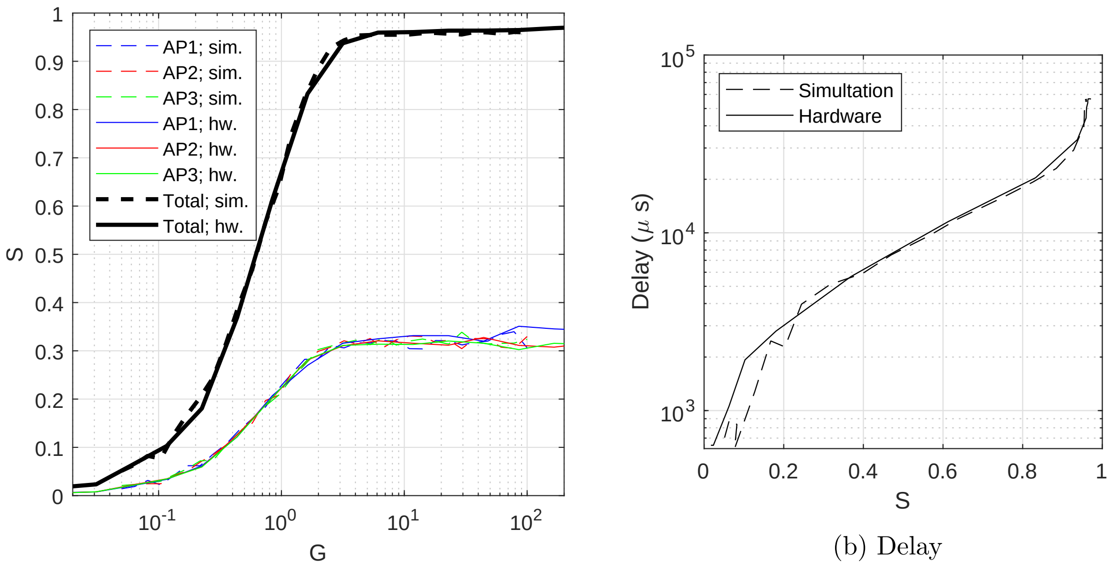
- Simultaneous transmissions of APs in X cannot collide.



(a) Throughput S

(b) Delay

Figure 5.12: Measured MAC throughput S and mean frame delay versus offered load (G) for 9 APs obtained from simulation for several protocols and MAC parameters



(b) Delay

(a) Throughput S

Figure 5.13: Measured MAC throughput S and mean frame delay versus offered load (G) for 3 APs obtained from hardware measurements (test) and simulation (sim) for VLC-CSMA/CD with optimal MAC parameters listed in table 5.3

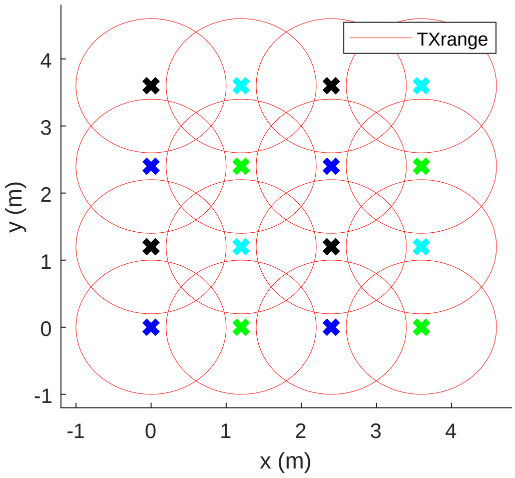


Figure 5.14: Room setup with a square grid of 16 APs (crosses). The AP colors denote the transmission groups in the TDMA protocol.

- For the unicast-CSMA and broadcast-CSMA protocols:

$$\forall AP_i,\ \forall AP_j \quad AP_i \in X \ \wedge\ dist(AP_i, AP_j) < R_{sense} \rightarrow AP_j \notin X \tag{5.29}$$

  The broadcast-CSMA protocol is similar to the DL MAC protocol in LightTour.

- For the unicast-CSMA/CA protocol:

$$\begin{aligned}\forall AP_i,\ \forall AP_j \quad & i \neq j \ \wedge\ isTransmitting(AP_i, U_i)\ \wedge \\ & [dist(U_i, AP_j) < d_{tx} \ \vee\ dist(AP_i, AP_j) < d_{sense}] \rightarrow AP_j \notin X.\end{aligned} \tag{5.30}$$

  This protocol is similar to the IEEE 802.15.7 MAC protocol [3].

- For the unicast-CA protocol:

$$\begin{aligned}\forall AP_i,\ \forall AP_j \quad & i \neq j \ \wedge\ isTransmitting(AP_i, U_i)\ \wedge \\ & dist(AP_i, AP_j) < d_{tx} \ \rightarrow\ AP_j \notin X.\end{aligned} \tag{5.31}$$

  This is a theoretical protocol in which APs do not sense the channel for transmissions of other APs. Instead APs can sense which users are currently receiving a transmission. An example of such a protocol is BTMA. The RTS/CTS mechanism in IEEE 802.11 [2] and floor acquisition multiple access (FAMA) [29] also provide APs with the information which users are currently receiving a transmission.

The primitive $dist(x,y)$ returns the Euclidean distance between x and y. The primitive $isTransmitting(x,y)$ is true if and only if x is transmitting to y.

- X is a maximal set. No extra AP can be added to X without breaking the applicable rule above.

The final protocol studied is time-division multiple access (TDMA) [18]. TDMA is not a random access protocol but is a reference to compare the other protocols. For the rectangular grid the APs are divided into four transmission groups as shown in figure 5.14. Each group is assigned one of four periodic slots. All APs in the group can only transmit during the group's assigned slot.

N users are distributed randomly within the room. We define the expected goodput $E[S_u]$ as the expected number of users in the room receiving a DL frame at any given instant under a random distribution of users. For each protocol, the simulation is ran for 20000 time slots. The users are randomly redistributed in space every 100 time slots. At the end of the simulation the average number of users served within a time slot (i.e. $E[S_u]$) is calculated. We run the simulation for APs with a narrow range ($d_{tx} = 1$m and $d_{sense} = 2$m), and with a wide range ($d_{tx} = 2$m) and $d_{sense} = 3$m). The narrow range corresponds with the ranges measured on the hardware setup.

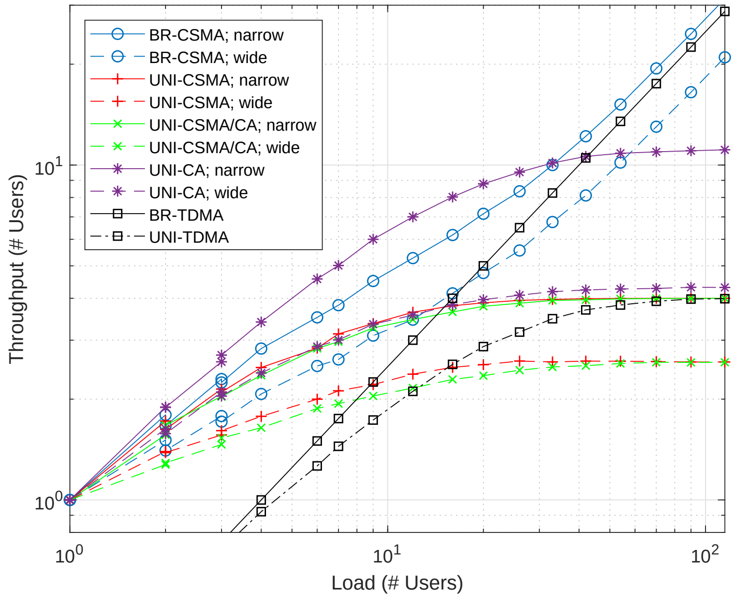


Figure 5.15: Expected total goodput ($E[S_u]$) versus the number of users in a room of 16 APs for several simplified protocols

Figure 5.15 shows the simulation results. The most important curves are the curve for the narrow range for BR-CSMA and UNI-CSMA/CA as those are the optimal versions of the protocols that correspond to our VLC-CSMA/CD protocol and the IEEE 802.15.7 CSMA/CA respectively. BR-CSMA always outperforms

UNI-CSMA/CA. At a high load the throughput of the unicast protocols plateaus since there exists an upper limit in a unicast protocol. UNI-CA outperforms BR-CSMA at low to medium loads (up to 33 users). This is expected since UNI-CA uses the CA mechanism to solve the exposed node problem [60, p.278-280]. Hence, by using CA neighbouring APs (APs that can sense each other) can sometimes transmit simultaneously depending on the location of the users they are serving. UNI-CSMA/CA has a lower throughput than UNI-CA due to the carrier sensing in UNI-CSMA/CA which results in the exposed terminal problem. All random access protocols outperform TDMA at low loads since in TDMA each AP has a fixed maximum rate (1/4 of the channel). In a random access protocol at low loads the APs are able to occupy a larger fraction of the channel since there are almost no collisions.

