TECHNISCHE
UNIVERSITÄT
DARMSTADT

DECENTRALIZED ULTRA-RELIABLE LOW-LATENCY COMMUNICATIONS THROUGH CONCURRENT COOPERATIVE TRANSMISSION

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
von

Robin Klose

Erstreferent: Prof. Dr.-Ing. Matthias Hollick
Korreferent: Prof. Francesco Gringoli, Ph.D.

Darmstadt 2023
Hochschulkennziffer D17

Robin Klose, *Decentralized Ultra-Reliable Low-Latency Communications through Concurrent Cooperative Transmission*, Dissertation, Technische Universität Darmstadt, 2023.

Fachgebiet Sichere Mobile Netze
Fachbereich Informatik
Technische Universität Darmstadt
Jahr der Veröffentlichung: 2023
Tag der mündlichen Prüfung: 10. Februar 2023
URN: urn:nbn:de:tuda-tuprints-240702

ABSTRACT

Emerging cyber-physical systems demand for communication technologies that enable seamless interactions between humans and physical objects in a shared environment. This thesis proposes decentralized URLLC (dURLLC) as a new communication paradigm that allows the nodes in a wireless multi-hop network (WMN) to disseminate data quickly, reliably and without using a centralized infrastructure. To enable the dURLLC paradigm, this thesis explores the practical feasibility of concurrent cooperative transmission (CCT) with orthogonal frequency-division multiplexing (OFDM). CCT allows for an efficient utilization of the medium by leveraging interference instead of trying to avoid collisions. CCT-based network flooding disseminates data in a WMN through a reception-triggered low-level medium access control (MAC). OFDM provides high data rates by using a large bandwidth, resulting in a short transmission duration for a given amount of data.

This thesis explores CCT-based network flooding with the OFDM-based IEEE 802.11 Non-HT and HT physical layers (PHYs) to enable interactions with commercial devices. An analysis of CCT with the IEEE 802.11 Non-HT PHY investigates the combined effects of the phase offset (PO), the carrier frequency offset (CFO) and the time offset (TO) between concurrent transmitters, as well as the elapsed time. The analytical results of the decodability of a CCT are validated in simulations and in testbed experiments with Wireless Open Access Research Platform (WARP) v3 software-defined radios (SDRs). CCT with coherent interference (CI) is the primary approach of this thesis.

Two prototypes for CCT with CI are presented that feature mechanisms for precise synchronization in time and frequency. One prototype is based on the WARP v3 and its IEEE 802.11 reference design, whereas the other prototype is created through firmware modifications of the Asus RT-AC86U wireless router. Both prototypes are employed in testbed experiments in which two groups of nodes generate successive CCTs in a ping-pong fashion to emulate flooding processes with a very large number of hops. The nodes stay synchronized in experiments with 10 000 successive CCTs for various modulation and coding scheme (MCS) indices and MAC service data unit (MSDU) sizes. The URLLC requirement of delivering a 32-byte MSDU with a reliability of $1 - 10^{-5}$ and with a latency of 1 ms is assessed in experiments with 1 000 000 CCTs, while the reliability is approximated by means of the frame reception rate (FRR). An FRR of at least 99.999 % is achieved at PHY data rates of up to 48 Mbit/s under line-of-sight (LOS) conditions and at PHY data rates of up to 12 Mbit/s under non-line-of-sight (NLOS) conditions on a 20 MHz wide channel, while the latency per hop is 48.2 µs and 80.2 µs, respectively. With four multiple input multiple output (MIMO) spatial streams on a 40 MHz wide channel, a LOS receiver achieves an FRR of 99.5 % at a PHY data rate of 324 Mbit/s. For CCT with incoherent interference, this thesis proposes equalization with time-variant zero-forcing (TVZF) and presents a TVZF receiver for the IEEE 802.11 Non-HT PHY, achieving an FRR of up to 92 % for CCTs from three unsyntonized commercial devices. As CCT-based network flooding allows for an implicit time synchronization of all nodes, a reception-triggered low-level MAC and a reservation-based high-level MAC may in combination support various applications and scenarios under the dURLLC paradigm.

ZUSAMMENFASSUNG

Aufkommende cyber-physische Systeme verlangen nach Kommunikationstechnologien, die reibungslose Interaktionen zwischen Menschen und physischen Objekten in einer gemeinschaftlichen Umgebung ermöglichen. Diese Dissertation schlägt dezentralisiertes URLLC (dURLLC) als ein neues Kommunikationsparadigma vor, das es Knoten in einem drahtlosen Multi-Hop-Netzwerk (WMN) erlaubt, Daten schnell, zuverlässig und ohne zentralisierte Infrastruktur zu verbreiten. Um das dURLLC-Paradigma zu ermöglichen, erforscht diese Dissertation die praktische Umsetzbarkeit gleichzeitiger kooperativer Übertragungen (CCT) mit dem orthogonalen Frequenzmultiplexverfahren (OFDM). CCT erlaubt eine effiziente Nutzung des Mediums durch Ausnutzen von Interferenzen, statt Kollisionen möglichst zu vermeiden. CCT-basiertes Netzfluten verteilt Daten in einem WMN durch eine empfangsgetriggerte niedere Medienzugriffssteuerung (MAC). OFDM bietet hohe Datenraten durch die Nutzung einer großen Bandbreite und erzielt somit eine kurze Sendedauer für eine bestimmte Menge an Daten.