Finally, since APs with a narrow range outperform APs with a wide range, we can conclude that some calibration of d_{tx} and d_{sense} is necessary to achieve optimal throughput. This throughput decrease is due to less spatial reuse of the channel.

5.2.1.3 Application Layer Delay

A user can arrive at an AP that is already transmitting a stream to some other user. The arriving user has to wait until the start of a new stream before useful audio can be received. In this simulation, we measure the time between the moment a user arrives at an AP and the moment a new stream is started. Since only the application layer delay is studied, the simulation consists of one AP. Time is slotted into slots of 320ms which is the amount of compressed audio data in each frame payload. Users arrive randomly in each slot following a Poisson process with user arrival rate λ (users/slot) [23, p.131]. When no stream is available, users wait. When a stream is available, the AP starts broadcasting the stream to all users that are waiting. Stream broadcasting continues until the end of the stream is reached. In this simulation a stream lasts 150 frames. With each frame carrying 320ms (appendix C) of audio the stream length is thus 48s. The PHY layer provides a frame rate of R_f frames per slot. We simulate three protocols:

P1: Low rate parallel streams: each AP can have up to R_f parallel streams at any given time. Each stream is transmitted at a rate of one frame per slot. This protocol is used at the application layer of LightTour (section 3.2.1).

P2: High rate single stream: each AP can have only one stream at any given time. The AP transmits its stream at a rate of R_f frame per slot.

P3: Adaptive protocol: each AP can have up to R_f parallel streams at any given time. When the number of current streams is smaller than R_f, the AP can transmit some of its current streams at a higher rate so the total frame rate of the AP equals R_f.

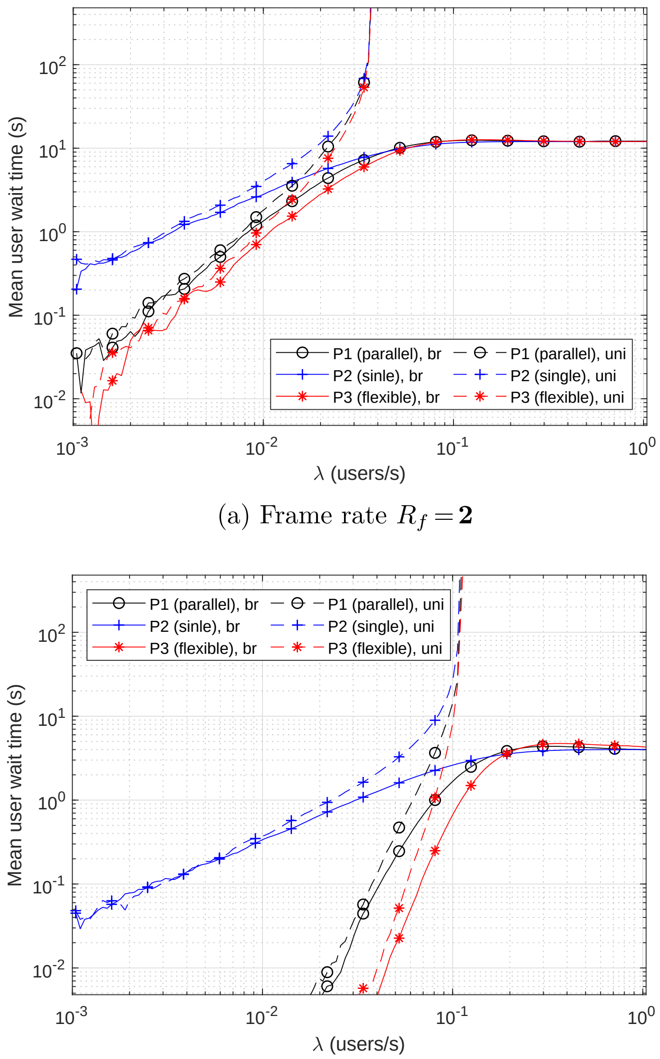
We simulate all three protocols in a broadcast and a unicast scenario. We perform the simulation for $R_f = 2$ and for $R_f = 6$ and for a wide range of values for λ. The results are shown in figure 5.16a and 5.16b. When broadcasting at high λ, all three protocols perform the same with a mean wait time of $\frac{1}{2R_f}$. For the unicast protocols the wait time goes to infinity when $\lambda > \frac{1}{24}sec$ for $R_f = 2$ and $\lambda > \frac{1}{8}sec$ for $R_f = 6$. Hence, more than R_f users arriving per stream results in an infinite wait time. At low λ, P1 outperforms P2. The reason for this is as follows. For P1 at low λ there is a high probability of a stream still being available when a user arrives since at least R_f users need to be at the AP in order for the user to wait. For P2 on the other hand, the user always has to wait if at least one other user is already present. The best performing protocol is P3. This is because P3 combines the best of P1 and P2. P3 allows streams to finish more quickly effectively reducing the expected wait time while also having the ability to immediately transmit another stream when a user arrives.

5.2.1.4 Complete Simulation of MAC and Application Layer

We simulate the complete DL MAC and application layer in a room with a 6x6 square grid of APs. AP spacing is 1.2m, $d_{sense} = 2$m and $d_{tx} = 1$ as in the hardware setup. For the MAC protocol, the methodology is the same as the simulation in section 5.2.1.1. The difference is that now APs can sense only other APs within d_{sense} and that frame arrival at the MAC buffer is controlled by an application layer. The application layer implements the LightTour application. For simplicity, we assume application layer protocol P2 from the previous section. Protocol P2 only allows one stream per AP at any time. The flow control algorithm (section 3.2.1.1) is also implemented. The `flowRate` parameter is variable. We assume that each user always listens for the entire duration of the stream.

Users are distributed randomly within the room. Users that are currently not receiving a stream follow a random Gaussian walk [75] through the room. Since it makes the simulation much simpler, the exact location of each user is only determined by the simulation once the user starts listening to an audio stream. For each AP, a variable stores the number of users that are waiting. Every second, for each AP, the waiting users are redistributed to neighbouring APs following a multivariate Gaussian distribution with $\vec{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\vec{\Sigma} = \begin{bmatrix} \sigma_x = 0.5 & 0 \\ 0 & \sigma_y = 0.5 \end{bmatrix}$. The values of σ_x and σ_y are chosen as it results in a probability of 68% a waiting user stays at an AP and 32% to move to some other AP. We believe these values represent a real-life situation of users that are waiting and moving.

The simulation is done for multiple values of N users. The length of each audio stream is 48s, the simulation time for each value of N is $4000s$. For the MAC protocol, we study our broadcast VLC-CSMA/CD protocol, the unicast IEEE 802.15.7 CSMA/CA protocol [3] and broadcast and unicast TDMA [18]. For VLC-CSMA/CD we use the optimal MAC paramaters in table 5.3. For IEEE CSMA/CA we use the optimal MAC parameters complying with the IEEE standard in table 5.3. For each AP we measure the total throughput S as the fraction of time the channel is



(a) Frame rate $R_f = \mathbf{2}$

(b) Frame rate $R_f = \mathbf{6}$

Figure 5.16: Result of application layer delay simulation; mean user wait time versus user arrival rate (λ) for application layer protocol P1, P2 and P3 and for broadcast and unicast

used for successful transmissions, the average user wait time in seconds, the channel utilization β which is the fraction of time the channel is used for any transmissions and the goodput. We define the goodput S_u as the throughput S multiplied by the average number of users per broadcast transmission. S_u is thus an indication for the number of users that are being served a packet at any given time. Finally, we calculate the channel efficiency η as:

$$\eta = \frac{S_u}{\beta} \tag{5.32}$$

The channel efficiency is also a measure of power efficiency since the power of the white LED is proportional to β(section 4.1.3.1).