Diese Dissertation untersucht CCT-basiertes Netzfluten mit den OFDM-basierten IEEE 802.11 Non-HT und HT Bitübertragungsschichten (PHYs), um Interaktionen mit kommerziellen Geräten zu ermöglichen. Eine Analyse zu CCT mit dem IEEE 802.11 Non-HT PHY untersucht die zusammenwirkenden Effekte des Phasenversatzes (PO), des Trägerfrequenzversatzes (CFO) und des Zeitversatzes (TO) zwischen gleichzeitigen Sendern, sowie der verstrichenen Zeit. Die analytischen Ergebnisse der Dekodierbarkeit von CCT werden durch Simulationen und Testbedexperimente mit softwaredefinierten Funksystemen (SDRs) der Wireless Open Access Research Platform (WARP) v3 validiert. CCT mit kohärenter Interferenz (CI) ist der primäre Ansatz dieser Dissertation.

Es werden zwei Prototypen für CCT mit CI vorgestellt, die sich durch Mechanismen zur präzisen Frequenz- und Zeitsynchronisierung auszeichnen. Ein Prototyp basiert auf WARP v3 und dem zugehörigen IEEE 802.11 Referenzdesign, während der andere Prototyp durch Veränderungen der Firmware eines Asus RT-AC86U Drahtlosrouters erstellt wird. Beide Prototypen werden in Testbedexperimenten eingesetzt, in denen zwei Knotengruppen aufeinanderfolgende CCTs erzeugen, die sich gegenseitig auslösen, um Flutprozesse mit einer sehr großen Anzahl an Hops nachzuahmen. Die Knoten erhalten ihre Synchronisierung in Experimenten mit 10 000 aufeinanderfolgenden CCTs bei verschiedenen Indizes der Modulations- und Kodierverfahren (MCS) und Größen der MAC-Service-Dateneinheit (MSDU) aufrecht. Die URLLC-Anforderung, welcher zufolge eine MSDU mit 32 Bytes mit einer Zuverlässigkeit von $1 - 10^{-5}$ und mit einer Latenz von 1 ms zu übermitteln ist, wird in Experimenten mit 1 000 000 CCTs geprüft, während die Zuverlässigkeit durch die Frame-Empfangsrate (FRR) angenähert wird. Bei 20 MHz Kanalbandbreite wird eine FRR von 99,999 % unter Sichtverbindung (LOS) bei PHY-Datenraten von bis zu 48 Mbit/s erzielt und ohne Sichtverbindung (NLOS) bei PHY-Datenraten von bis zu 12 Mbit/s, während die Latenz pro Hop jeweils 48,2 µs und 80,2 µs beträgt. Mit vier räumlichen Strömen via Multiple Input Multiple Output (MIMO) auf einem 40 MHz breiten Kanal erzielt ein LOS-Empfänger bei einer PHY-Datenrate von 324 Mbit/s eine FRR von 99,5 %. Für CCT mit inkohärenter Interferenz schlägt diese Dissertation die Entzerrung durch zeitvariantes Zero-Forcing (TVZF) vor und stellt

einen TVZF-Empfänger für den IEEE 802.11 Non-HT PHY vor, der für CCTs von drei kommerziellen, nicht frequenzsynchronisierten Geräten eine FRR von bis zu 92 % erzielt. Da CCT-basiertes Netzfluten eine implizite Zeitsynchronisierung aller Knoten erlaubt, können ein empfangsgetriggerter niederer MAC und ein reservierungsbasierter höherer MAC in Kombination verschiedenartige Anwendungen und Szenarien unterstützen, die unter das dURLLC-Paradigma fallen.

ACKNOWLEDGMENTS

Special thanks to my supervisor Prof. Dr.-Ing. Matthias Hollick for his confidence in my research abilities, for suggesting a compelling research topic and for providing valuable scientific advice.

Special thanks to Prof. Francesco Gringoli, Ph.D., for the fruitful collaboration, for inviting me to a research stay in Brescia and for providing comprehensive scientific and technical advice.

Thanks to all the students I have supervised over the past years for the successful cooperations. Special thanks to my students Vladimir Lieber (née Roskin), Konrad Altenhofen and Niklas Müller, whose contributions are partly included and discussed in this dissertation.

Thanks to all my colleagues at SEEMOO that supported me during the last couple of years. Special thanks to Lars Almon and Eicke Hauck for their first-class technical support.

Thanks to my colleagues at MAKI for interdisciplinary discussions and collaborations.

Special thanks to my family and friends for their love and support!

This work has been funded by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 MAKI.