Before we discuss the results we study the effect of the `flowRate` parameter in the traffic smoothing algorithm. Figure 5.17a shows the throughput per AP for all the protocols when `flowRate` is infinity (no traffic smoothing). In figure 5.17b, `flowRate=2.4`. Hence once the bucket is full, an AP can only transmit at 2.4 times the audio playback rate. Without traffic smoothing, throughput is very unfair. APs with fewer neighbours see fewer collisions and as a result don't have to increase their BE as often. This results in a higher probability of transmission for APs with few neighbours. This phenomenon is similar to the channel capture effect [49] but unlike the channel capture effect the unfairness is not temporary. Traffic smoothing with `flowRate=2.4` solves the problem since it limits the data rate of each AP and as a result prevents APs with few neighbours from capturing the channel. The value of `flowRate=2.4` is found experimentally and is the highest value that provides a reasonable fairness.

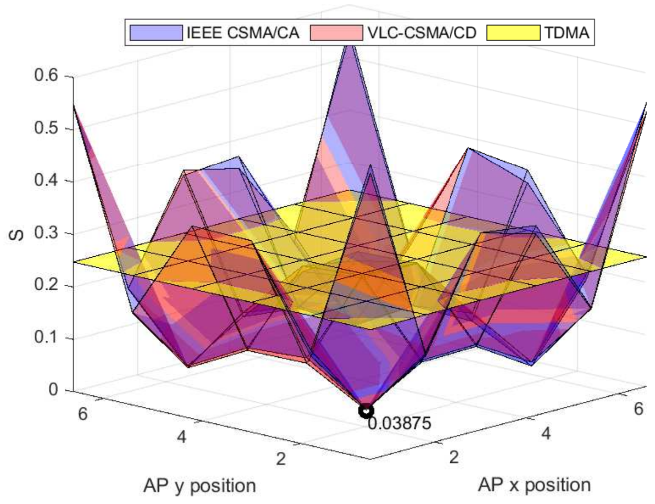
Figure 5.18a to 5.18d show the results of the simulation with `flowRate=2.4`. The goodput graph is similar to figure 5.15. The user wait time of the unicast and broadcast protocols is similar for fewer than 30 users. Above 30 users the broadcast protocols never exceed 10 seconds average user wait time while for unicast the wait time is an order of magnitude higher. The channel utilization curve shows that traffic smoothing limits the utilization. The channel efficiency curve shows that broadcast VLC-CSMA/CD gets increasingly more efficient at high loads.

5.2.2 Uplink

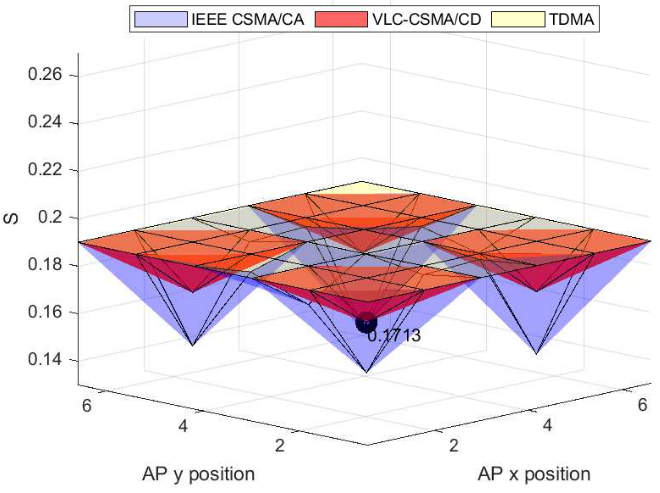
For the UL we simulate the MAC protocol (BSMA) throughput and frame delay in a similar way to section 5.2.1.1 for the DL. The simulation for the UL MAC protocol follows the same assumptions as the simulation for the DL MAC protocol. As explained in section 3.3.3.1, the UL MAC protocol can be modelled with a star topology. Therefore, this simulation considers one AP and N users that are in range of the AP. The AP continuously transmits DL frames. Each DL frame is transmitted immediately after the previous DL frame finishes transmission. We simulate the throughput and mean frame delay of the UL protocol for $N=5$ and for $N=15$ users.

We also simulate the BTMA protocol by repeating the simulation with an UL vulnerable period decreased by $t_{bsi}/2=40\mu$s where $t_{bsi}/2$ is the average increase in vulnerable period $t_{v,ul}$ added by the BS mechanism. On the hardware, the measured average UL vulnerable period for BSMA is $t_{v,ul}=200\mu$s. This time includes the time for ADC buffering and UL preamble detection.

The simulation results are shown in Figure 5.19a and Figure 5.19b. The p-persistent protocol with $p=0.2$ performs significantly better than the 1-persistent protocol at both $N=5$ and $N=15$. Since $N=5$ is a realistic value, we choose a value $p=0.2$ for the final system. At $p=0.2$, BTMA increased S up to 0.025 for N=5



(a) `flowRate=inf`

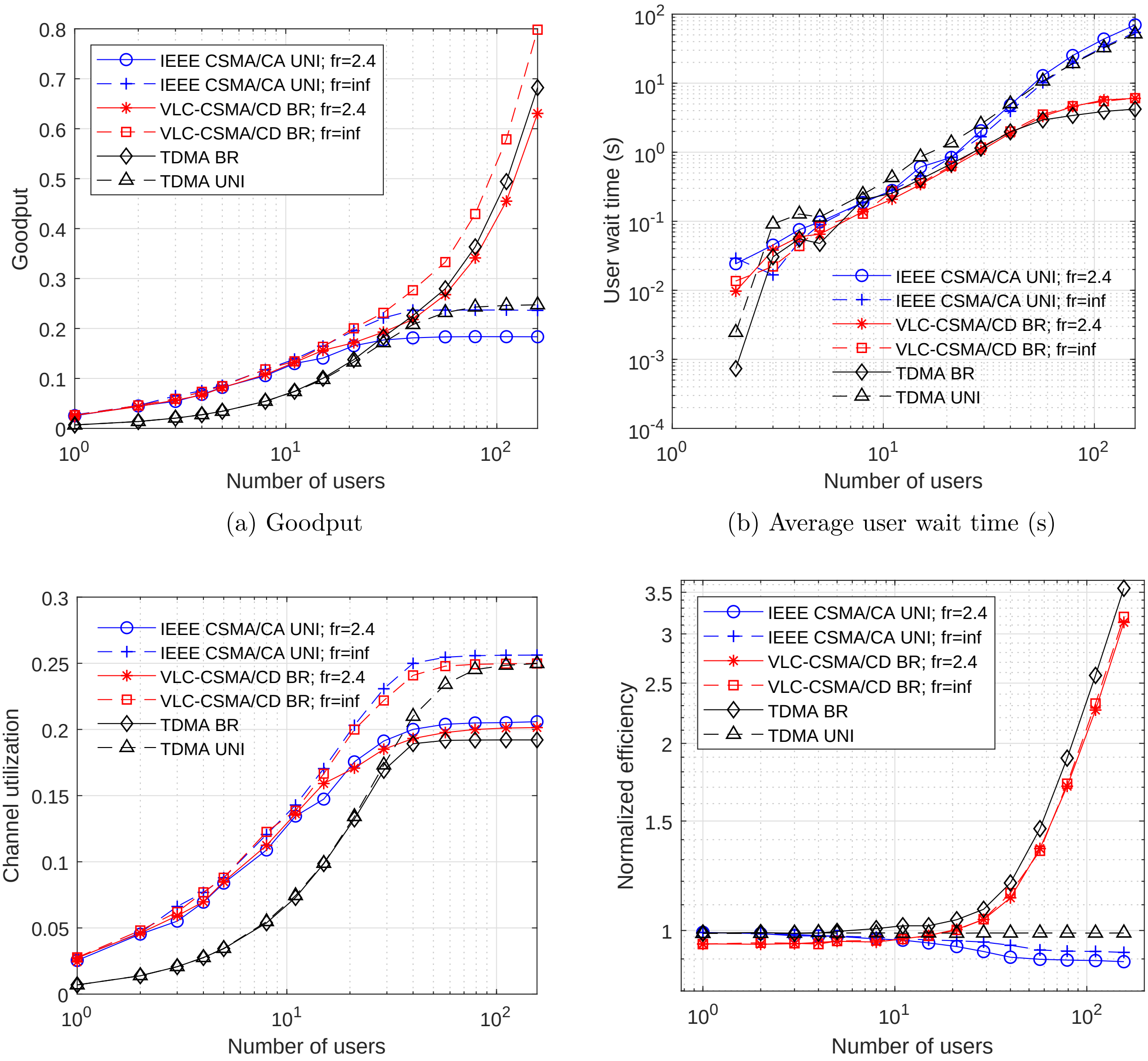


(b) `flowRate=2.4`

Figure 5.17: Application and MAC DL simulation measured throughput S per AP

and for N=15 compared to the BSMA protocol with $p=2$. We can conclude the BS mechanism has a not zero but almost negligible impact on the throughput.