This work has been funded by the Federal Ministry of Education and Research (BMBF) and by the Hessian Ministry of Higher Education, Research and the Arts (HMWK) within the national research center for applied cyber security ATHENE.

This work has been funded by the Federal Ministry of Education and Research (BMBF) within the national research center Open6GHub.

This work has been supported through 80 000 core hours of the high-performance computer Lichtenberg at Technische Universität Darmstadt, which is part of the national high-performance computing (NHR) center NHR4CES. These resources have been funded by the Federal Ministry of Education and Research (BMBF) and by the state governments participating on the basis of the resolutions of the Joint Science Conference (GWK) for NHR at universities.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

3GPP	3rd Generation Partnership Project
4G	fourth generation
5G	fifth generation
A-MSDU	aggregated MSDU
ACK	acknowledgment
ADC	analog-to-digital converter
AGC	automatic gain control
AODV	Ad-hoc On-demand Distance Vector
AP	access point
AWGN	additive white Gaussian noise
BER	bit error rate
BFSK	binary FSK
BPSK	binary phase-shift keying
BS	base station
BSS	basic service set
BSSID	BSS identifier
CANDI	Cooperative Analog and Digital

CCK	complementary code keying
CCT	concurrent cooperative transmission
CDMA	code-division multiple access
CDN	content delivery network
CDS	channel delay spread
CFO	carrier frequency offset
CG	configured grant
CI	coherent interference
CoMP	cooperative multi-point
CP	cyclic prefix
CPE	common phase error
CPU	central processing unit
CRC	cyclic redundancy check
CSAC	chip-scale atomic clock
CSI	channel state information
CSMA	carrier sense multiple access
CSMA/CA	CSMA with collision avoidance
CSMA/CD	CSMA with collision detection
CTS	clear to send
D2D	device-to-device
DAC	digital-to-analog converter
DAS	distributed antenna system
DBPSK	differential binary phase-shift keying
DCF	distributed coordination function
DDS	direct digital synthesis
DL	downlink
DMIMO	distributed MIMO
DNN	deep neural network
DQ	distributed queueing
DQPSK	differential QPSK
DS	distribution system
DSAP	destination service access point
DSR	Dynamic Source Routing
DSRC	dedicated short range communications
DSSS	direct-sequence spread spectrum
dURLLC	decentralized URLLC
DUT	device under test

DVB-T2	Digital Video Broadcasting – Second Generation Terrestrial
EEPROM	electrically erasable programmable read-only memory
EKF	extended Kalman filter
eMBB	enhanced Mobile BroadBand
EVM	error vector magnitude
FCS	frame check sequence
FEC	forward error correction
FFT	fast Fourier transform
FHSS	frequency-hopping spread spectrum
FM	frequency modulation
FMC	FPGA mezzanine card
FPGA	field-programmable gate array
FRR	frame reception rate
FSK	frequency-shift keying
FTR	frame transmission rate
FTR-G	frame transmission rate of group
FTSP	Flooding Time Synchronization Protocol
GI	guard interval
GNSS	global navigation satellite system
GNU	GNU's Not Unix
GPIO	general-purpose input/output
GPS	Global Positioning System
HE	high efficiency
HT	high throughput
HT-LTF	HT long training field
HT-MF	HT-mixed format
HT-STF	HT short training field
IBSS	independent basic service set
IC	integrated circuit
ICI	intercarrier interference
IEEE	Institute of Electrical and Electronics Engineers
IFFT	inverse FFT

IFS	interframe space
IIC	iterative interference cancellation
INP	in-network processing
IP	Internet Protocol
IQ	in-phase quadrature
IR	infrared
ISI	intersymbol interference
ISM	Industrial, Scientific and Medical
ISP	Internet service provider
ITS	intelligent transportation system
ITU	International Telecommunication Union
ITU-T	Telecommunication Standardization Sector of ITU
L-LTF	legacy long training field
L-SIG	legacy SIGNAL field
L-STF	legacy short training field
LAN	local area network
LDPC	low-density parity check
LED	light-emitting diode
LFSR	linear-feedback shift register
LLC	logical link control
LOS	line-of-sight
LTE	Long-Term Evolution
LTS	long training sequence
LVDS	low-voltage differential signaling
LWB	Low-Power Wireless Bus
MAC	medium access control
MBMS	Multimedia Broadcast and Multicast Service
MBSFN	Multicast/Broadcast over SFN
MCS	modulation and coding scheme
MCU	microcontroller unit
MEC	mobile edge computing
MIMO	multiple input multiple output
MISO	multiple input single output
MLME	MAC sublayer management entity
MMCX	micro-miniature coaxial
MMSE	minimum mean square error

mMTC	massive Machine Type Communications
MPDU	MAC protocol data unit
MRC	maximum-ratio combining
MS/s	megasamples per second
MSDU	MAC service data unit
MSE	mean square error
MTU	maximum transmission unit
MU-MIMO	multi-user MIMO
MUD	multi-user detection
MUSIC	MUltiple SIgnal Classification
NAV	network allocation vector
NET	network layer
NLOS	non-line-of-sight
Non-HT	Non-HT
NR	New Radio
NTP	Network Time Protocol
O-QPSK	offset QPSK
OFDM	orthogonal frequency-division multiplexing
OLA	opportunistic large array
OLACRA	OLA Concentric Routing Algorithm
OLAROAD	OLA Routing On-Demand
OLSR	Optimized Link State Routing
OOK	on-off keying
OUI	organizationally unique identifier
PCF	point coordination function
PDU	protocol data unit
PHY	physical layer
PIC	parallel interference cancellation
PLL	phase-locked loop
PO	phase offset
ppb	parts per billion
PPDU	PHY protocol data unit
ppm	parts per million
PPS	pulse per second
PSDU	PHY service data unit

PSM	programmable state machine
PTP	Precision Time Protocol
PUSCH	physical uplink shared channel
QAM	quadrature amplitude modulation
QoS	quality of service
QPSK	quadrature phase-shift keying
RA	receiver address
RAM	random-access memory
RAN	radio access network
RBS	reference-broadcast synchronization
RF	radio frequency
RMS	root mean square
RMSE	RMS error
RREP	route reply
RREQ	route request
RSS	received signal strength
RSSI	received signal strength indicator
RTS	request to send
RTT	round-trip time
RTx	retransmission
Rx	receive
SCSF	Semi-Cooperative Spectrum Fusion
SDR	software-defined radio
SER	symbol error rate
SFD	start of frame delimiter
SFN	single frequency network
SFO	sampling frequency offset
SIC	successive interference cancellation
SIFS	short interframe space
SINR	signal-to-interference-plus-noise ratio
SNAP	Subnetwork Access Protocol
SNR	signal-to-noise ratio
SOF	start of frame
SSAP	source service access point
SSH	Secure Shell

STBC	space-time block code
STS	short training sequence
SyncE	synchronous Ethernet
TA	transmitter address
TCXO	temperature-compensated crystal oscillator
TDD	time-division duplex
TGn	High Throughput Task Group
TO	time offset
ToF	time-of-flight
TPSN	Timing-sync Protocol for Sensor Networks
TVZF	time-variant zero-forcing
Tx	transmit
UART	universal asynchronous receiver-transmitter
UDP	User Datagram Protocol
UL	uplink
URLLC	Ultra-Reliable Low-Latency Communications
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
UWB	ultra-wideband
VCO	voltage-controlled oscillator
VCTCXO	voltage-controlled TCXO
VHT	very high throughput
WARP	Wireless Open Access Research Platform
WLAN	wireless LAN
WMN	wireless multi-hop network
WSN	wireless sensor network

LIST OF PUBLICATIONS

During the course of working on this thesis, I co-authored several papers and articles:

First Author and Co-First Author

[1] Francesco Gringoli, Robin Klose, Matthias Hollick, and Nahla Ali. "Making Wi-Fi Fit for the Tactile Internet: Low-Latency Wi-Fi Flooding Using Concurrent Transmissions." In: *IEEE International Conference on Communications Workshops (ICC Workshops)*. Kansas City, Missouri, USA: IEEE, May 2018, pp. 1–6. DOI: 10.1109/ICCW.2018.8403487. **Part of this thesis.**

[2] Robin Klose and Matthias Hollick. "Riding the Waves: Decoding Asynchronous Multi-User MISO via Time-Variant Zero-Forcing." In: *IEEE Global Communications Conference (GLOBECOM)*. Waikoloa, Hawaii, USA: IEEE, Dec. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9013132. **Part of this thesis.**

[3] Kay Luis Wallaschek, Robin Klose, Lars Almon, and Matthias Hollick. "NEAT-TCP: Generation of TCP Congestion Control through Neuroevolution of Augmenting Topologies." In: *IEEE International Conference on Communications Workshops (ICC Workshops)*. Dublin, Ireland: IEEE, June 2020, pp. 1–6. DOI: 10.1109/ICCWorkshops49005.2020.9145446.

Co-Author

[4] Bastian Alt, Markus Weckesser, Christian Becker, Matthias Hollick, Sounak Kar, Anja Klein, Robin Klose, Roland Kluge, Heinz Koeppl, Boris Koldehofe, Wasiur R. Khudabukhsh, Manisha Luthra, Mahdi Mousavi, Max Mühlhäuser, Martin Pfannemüller, Amr Rizk, Andy Schürr, and Ralf Steinmetz. "Transitions: A Protocol-Independent View of the Future Internet." In: *Proceedings of the IEEE* 107.4 (Apr. 2019), pp. 835–846. ISSN: 1558-2256. DOI: 10.1109/JPROC.2019.2895964.

[5] Martin Pfannemüller, Janick Edinger, Markus Weckesser, Roland Kluge, Manisha Luthra, Robin Klose, Christian Becker, and Andy Schürr. "Demo: Visualizing Adaptation Decisions in Pervasive Communication Systems." In: *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. Kyoto, Japan: IEEE, Mar. 2019, pp. 335–337. DOI: 10.1109/PERCOMW.2019.8730661.