In the UL protocol the BE at each user is only updated once per DL frame. The UL BE update algorithm is shown in section in algorithm 1. This makes the backoff algorithm different from normal exponential backoff where the BE is updated after every frame transmission. As a result, verification of the convergence speed and fairness of the new backoff algorithm is necessary. For this purpose we



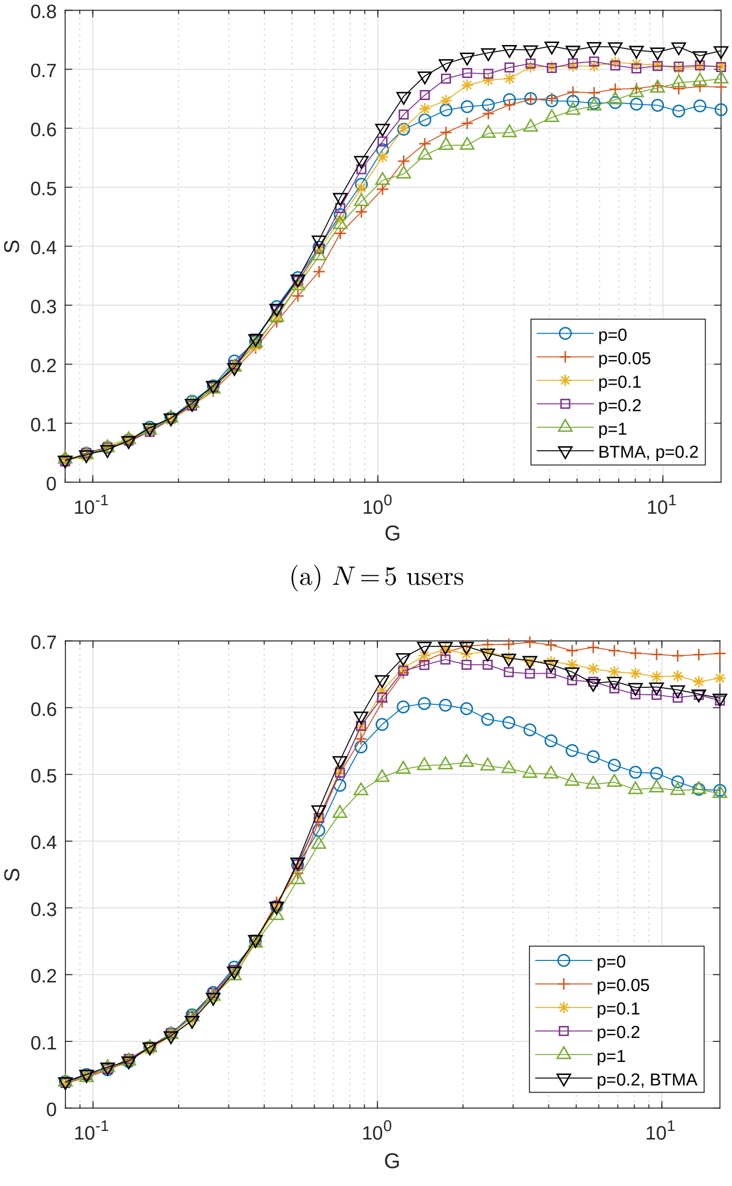
(a) Goodput

(b) Average user wait time (s)

(c) Channel utilization (%)

(d) Normalized efficiency

Figure 5.18: Measurement results of complete MAC and application layer simulation



(a) $N = 5$ users

(b) $N = 15$ users

Figure 5.19: Uplink MAC protocol simulation results: throughput S versus offered load G

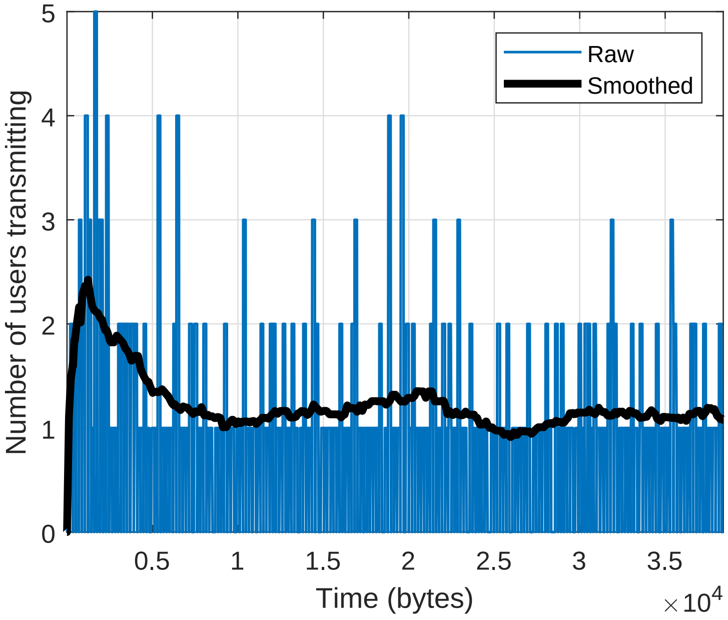


Figure 5.20: Time plot of the number of users transmitting on the UL channel in an simulation with N=15, p=0.2 and G=64.

show a time plot of the simulated channel for $N=15$, $p=0.2$ and $G=64$ in figure Figure 5.20. The time plot illustrates that the UL throughput converges after approximately two DL frames. The throughput per user is also measured. The Jain's fairness index [72] is calculated on the set of throughputs per users and is found to be 0.9955. Hence, from the simulation we can conclude that the backoff algorithm is fair.

5.3 Conclusion

This chapter has provided a detailed study on the performance of the system which was designed and implemented in the previous two chapters. Signal strength measurements on the hardware show that the self-interference can not be ignored as assumed in related work. The reason for this is the small radiant flux of the IR LED at the UD compared to the white LED at the AP.

We conducted a large amount of measurements to study the performance of our SIC mechanism. We find that SIC is dependent on the line code because of the high pass filter at the receiver front-end. At low self-interference levels, P_ρ close to the noise floor is achieved while at high self-interference levels, a cancellation of $\chi = 30dB$ is achieved. Additionally, we find that high DC light interference has little effect on the SIC performance other than increasing the noise floor. Measurements of SIC performance at all three APs indicate that performance can be increased if the LUTs are trained for each AP separately.

The FER is studied using measurements and theoretical calculations. For the downlink our results show the hardware implementation requires an increase of 1dB in SNR at low SNR levels. At high SNR levels we measure an FER floor of $\sim 10^{-4}$. Although not proven, we presume this is due to the imperfect CCR. Furthermore, our results show that the novel CCR mechanism is able to reduce FER from 7% to 2×10^{-4}. The uplink shows a FER floor of 2% which we presume is caused by the fact that no CCR is implemented at the AP.

CCA and CD classification accuracy are measured on the hardware. We find a near-perfect classification accuracy for both at an inter-AP spacing of 1.2m and window lengths of $L_{tr} = 80$ and $L_{te} = 240$ respectively. We speculate the increase in window length required by CD is due to the sampling rate of fact the sample rate and symbol rate of $\rho[k]$ are equal. We propose a solution that can be implemented in future work based on blind oversampling clock recovery which could lower the minimum requirement of L_{te}.

Through Monte Carlo simulations of the downlink MAC protocol we find optimal value of the downlink MAC parameters. We compare our protocol to the IEEE CSMA/CA standard and conclude a 10% improvement in throughput is achieved under high loads. The Monte Carlo simulations are verified by hardware measurements on the small scale setup. Another simulation measures the goodput of theoretically optimal contention based protocols in a typical room setup of APs. The simulation shows that a broadcast CSMA protocol significantly outperforms unicast CSMA/CA. For example at a load of 2 users/m^2 the CSMA/CD achieves a goodput three times higher than CSMA/CA.

Three different downlink application layer strategies are simulated though Monte Carlo simulations for a setup with one AP. It is found that multiple parallel audio streams transmitted at a lower rate leads to a decrease in mean user wait time compared to transmitting a single audio stream at a high rate. For example when users arrive on average every 50s at the AP, wait time is decreased from $6s$ with two parallel streams compared to $3.2s$ for a single parallel stream at twice the stream rate. Furthermore, this simulation again shows the benefit of broadcast compared to unicast at high loads. The wait time in a broadcast protocol is bounded while for unicast the wait time goes to infinity at a certain user arrival rate.