[6] Martin Pfannemüller, Markus Weckesser, Roland Kluge, Janick Edinger, Manisha Luthra, Robin Klose, Christian Becker, and Andy Schürr. "CoalaViz: Supporting Traceability of Adaptation Decisions in Pervasive Communication Systems." In: *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. Kyoto, Japan: IEEE, Mar. 2019, pp. 590–595. DOI: 10.1109/PERCOMW.2019.8730818.

[7] Martin Pfeiffer, Jan-Pascal Kwiotek, Jiska Classen, Robin Klose, and Matthias Hollick. "Analyzing TETRA Location Privacy and Network Availability." In: *6th Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)*. Vienna, Austria: ACM, Oct. 2016, pp. 117–122. DOI: 10.1145/2994459.2994463.

Patents

[8] Robin Klose and Matthias Hollick. *Vorrichtung und Verfahren zum Dekodieren von überlagerten, asynchronen Übertragungen*. Patent No. 10 2019 117 607.7. Deutsches Patent- und Markenamt, Oct. 2020. **Part of this thesis.**

Under Peer Review

[9] Francesco Gringoli, Robin Klose, and Matthias Hollick. "Wi-Fi on Steroids: on the Practical Feasibility of Concurrent Transmission with IEEE 802.11 DSSS and OFDM Physical Layers." In: *(to be resubmitted)* (2023). **Part of this thesis.**

[10] Robin Klose, Matthias Hollick, and Francesco Gringoli. "Riding the Waves in Uncharted Waters: A Deep Dive into Decoding Asynchronous Multi-User MISO via Time-Variant Zero-Forcing." In: *(to be resubmitted)* (2023). **Part of this thesis.**

[11] Artur Sterz, Robin Klose, Markus Sommer, Jonas Höchst, Jakob Link, Bernd Simon, Anja Klein, Matthias Hollick, and Bernd Freisleben. "Energy-Efficient Broadcast Trees for Decentralized Data Dissemination in Wireless Networks." In: *48th IEEE Conference on Local Computer Networks (LCN)* (2023).

COLLABORATIONS AND MY CONTRIBUTIONS

This section lists the respective contributions of my co-authors and of myself per chapter. In addition, to comply with the regulations of the Department of Computer Science at Technische Universität Darmstadt, this section declares the parts that include verbatim or revised fragments of previous publications that are incorporated into this thesis as indicated in the preceding list of publications.[1]

Chapter 1 outlines the motivation, the goals and the contributions [9, 1, 2, 8, 10]. Matthias Hollick contributed the initial idea of investigating the practical feasibility of low-latency network flooding through CCT with an OFDM-based IEEE 802.11 PHY, with the right intuition for performance gains in comparison to existing technologies. Early testbed experiments with the prototypes of this work showed that the CCTs have an excellent signal quality, giving rise to the idea to use the URLLC requirement of the 3GPP as a performance benchmark and to define the dURLLC paradigm accordingly for decentralized WMNs, which are my contributions. Chapter 2 presents related work and technical background information on IEEE 802.11 [9, 1, 2, 8, 10].

Chapter 3 comprises several contributions of Francesco Gringoli as well as of myself, the creation of which was also encouraged by Matthias Hollick in joint discussions [9, 1]. Section 3.1 presents an experimental study that was conducted by Francesco Gringoli. Section 3.2 presents a PHY-independent system model devised by Francesco Gringoli. Section 3.3 presents an analysis of CCT with the IEEE 802.11 DSSS PHY that comprises various contributions. Section 3.3.1 and Section 3.3.2 present a theoretical analysis as well as simulation results that were worked out by Francesco Gringoli. Figure 13 was added by myself to illustrate some relevant principles. Section 3.3.3 presents testbed experiments that were conducted by myself in collaboration with Francesco Gringoli. The software encoder and decoder for IEEE 802.11 DSSS PPDUs was provided by Francesco Gringoli. Section 3.4 presents an analysis of CCT with the IEEE 802.11 Non-HT PHY that comprises various contributions. Section 3.4.1 presents a theoretical analysis that considers the case without symbol errors and that was devised by Francesco Gringoli. Figure 20 was added by myself to illustrate some relevant principles. Section 3.4.2 presents an analysis that considers the ability of FEC to correct a moderate amount of symbol errors and that was devised by myself, while building upon the preceding analysis in Section 3.4.1. Section 3.4.3 presents testbed experiments that were conducted by myself in collaboration with Francesco Gringoli. The software encoder for IEEE 802.11 Non-HT PPDUs employed here is based on a Matlab implementation that was kindly provided by my colleagues Matthias Schulz and Jiska Classen. This software encoder and decoder was also adapted by myself to create the plots in Figure 20, Figure 22 and Figure 24 as well as to conduct the CFO measurements presented in Section 3.5.1. Section 3.5 presents the CFO characteristics of WARP v3 SDRs and Asus RT-AC86U wireless routers, which were measured by myself. Francesco Gringoli provided the modified firmware for the Asus RT-AC86U devices, as described in Section 3.5.2. Section 3.6 presents an enhanced CFO estimation technique for the IEEE 802.11 Non-HT and HT PHYs that was devised by myself.

1 References in this chapter refer to my list of publications given on Pages xxiii to xxiv.

Chapter 4 comprises joint contributions of Konrad Altenhofen and myself as well as various individual contributions of myself, while Matthias Hollick was involved in several discussions. Section 4.1 presents the design of an enhanced CFO estimator integrated into the PHY Rx core and a frequency shifter integrated into the PHY Tx core of the WARP prototype, which are joint contributions of Konrad Altenhofen and myself, while he worked on this topic in the course of his bachelor thesis under my supervision. The initial idea of a frequency synchronization mechanism through baseband processing as well as the conceptualization of the idea in the context of the IEEE 802.11 reference design for WARP v3 were contributed by myself. Konrad Altenhofen implemented these concepts independently, ran simulations and synthesized the updated designs for the FPGA of the WARP v3. In this collaboration, we discussed intermediate results and refined the design in several iterations. Moreover, Konrad Altenhofen validated the functionality of individual components while considering my ideas on the methodology. The hardware design presented in Section 4.1 was slightly modified by myself to address certain corner cases. The remainder of Chapter 4 comprises contributions of myself.

Chapter 5 comprises several contributions of Francesco Gringoli as well as of myself, while Matthias Hollick was involved in several discussions, encouraging the creation of a prototype based on a commercial device. Section 5.1 was contributed by myself, while Francesco Gringoli provided feedback. Section 5.2 presents the implementation of the Asus RT-AC86U prototype. Francesco Gringoli developed the firmware modifications of the Asus RT-AC86U wireless router, exposed certain functions through shell scripts and provided the command server for efficient command invocations over Ethernet. Section 5.2.1 presents an automated reverse engineering process searching for PHY registers that reveal the CFO of a received frame, which was developed and executed by myself, while the modified firmware for the Asus RT-AC86U wireless router was provided by Francesco Gringoli to expose selected PHY registers in a retransmitted frame. The features presented in Section 5.2.2, Section 5.2.3 and Section 5.2.4 were developed by Francesco Gringoli. Section 5.3 presents a functional validation with testbed experiments that were designed and conducted by myself after consultation with Francesco Gringoli. Section 5.4 presents testbed experiments that were designed and conducted by myself, while the results were jointly discussed with Francesco Gringoli and Matthias Hollick.

Chapter 6 presents TVZF equalization, which was devised, implemented and evaluated by myself, while Matthias Hollick provided valuable feedback in discussions [2, 8, 10]. Section 6.3 presents a TVZF receiver for the IEEE 802.11 Non-HT PHY that extends the basic PHY implementation in Matlab provided by Matthias Schulz and Jiska Classen. Section 6.5 presents a testbed experiment with CCTs from three Asus RT-AC86U devices, for which Francesco Gringoli provided the modified firmware. Section 6.7 refers to DNN-based TVZF extensions that were developed by Vladimir Roskin in the course of his bachelor thesis and his master thesis under my supervision.

Chapter 7 presents a comprehensive discussion that reflects my views on the various aspects of this thesis, while also taking results into account that originated from the aforementioned collaborations [9, 1, 2, 8, 10]. Section 7.3.4 discusses experiments on video streaming through CCT-based network flooding that were conducted by Niklas Müller in the course of his bachelor thesis under my supervision. Niklas Müller implemented the video source and the video receivers on commercial devices. Chapter 8 summarizes the conclusions of the individual chapters of this thesis [9, 1, 2, 8, 10].

Part I

INTRODUCTION

The introduction proposes dURLLC as a new communication paradigm for interactive cyber-physical systems and proposes CCT-based network flooding and extensive broadcasting as enablers of this paradigm. Chapter 1 outlines the challenges to facilitate CCT-based communications and highlights the goals and the contributions of this thesis. Chapter 2 presents related work with a focus on concurrent transmission, synchronization and mechanisms that might be compatible with the solutions and the prototypes of this thesis.

1 INTRODUCTION

Ultra-reliable low-latency wireless communications is becoming a key driver for future innovations. While improvements in electronics and system design have been driving exponential increases in data rates for decades, technological advances should not only sustain this trend but also seize opportunities to widen the spectrum of applications. The ubiquitous presence of embedded devices, their computational power and their connectivity call for solutions that go beyond the mere provisioning of higher data rates. Ultra-reliable low-latency communications is the enabler of distributed control algorithms and decentralized computation, bringing unprecedented possibilities for the interaction between humans, physical objects and their environment.