A final Monte Carlo simulation for the downlink simulates both the MAC and application layer of LightTour in a typical room setup. Similar conclusions can be made as the simulations which measure the performance for the MAC and application layer separately. Additionally, this simulation shows the need for congestion control in VLC-CSMA/CD if fairness is desired at high loads.

Finally, the uplink MAC protocol (BSMA) is simulated using a similar method as the downlink MAC protocol. A bandwidth efficiency of 70% is obtained at high loads. The overhead caused by the busy slot interval is measured as being at most 2.5%.

Chapter 6

Conclusion and Future Work

In this thesis a museum audio tour system based on VLC is proposed, designed, implemented and tested. Modules for APs and UDs are built based on Arduino Due microcontrollers. The downlink MAC protocol is based on a low time-complexity SIC-mechanism specifically designed for VLC. The SIC mechanism uses a LUT to decrease computation time by 90% over conventional methods at a cost of $2kB$ or RAM. The SIC mechanism achieves a cancellation down to the noise floor under normal conditions and $30dB$ of cancellation at high self-interference levels. Currently, the biggest drawback of the SIC mechanism is that LUTs have to be trained offline. We showed that up to $3.5db$ improved performance can be obtained if the LUT could be trained online.

Using the SIC mechanism, a high accurate CD mechanism and bandwidth efficient CSMA/CD protocol are implemented. Our hardware measurements and simulation indicate an increase in bandwidth efficiency of 10% compared to IEEE 802.15.7 CSMA/CA. In combination with the application layer, the broadcast VLC-CSMA/CD behaves similarly to the unicast IEEE CSMA/CA at low loads. At high loads, the proposed VLC-CSMA/CD protocol significantly outperforms the CSMA/CA protocol in goodput (3x greater), bandwidth efficiency (3x greater) and average user wait time (8x smaller). With simulation we show that to provide fairness, a congestion control mechanism is required. A simple leaky-bucket algorithm is implemented at the AP downlink application layer to provide both flow control and congestion control.

Additionally, the SIC mechanism allows full-duplex transmissions. A novel full-duplex random access protocol which we call BSMA is developed as an adaptation of BTMA. In combination with the DL protocol, it solves the hidden user problem while minimizing overhead on the DL channel. Busy slots are interleaved with DL payload bits and provide the same functionality as busy tone in BTMA without requiring additional hardware complexity. Users transmit using a random access algorithm during the DL payload and UL frames are acknowledged with an A-ACK at the end of the DL frame. Our simulations indicate a bandwidth efficiency of 70% at high loads.

A novel full duplex CT sequence mechanism enables APs to detect if there are nearby users that want audio data. The CT sequence can reduce unnecessary transmissions by 99.2 and power consumption of the white LED by 17%.

On the testbed, we find CCR is necessary to achieve low FER. A novel low-complexity software CCR mechanism is implemented and tested. The CCR mechanism uses the start and stop bits, inserted into the data stream by USART, to detect the best sampling phase at the receiver. At high SNR the CCR mechanism reduces the FER from 7% to 0.02%. A future study can compare the performance of the software CCR to a hardware PLL.

Future Research Opportunities

There are still many directions for future research on LightTour. We briefly mention some suggestions.

PHY Improvements Our measurements show that SIC performance is improved when the LUT is trained for each AP separately. In a future version LUT training can be implemented on the APs and UDs as part of the PHY layer. In addition, adapting the LUT mechanism to more bandwidth efficient modulation schemes such as PAM other than OOK could greatly increase PHY data rates.
CCA and CD are implemented using energy detection. Moreover, RF systems have already seen extensive research into collision detection based on SIC. Better techniques might exist that can provide improved classification accuracy for VLC CD [66]. As a result, the robustness of the system against poor SIC performance and noise can be improved.

Advanced Positioning In the current system each AP can only send one audio track because the AP can only determine the presence of the user but not its positions. LightTour can be extended with advanced visible light positioning (VLP) methods which allow for a higher resolution user position estimation. This position estimation can be used to decouple AP locations from audio track locations.

User mobility The LightTour system does not provide user mobility at this time. User mobility would benefit the system greatly since it allows users to walk around while listening to the same audio track.

Advanced congestion control Our proposed system only allows a fixed number of streams per AP. More complex load-balancing protocols can be implemented that can adapt the number of parallel audio streams at the AP.

A more flexible implementation The current software implementation is efficient but not very portable to other microcontrollers because many SAM3X-specific functionalities were used to increase performance. A system using a portable RTOS can provide the solution.

Further developing BSMA No extensive optimization of the uplink protocol parameters has been done. Moreover a possible way to improve the protocol is by using the busy slot as a way of collision detection on the uplink. If the UD does not see a change in the busy slot after it starts transmitting, an uplink collision has probably occurred. Further developing this idea is left for future research.

A look into other applications The protocols developed in this theses were specifically designed for the audio tour application but can probably also be used in other applications. The downlink MAC protocol (VLC-CSMA/CD) can be used for applications that requires efficient broadcasting using uncoordinated APs. The uplink MAC protocol (BSMA) provides a low-complexity way of solving the hidden node problem an is therefore a possible candidate to be used in many systems.

Appendices

Appendix A

Leaky-bucket algorithm

Algorithm 4: Audio stream flow control at the application layer

```
Parameters: frameSize, //$\mu s$ of audio in each DL frame
            bucketSize, //bucket size in $\mu s$
            IFS, // interframe space
            flowRate // rate of emptying of the bucket as a
                  //multiple of audio playback rate
bucketFullness=0;
timeLastTxEnd=0;
while 1 do
    timeNow = micros();
    timeExpired = timeNow - timeLastTxEnd;
    bucketCheck = bucketFullness - flowRate * timeExpired;
    if bucketCheck < (bucketSize-frameSize) and timeExpired > IFS
     then
        // transmit() returns the time at which the transmission finishes
        timeTXEnd=transmit();
        bucketFullness += frameSize - (timeTxEnd - timeLastTxEnd);
        timeLastTxEnd = timeTxEnd;
    end
end
```

Appendix B

Derivations

B.1 Overhead of Uplink on Downlink

The UL MAC protocol causes some additional bandwidth overhead on the DL channel. In particular, there are two sources of overhead. The first source of overhead are the BSes (3.4.1.3):

$$O_{bs} = \frac{l_{bs}}{l_{bsi}} = \frac{2T_{dl}}{40T_{dl}} = 5\% \tag{B.1}$$

where l_{bs} is the length of the BS, l_{bsi} is the length of the BS interval and T_{dl} is the downlink symbol period. An important note is that this overhead is independent of the number of UDs transmitting on the UL channel.

The second source of overhead is the A-ACK message at the end of a DL frame. Figure 3.4 shows the contents of the A-ACK message. We calculate the length of the A-ACK message in seconds:

$$t_{aack} = (4 + 2\ \#UL) \times O_{4b6b} \times O_{usart} \times 8bit/byte \times \frac{1}{fo,dl} \tag{B.2}$$

where $\#UL$ is the number of UL frames correctly received by the AP during the DL frame, $O_{4b6b} = 6/4$ is the overhead of the 4b6b line code, $f_{o,dl} = 500kHz$ is the OCR of the DL. $O_{usart} = 10/8$ is the overhead of universal synchronous/asynchronous receiver transmitter (USART) which is the hardware module that modulates the transmitted signal. Additionaly, if we assume DL and UL frame lengths listed in appendix C:

$$\max \#UL = \lfloor \frac{t_{payload,dl - t_{ct} - 320\mu s}}{t_{ul}} \rfloor = \lfloor \frac{24.28ms}{3.2ms} \rfloor = 7 \tag{B.3}$$

where $t_{payload,dl}$ and t_{ul} and t_{ct} are the DL payload and UL length and the CT sequence deadline. At $320\mu s$ before the end the DL frame the A-ACK is assembled. Hence no UL transmissions are allowed during that time. The maximum overhead caused by the UL channel on the DL transmissions is thus:

$$\max O_{ULonDL} = \frac{\max t_{aack}}{t_{dl}} + O_{bs} = \frac{0.54ms}{27ms} \times 100\% + 5\% = 7\%. \tag{B.4}$$

where t_{dl} is the time of a complete DL frame transmission.