Modern manufacturing procedures facilitate the integration of computational power and communication interfaces into sensors, actuators and virtually any type of machine. Interconnecting machines and devices wirelessly and letting them become an integral part of the environment will allow for novel applications with a high degree of interactivity. Such applications will include, for instance, industrial automation, autonomous driving, intelligent electric power distribution and remote healthcare [74]. While humans will continue to use mobile devices as interfaces to access the Internet, the communication patterns of interconnected machines will be different, as they have to coordinate with each other to accomplish complex tasks in a shared environment. In industrial automation, the sensor readings from a device may have to reach several locations for interrelated control algorithms [61]. In autonomous driving, cars have to carry out route and traffic planning, lane-keeping and collision avoidance in cooperation with other cars [261]. A self-driving car may also be part of a platoon, i.e., a convoy driving energy-efficiently in close formation, and may have to coordinate with other cars to adapt the safety distance according to traffic and road conditions [7, 141]. With swarm intelligence, multiple agents may perform a task collectively in cooperation [151]. These applications and scenarios have in common that the algorithms running on distributed machines are tightly coupled through communication technology. The messages from one device may have to reach multiple recipients, and each recipient may also rely on data from several other devices. Hence, methods for quick and reliable data dissemination are required.

Wireless communications is an appropriate means to deliver data to multiple nodes at once through broadcasting. However, current technologies like IEEE 802.11 [104] can meet strict latency demands only when an access point (AP) manages traffic with a medium access control (MAC) mechanism that is specialized for low-latency communications [71]. While wireless multi-hop networks (WMNs) can relay data also over larger distances, they usually employ IEEE 802.11 with a MAC mechanism based on carrier sense multiple access with collision avoidance (CSMA/CA), which introduces random waiting times between transmissions to avoid collisions by chance [27, 104]. Hence, WMNs incur delays in the range of tens of milliseconds to a few seconds due to queueing, waiting and packet loss [28, 143, 226]. Further, WMNs use broadcasting for route discovery and relaying, which limits their scalability due to broadcast storms [171, 226].

Cellular networks may seem more suitable for low-latency communications due to their centralized control and their broader coverage. In fact, the 3rd Generation Partnership Project (3GPP) introduces Ultra-Reliable Low-Latency Communications (URLLC) as one of three main service categories of fifth generation (5G) networks, besides enhanced Mobile BroadBand (eMBB) and massive Machine Type Communications (mMTC) [3]. The 3GPP states that a "general URLLC reliability requirement for one transmission of a packet is $1 - 10^{-5}$ for 32 bytes with a user plane latency of 1 ms", where the term *reliability* refers to the success probability and the term *user plane latency* is "the time it takes to successfully deliver an application layer packet/message from the radio protocol layer 2/3 SDU ingress point to the radio protocol layer 2/3 SDU egress point via the radio interface in both uplink and downlink directions" [3, Section 7]. Still, this guarantee can only be given in areas with cellular network coverage, and a base station's capacity for URLLC may eventually be limited according to the amount of resources that can be reserved for uplink (UL) configured grant (CG) physical uplink shared channel (PUSCH) transmissions [126]. Hence, to ensure operation everywhere and anytime, applications like vehicular networks have to stick to decentralized ad-hoc communications [26, 141]. Nevertheless, communications should work reliably with a large number of devices.

This thesis proposes decentralized URLLC (dURLLC) to facilitate the quick and reliable dissemination of messages through broadcasting and multi-hop relaying by means of distributed nodes, without routing data through cellular infrastructure. The coverage should be variable and may range from a few hops to tens or even hundreds of hops. To this end, this thesis explores methods and solutions that are based on the principle of concurrent cooperative transmission (CCT). Instead of mitigating collisions, CCT draws on the interference of multiple transmitters to eliminate waiting times during the medium access, allowing for quick message dissemination in an extensive area. In particular, distributed nodes in a WMN can form a so-called opportunistic large array (OLA) [202] upon the reception of a message to relay it further through a CCT. In other words, when a node receives a message, it retransmits it immediately, and as multiple nodes retransmit the message concurrently, their signals should superimpose such that the next-hop neighbors can decode the message. The next-hop neighbors, in turn, can form another OLA and relay the message further through another CCT, which leads to the principle of CCT-based network flooding. Glossy [73] shows that this principle works through accurate transmission timing with IEEE 802.15.4 [107], which has a physical layer (PHY) based on direct-sequence spread spectrum (DSSS) technology. However, Glossy cannot meet the URLLC requirement since a single transmission of 32 bytes already takes longer than 1 ms at 250 kbit/s with the IEEE 802.15.4 PHY [107]. Further, to facilitate applications of the Tactile Internet [74], low-latency wireless networks are envisioned to meet a round-trip time (RTT) of below 1 ms, i.e., even quicker message delivery and tighter scheduling are required.

This thesis explores the feasibility of CCT with broadband communications through orthogonal frequency-division multiplexing (OFDM) [243] and presents solutions that comply with selected IEEE 802.11 PHYs [104]. The OFDM-based IEEE 802.11 PHYs do not only offer higher data rates than the IEEE 802.15.4 PHY, but they also allow for a shorter transmission duration of each CCT. Thus, this thesis aims at a significant latency reduction in combination with capacity and reliability increases through message dissemination via CCT to make dURLLC applications possible.