B.2 Derivation Simplified SIC Algorithm

The system model is:

$$y[k] = H \cdot x[k] + V + n[k] \tag{B.5}$$

H and V are found from the following minimization problem which minimizes the sum of squared noise terms [78]:

$$\text{Find } \min_{H,V} U(H,V), \quad \text{for } U(H,V) = \sum_{k=1}^{N_{tr}} n^2[k] = \sum_{k=1}^{N_{tr}} (y[k=0] - V - H \cdot x[k])^2. \tag{B.6}$$

This problem has the following solution:

$$\hat{H} = \frac{\sum_{k=1}^{N_{tr}} (x[k] - \bar{x})(y[k] - \bar{y})}{\sum_{k=1}^{N_{tr}} (x[k] - \bar{x})^2}, \tag{B.7}$$

$$\hat{V} = \bar{y} - \hat{H}\bar{x}. \tag{B.8}$$

where $\bar{x}$ and $\bar{y}$ denotes the sample mean of $x[k]$ and $y[k]$ during the training sequence.

Equation B.7 can be simplified since $\bar{x} = 0$ which gives:

$$\hat{H} = \frac{\sum_{k=1}^{N_{tr}} x[k](y[k] - \bar{y})}{N}. \tag{B.9}$$

Since the number of times $x[k] = -1$ and the number of times $x[k] = +1$ are both $\frac{N}{2}$ we can further simplify the above equation:

$$\hat{H} = \frac{\sum_{k=1}^{N_{tr}/2} 1 \cdot (y^p[k] - \bar{y}) + \sum_{k=1}^{N_{tr}/2} -1 \cdot (y^n[k] - \bar{y})}{N_{tr}} \tag{B.10}$$

$$= \frac{\sum_{k=1}^{N_{tr}} \cdot (y^p[k]) - \sum_{k=0}^{N_{tr}} \cdot (y^n[k])}{N_{tr}} \tag{B.11}$$

$$= \frac{\bar{y^p} - \bar{y^n}}{2}. \tag{B.12}$$

Where:

$$y^n[k] = \begin{cases} y[k] & \text{when } x[k] = -1 \\ 0 & \text{when } x[k] = 1 \end{cases}$$

and

$$y^p[k] = \begin{cases} y[k] & \text{when } x[k] = 1 \\ 0 & \text{when } x[k] = -1. \end{cases}$$

The means μ^n and μ^p are mean of $y[k]$ when $x[k] = -1$ and when $x[k] = 1$ respectively. Equation B.8 can be rewritten as:

$$\hat{V} = \bar{y} = \frac{\bar{y}^p + \bar{y}^n}{2}. \tag{B.13}$$

Plugging B.12 and B.13 in B.5 gives:

$$\hat{y}[k] = \hat{H} \cdot x[k] + \hat{V} = \frac{\bar{y}^p - \bar{y}^n}{2} \cdot x[k] + \frac{\bar{y}^p + \bar{y}^n}{2}. \tag{B.14}$$

Which gives:

$$\hat{y}[k] = \begin{cases} \bar{y}^p & \text{when } x[k] = 1 \\ \bar{y}^n & \text{when } x[k] = -1. \end{cases}$$

During the transmission of the frame payload the AP estimates the clean channel $\rho[k]$:

$$\rho[k] = y[k] - \hat{y}[k]. \tag{B.15}$$

where $\rho[k]$ is the residual self interference (RSIC) after cancellation. Estimating H and V requires only 1 addition per sample in the training sequence.

B.3 Data Rate

The data rate of the DL and UL channel is shown in table 3.1. The calculation for the DL data rate is as follows. The DL frame payload carries $L_{data,dl} = 672$ bytes of useful data. The transmission time of the payload is:

$$t_{payload,dl} = L_{data,dl} O_{rs} O_{4b6b} O_{bs} O_{usart} \times 8bit/byte \frac{1}{t_{o,dl}} \tag{B.16}$$

where O_x indicates the overhead of component x of the PHY layer. Filling in the parameters gives:

$$t_{payload,dl} = 672\ byte \frac{200}{168} \frac{6}{4} \frac{32}{30} \frac{10}{8} \times 8\ bit/byte \frac{1}{500kHz} = 25.6ms. \tag{B.17}$$

The DL payload data rate is thus:

$$R_{dl} = \frac{L_{data,dl}}{t_{payload,dl}} 8bit/byte = 210kb/s \tag{B.18}$$

Similarly for the UL:

$$T_{payload,ul} = L_{data,ul} O_{Manchester} O_{usart} \times 8\ bit/byte \frac{1}{t_{o,ul}} \tag{B.19}$$

$$= 64\ byte \times 2 \times \frac{10}{8} \times\ 8bit/byte \times \frac{1}{250kHz} = 5.12ms \tag{B.20}$$

$$R_{ul} = \frac{L_{data,ul}}{t_{payload,ul}} \times 8\ bit/byte = 100kb/s. \tag{B.21}$$

However, since a UD is only allowed to transmit to an AP while the AP is transmitting a DL payload to the UD (see section 3.3.3.1), the maximum effective UL PHY data rate for a single UD is:

$$R_{ul,ue} = \frac{R_{audio}}{R_{dl}} R_{ul} = \frac{16.8kb/s}{210kb/s} \times 100kb/s = 8kb/s \tag{B.22}$$

where R_{audio} is the audio bit rate of the DL application layer.

The total UL data rate to a single AP is dependent on the fraction of time the AP successfully transmits on the DL channel. In section 5.2.1.4 we find that under high loads each AP has a maximum throughput of ∼18% of the total bandwidth of the DL channel (figure 5.17b). Hence, the maximum effective PHY UL data rate to a single AP is:

$$R_{ul,ap} = 0.18 R_{ul} = 18kb/s \tag{B.23}$$

The approximate maximum UL data rate for the UL application layer to a single AP is:

$$R_{ulapp,ap} = \frac{R_{ul,ap} \times S_{ul}}{O_{ul,frame} O_{ct}} = 18kb/s \times 0.7 \times \frac{32}{39} \times \frac{24.6}{25.6} = 9.93kb/s \tag{B.24}$$

where S_{ul} is the maximum throughput under high loads of the UL MAC layer (section 5.2.2, $O_{ul,frame}$ is the overhead caused by the UL frame header and payload CRC and O_{ct} is the overhead caused by the CT sequence.

Appendix C

System Parameters

Parameter	Value	Unit	Description
Topology			
d_{hor}	1.2	m	horizontal distance between APs in hardware setup and simulations
d_{ver}	1.7	m	vertical distance between APs and floor in hardware setup
AP transmitter front-end			
$\phi_{w,1/2}$	15	°	white LED half power semi-angle
$I_{w,high}$	900	mA	current draw when transmitting high symbol
$P_{w,comm}$	3.04	W	average power consumption in communication mode
$P_{w,illum}$	2.51	W	power consumption when in illumination mode
$\Phi_{w,high}$	$\sim$300	lumen	luminous flux at $I_{w,high}$
IR transmitter front-end			
$\phi_{ir,1/2}$	19	°	IR LED half power semi-angle
$I_{ir,high}$	100	mA	current draw when transmitting high symbol
$P_{ir,comm}$	3.04	W	average power consumption in communication mode
B	7	MHz	ir TX front-end bandwidth
$\tilde{P}ir, high$	55	mW	radiant power at $I_{uetx,max}$
Receiver front-end			
$f_{c,1}$	7.2	kHz	high pass filter cut-off frequency
$f_{c,2}$	695	kHz	low pass filter cut-off frequency
λp	900	nm	photodiode peak sensitivity wavelength
γ_{pd}	0.64	A/W	photodiode peak sensitivity
PHY			
f_{adc}	1	MHz	ADC sample rate
Ω_{dl}	500	kHz	downlink optical clock rate
Ω_{ul}	250	kHz	uplink optical clock rate
T_{adc}	1	μs	ADC sample period

T_{dl}	2 μs		downlink symbol period
T_{ul}	4	μs	uplink symbol period
payloadRSn	200	byte	payload Reed-Solomon block size
payloadRSk	168	byte	payload Reed-Solomon message size
headerRSn	12	byte	header Reed-Solomon block size
headerRSk	6	byte	header Reed-Solomon message size
l_{bsi}	40	T_{dl}	busy slot interval
l_{ct}	40	T_{ul}	length of CT sequence
dlPreambleLengt	5	byte	length of downlink preamble
ulPreambleLengt	2	byte	length of uplink preamble
N_{CCA}	80	T_{adc}	length of CCA window
t_{ct}	480	μs	CT detection deadline
$t_{v,dl}$	80	μs	vulnerable period for downlink CCA
$t_{v,ul}$	200	μs	vulnerable period uplink: maximum delay between start of uplink transmission and busy slot update
$t_{f,dl}$ 27	ms		DL frame length
$t_{f,ul}$ 3.2	ms		DL frame length
Θ_{cca}	2.625	$\frac{LSB^2}{sample}$	CCA detection threshold
L_{tr}	12	byte	length of training sequence before manchester encoding
L_{te}	12	byte	length of test sequence before manchester encoding
L_{pad}	4	byte	length of test sequence before manchester encoding
Θ_{cd}	3.5	$\frac{LSB^2}{sample}$	CD threshold
$\Theta_{ccr,a}$	0.5	$\frac{LSB}{sample}$	CCR EMA hysteresis threshold
$\Theta_{ccr,v}$	3		CCR EMV hysteresis threshold
α	1/32		weight in EMA/EMV update
Θ_{pr}	5	$\frac{LSB}{sample}$	preamble detection threshold
Γ	11		LUT order
MAC			
ulFIFOsize	6		size of the UL MAC frame buffer (Q_{ul}) at the user
dlMacMinBE	0		downlink minimum backoff exponent
dlMacMaxBE	4		downlink maximum backoff exponent
dlMacMaxNB	9		downlink maximum number of backoffs
dlMacMax-BackoffTime	640	ms	downlink maximum time attempting to transmit a frame before the frame is dropped
p (ul)	0.2		probability in uplink p-persistent backoff strategy
ulMacMinBE	3		uplink minimum backoff exponent
ulMacMaxBE	9		uplink maximum backoff exponent
ulMacMaxNB	7		uplink maximum number of backoffs
$t_{ubp,dl}$	160	μs	unit backoff period of the downlink MAC protocol
$t_{ubp,ul}$	200	μs	unit backoff period of the uplink MAC protocol
Application			
payloadRSblocks	4		number of RS blocks in downlink payload

speexFrameSize	42	byte	length of Speex frame containing 20ms of compressed audio
speexFramesPerRSblock	4		number of speex frames per payload RS block
bucketSize	4.8	s	downlink traffic smoothing algorithm bucket size
flowRate	1		downlink traffic smoothing algorithm flow rate
dlIFS	112	ms	downlink inter frame space
t_{probe}	1	s	time between downlink probing frames when the AP detects no users
noCTthreshold	3		number of consecutive downlink frames without a CT sequence reply before the stream is ended
maxNbStreams	3		maximum number of parallel audio streams on the hardware setup
Θ_{loc}	2		location hysteresis threshold for determining the closest AP

Bibliography

[1] Ieee standard for local and metropolitan area networks–part 15.7: Short-range wireless optical communication using visible light. *IEEE Std 802.15.7-2011*, pages 1–309, Sep. 2011.

[2] Ieee standard for information technology-telecommunications and information exchange between systems local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, Dec 2016.

[3] Ieee standard for local and metropolitan area networks-part 15.7: Short-range optical wireless communications. *IEEE Std 802.15.7-2018 (Revision of IEEE Std 802.15.7-2011)*, Apr 2019.

[4] E. Ahmed and A. M. Eltawil. All-digital self-interference cancellation technique for full-duplex systems. *IEEE Transactions on Wireless Communications*, 14(7):3519–3532, July 2015.

[5] V. Alessandrini. *Shared Memory Application Programming: Concepts and strategies in multicore application programming*, pages 435,436. Elsevier, 2016. ISBN 978-012803761.

[6] S. E. Anderson. Bit twiddling hacks. URL: `https://graphics.stanford.edu/~seander/bithacks.html`, last checked on 29-05-2019.

[7] Arduino. Arduino due. URL: `https://store.arduino.cc/due`, last checked on 31-05-2019.

[8] Arduino. Simple audio player. URL: `https://www.arduino.cc/en/Tutorial/SimpleAudioPlayer`, last checked on 31-05-2019.

[9] Arduino. What is arduino? URL: `https://www.arduino.cc/en/Guide/Introduction`, last checked on 31-05-2019.

[10] ARM. *Cortex-M Technical Reference Manual*, 2005,2006. r1p1.

[11] Atmel. *SMART ARM-based MCU, SAM3X / SAM3A Series*, May 2015.

[12] D. Barrett and T. King. *Computer Networking Illuminated*, pages 202–205. Elsevier, 2005. ISBN 0763726761.

[13] T. Bartley, S. Tanaka, Y. Nonomura, T. Nakayama, and M. Muroyama. Delay window blind oversampling clock and data recovery algorithm with wide tracking range. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1598–1601, May 2015.

[14] A. G. Bell. The photophone. *Journal of the Franklin Institute*, 110(4):237 – 248, 1880.

[15] M. Bellanger. *Digital Processing of Signals.* Wiley, 1984.

[16] J. Beysens, A. Galisteo, Q. Wang, D. Juara, D. Giustiniano, and S. Pollin. Densevlc: A cell-free massive mimo system with distributed leds. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '18, pages 320–332. ACM, 2018.

[17] E. Bogatin. Rule of thumb # 1: Bandwidth of a signal from its rise time. URL: `https://www.edn.com/electronics-blogs/bogatin-s-rules-of-thumb/4424573/Rule-of-Thumb-1-The-bandwidth-of-a-signal-from-its-rise-time`, last checked on 31-05-2019.

[18] A. Brand and H. Aghvami. *Multiple Access Protocols for Mobile Communications*, page 4. John Wiley & Sons, 2002. ISBN 0471498777.

[19] D. C. O'Brien, G. Faulkner, H. Le Minh, O. Bouchet, M. El Tabach, M. Wolf, W. Joachim, S. Randel, S. Nerreter, M. Franke, K.-D. Langer, J. Grubor, and T. Kamalakis. Home access networks using optical wireless transmission. pages 1 – 5, 10 2008.

[20] L. W. Couch. *Digital and Analog Communication Systems.* Pearson, 8 edition, 2013. ISBN-13 978-0132915380.

[21] CREE. *Cree XLamp XT-E LEDs*, 2019. 2600K - 3700K CCT.

[22] E. Dahlman. *Communications Engineering Desk Reference*, page 208. Academic Press, 2009. ISBN-13: 978-01237468481.

[23] J. L. Devore. *Probability and Statistics for Engineering and the Sciences.* Brooks/Cole, Cengage Learning, 2012. ISBN-13: 978-0538733526.

[24] T.-H. Do and M. Yoo. An in-depth survey of visible light communication based positioning systems. *Sensors*, 16 5, 2016.

[25] T. Elganimi. Performance comparison between ook, ppm and pam modulation schemes for free space optical (fso) communication systems: Analytical study. *International Journal of Computer Applications - Foundation of Computer Science - New York, NY 10001, USA*, Vol.79:pp. 22–27, 10 2013.

[26] T. Finch. Incremental calculation of weighted mean and variance. Technical report, Feb 2009.

[27] FrankBoesing. Fast crc library for arduino. URL: `https://github.com/FrankBoesing/FastCRC`, last checked on 31-05-2019.

[28] freepik. Laptop free icon. URL: `https://www.flaticon.com/free-icon/laptop_114734#term=laptop&page=1&position=7`, last checked on 31-05-2019.

[29] J. Garcia-Luna-Aceves and C. L. Fullmer. Floor acquisition multiple access (fama) in single-channel wireless networks. *Mobile Networks and Applications*, 4(3):157–174, Sep 1999.

[30] Geyer Electronic. *Quartz Crystal*, 2018. part no: 12Mhz, model: KX-7.

[31] G. Ghassemlooy, W. Popoola, and S. Rajbhandari. *Optical Wireless Communications System and Channel Modelling in MATLAB.* Taylor & Francis Group, 2013. ISBN-13: 978-1439852354.

[32] Z. Ghassemlooy, L. N. Alves, S. Zvanovec, and M.-A. Khalighi. *Visible Light Communications Theory and Applications*, pages 130–139. Taylor & Francis Group, 2017.

[33] Guan-Chyun Hsieh and J. C. Hung. Phase-locked loop techniques. a survey. *IEEE Transactions on Industrial Electronics*, 43(6):609–615, Dec 1996.

[34] T. C. Guide. Beacon towers of china great wall. URL: `https://www.travelchinaguide.com/china_great_wall/construction/tower/`, last checked on 5-06-2019.

[35] Hamamatsu. *Si PIN photodiode S5971, S5972, S5972 series*, 2003. model s5971.

[36] X.-L. Hu, P.-H. Ho, and L. Peng. Fundamental limitations in energy detection for spectrum sensing. *Journal of Sensor and Actuator Networks*, Jun 2018.

[37] Jaeha Kim and Deog-Kyoon Jeong. Multi-gigabit-rate clock and data recovery based on blind oversampling. *IEEE Communications Magazine*, 41(12):68–74, Dec 2003.

[38] S.-Y. Jung, D.-H. Kwon, S.-H. Yang, and S.-K. Han. Reduction of inter-cell interference in asynchronous multi-cellular vlc by using ofdma-based cell partitioning. *2016 18th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4, 2016.

[39] T. Kabra. Nonlinearity mitigation in visible light communication. Master's thesis, Indraprastha Institute of Information Technology, Apr 2017.

[40] T. G. Kang, S.-K. Lim, D. H. Kim, I. S. Jang, and D. W. Han. Ieee 802.15.7 vlc phy/mac specification proposal, Oct 2009.

[41] R. C. Kizilirmak, O. Narmanlioglu, and M. Uysal. Centralized light access network (c-lian): A novel paradigm for next generation indoor vlc networks. *IEEE Access*, 5:19703–19710, 2017.

[42] R. B. Knapp, E. Finkelman, M.-K. Kee, and A. Tanaka. Accessible design of a portable electronic museum guide for universal access. 2004.

[43] H. Li, X. Chen, B. Huang, D. Tang, and H. Chen. High bandwidth visible light communications based on a post-equalization circuit. *IEEE Photonics Technology Letters*, 26(2):119–122, Jan 2014.

[44] M. Mandal and A. Asif. *Continuous and Discrete Time Signals and Systems*, page 14. Cambridge University Press, 2007. ISBN-13 978-0-521-85455-9.

[45] R. T. Marler and J. S. Arora. Function-transformation methods for multi-objective optimization. *Engineering Optimization*, pages 551–570, 2005.

[46] A. Masmoudi and T. Le-Ngoc. Residual self-interference after cancellation in full-duplex systems. In *2014 IEEE International Conference on Communications (ICC)*, pages 4680–4685, Jun 2014.

[47] Modern and Antenna Audio. *Tate Modern Multimedia Tour Pilots*, 2003.

[48] E. Nasif, T. Salih, S. Kucukoner, and K. Fidanboylu. An overview of handoff techniques in cellular networks. *Information Technology - IT*, 2(3), 01 2005.

[49] J. D. Natale and S. J. Andrabi. Channel capture effect in 802.11: A study.

[50] T. Needham. A visual explanation of jensen's inequality. *American Mathematical Monthly*, 100, 10 1993.

[51] A. F. Oncel. *ON SELF-INTERFERENCE CANCELLATION IN WIRELESS FULL DUPLEX*. PhD thesis, Middle East Technical University, 2016.

[52] P. H. Pathak, X. Feng, P. Hu, and P. Mohapatra. Visible light communication, networking, and sensing: A survey, potential and challenges. *IEEE Communications Surveys Tutorials*, 17(4):2047–2077, Fourthquarter 2015.

[53] M. Puckette. *The Theory and Technique of Electronic Music*, page 12. World Scientific Publishing, Dec 2007. ISBN-13 978-9812700773.

[54] S. Rajagopal, R. D. Roberts, and S. Lim. Ieee 802.15.7 visible light communication: modulation schemes and dimming support. *IEEE Communications Magazine*, 50(3):72–82, Mar 2012.

[55] Research and Markets. *Free Space Optics (FSO) and Visible Light Communication (VLC)/Light Fidelity (Li-Fi) Market by Component (LED, Photodetector, Microcontroller, and Software), Transmission Type, Application, and Geography - Global Forecast to 2023 ID: 4622797 Report September 2018 Region: Global 18*, 2018.

[56] M. Schroeder and B. Atal. Code-excited linear prediction(celp): High-quality speech at very low bit rates. In *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 937–940, April 1985.

[57] simonyipeter. Reed-solomon forward error correction library. URL: `https://github.com/simonyipeter/Arduino-FEC`, last checked on 31-05-2019.

[58] Smashicons. Socket free icon. URL: `https://www.flaticon.com/free-icon/socket_222538#term=plug%20and%20socket&page=1&position=48`, last checked on 31-05-2019.

[59] Y. Tanaka, S. Haruyama, and M. Nakagawa. Wireless optical transmissions with white colored led for wireless home links. In *11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications. PIMRC 2000. Proceedings (Cat. No.00TH8525)*, volume 2, pages 1325–1329 vol.2, Sep. 2000.

[60] A. S. Tanenbaum. *Computer Networks Fifth Edition*. Pearson Education, 2011. ISBN-13: 978-0132126953.

[61] S. Tasaka. *Mathematical Statistics with Mathematica*. The MIT Press, 1986. ISBN 0262200589.

[62] S. V. Tracy. Darkness from light: The beacon fire in the agamemnon. *The Classical Quarterly*, 36(1):257–260, 1986.

[63] c.-y. Tsai, S.-Y. Chou, and S.-W. Lin. *Location-Aware Tour Guide Systems in Museum*, volume 5, pages 349–356. 01 2008.

[64] A. Tsiatmas, F. M. Willems, J. M. Linnartz, S. Baggen, and J. W. Bergmans. Joint illumination and visible-light communication systems: Data rates and extra power consumption. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 1380–1386, June 2015.

[65] United States Government Accountability Office. *FFCC Should Track Growth to Ensure Sufficient Spectrum Remains Available*, 2017.

[66] T. Vermeulen. *Reliable and Low Delay Communication for Wireless Sensor Networks using In-Band Full Duplex*. PhD thesis, KU Leuven, 2017.

[67] T. Vermeulen, M. Laghate, G. Hattab, D. Cabric, and S. Pollin. Towards instantaneous collision and interference detection using in-band full duplex. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, May 2017.

[68] Vishay Semicondutors. *High Speed Infrared Emitting Diode, 830 nm, GaAlAs Double Hetero*, Aug 2011.

[69] K. Voulgaris, A. Gkelias, I. Ashraf, M. Dohler, and A. H. Aghvami. Throughput analysis of wireless csma/cd for a finite user population. In *IEEE Vehicular Technology Conference*, Sep 2006.

[70] Q. Wang and D. Giustiniano. Intra-frame bidirectional transmission in networks of visible leds. *IEEE/ACM Transactions on Networking*, 24:1–13, 12 2016.

[71] Wikipedia. Bernoulli process. URL: `https://en.wikipedia.org/wiki/Bernoulli_sequence`, last checked on 31-05-2019.

[72] Wikipedia. Fairness measure. URL: `https://en.wikipedia.org/wiki/Fairness_measure`, last checked on 31-05-2019.

[73] Wikipedia. Hysteresis. URL: `https://en.wikipedia.org/wiki/Hysteresis`, last checked on 2-6-2019.

[74] Wikipedia. Location-based service. URL: `https://en.wikipedia.org/wiki/Location-based_service`, last checked on 3-06-2019.

[75] Wikipedia. Random walk. URL: `https://en.wikipedia.org/wiki/Random_walk`, last checked on 31-05-2019.

[76] Wikipedia. Reed-solomon error correction. URL: `https://en.wikipedia.org/wiki/Reed-Solomon_error_correction`, last checked on 3-06-2019.

[77] Wikipedia. Signal lamp. URL: `https://en.wikipedia.org/wiki/Signal_lamp`, last checked on 4-06-2019.

[78] Wikipedia. Simple linear regression. URL: `https://en.wikipedia.org/wiki/Simple_linear_regression`, last checked on 4-06-2019.

[79] Xiph.Org. Opus interactive audio codec. URL: `http://opus-codec.org/`, last checked on 31-05-2019.

[80] Xiph.Org. Speex: A free codec for free speech. URL: `https://www.speex.org/`, last checked on 31-05-2019.

[81] F. Zappa. *Microcontrollers. Hardware and Firmware for 8-bit and 32-bit devices*, page 345. Societa Editrice Esculapio, 2017. ISBN-13: 978-8893850223.

[82] Z. Zhang, X. Yu, L. Wu, J. Dang, and V. O. K. Li. Performance analysis of full-duplex visible light communication networks. In *2015 IEEE International Conference on Communications (ICC)*, pages 3933–3938, June 2015.

[83] K. Zhou, C. Gong, Q. Gao, and Z. Xu. Inter-cell interference coordination for multi-color visible light communication networks. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 6–10, Dec 2016.