The remainder of this chapter is organized as follows. Section 1.1 highlights a trend that reduces latencies by moving communication endpoints into proximity of each other and shows that dURLLC is an extension of this trend. Section 1.2 showcases scenarios with CCT-based network flooding and extensive broadcasting. Section 1.3 outlines the challenges of CCT with OFDM and defines corresponding goals. Section 1.4 highlights the contributions of this thesis. Section 1.5 gives an overview of the overall thesis.

1.1 Latency Reduction by Proximity

A prominent method to increase reliability and to reduce latency in computer networks is to bring communication endpoints into proximity of each other. In particular, the services that users interact with are often brought into a region close to the respective user devices. This section shows how this trend materializes in mobile networking and highlights the proposed dURLLC paradigm as an extension of this trend. Figure 1 illustrates scenarios at four different levels of proximity between mobile devices and their communication partners: wireless Internet access, mobile edge computing (MEC), URLLC and dURLLC, the latter of which is the paradigm proposed and explored in this thesis.

A major role of wireless communications is to serve the last hop of Internet connectivity. Mobile devices use network technologies such as Long-Term Evolution (LTE) or Wi-Fi to connect to a base station or an AP, from where they gain Internet access through an Internet service provider (ISP) network. With the advent of messaging services and videoconferencing, Internet traffic became increasingly interactive. The data associated with such applications are time-sensitive, i.e., they are most valuable to their users when they are timely and reliably delivered. Consequently, network operators prioritize certain traffic, such as Internet Protocol (IP) telephony, to provide a high quality of service (QoS) to their customers. QoS is typically concerned with throughput, latency, jitter, reliability in terms of packet delivery and packet ordering. Some QoS goals, however, are conflicting. For instance, provisioning buffers at intermediate routers can help to reduce packet loss, but it increases latency when queues build up.

As opposed to humans communicating interactively with each other, most websites and Internet services are not tied to specific geographical locations, so their reliability and latency can be improved by bringing them closer to users. Early adopters of this concept were content delivery networks (CDNs) [60, 230]. They comprise servers at various geographical locations to deliver cached versions of content, such as websites and files. When a user requests a piece of content, the CDN automatically selects a server in proximity of the respective user to deliver it. In doing so, CDNs reduce latency, increase reliability and mitigate Internet traffic. The network region in which CDNs operate their servers is often referred to as the *edge*. Rather soon after their launch, CDNs also started to open up their infrastructures to let customers run applications on the edge servers by means of virtualization technologies, which led to the concept of *edge computing* [60].

MEC is an emerging technology in 5G cellular networks that defines the radio access network (RAN) as the edge from the perspective of mobile devices. By integrating edge servers into the RAN infrastructure, MEC enables service operators to dynamically instantiate workloads on computing resources in the mobile edge on demand. With this, services can be elastically scaled and data can be immediately processed in proximity to users to improve reliability and response times [5, 99, 145].

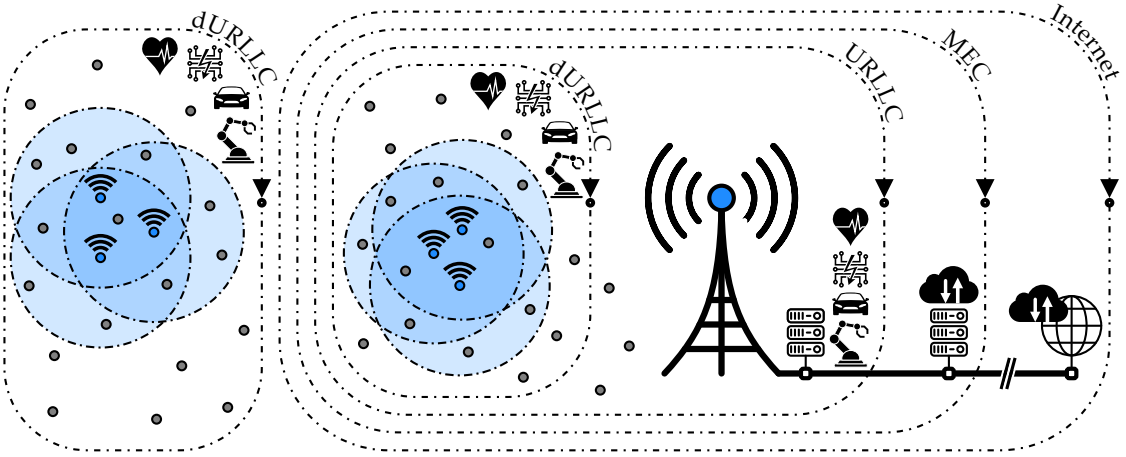


Figure 1: Overview of low-latency communication paradigms and their respective proximity between mobile devices and their communication partners: Internet access is subject to QoS policies in the core network. MEC brings applications and services closer to mobile devices. URLLC facilitates time-critical transmission scheduling at base stations. dURLLC disseminates time-sensitive data in areas around their origin.

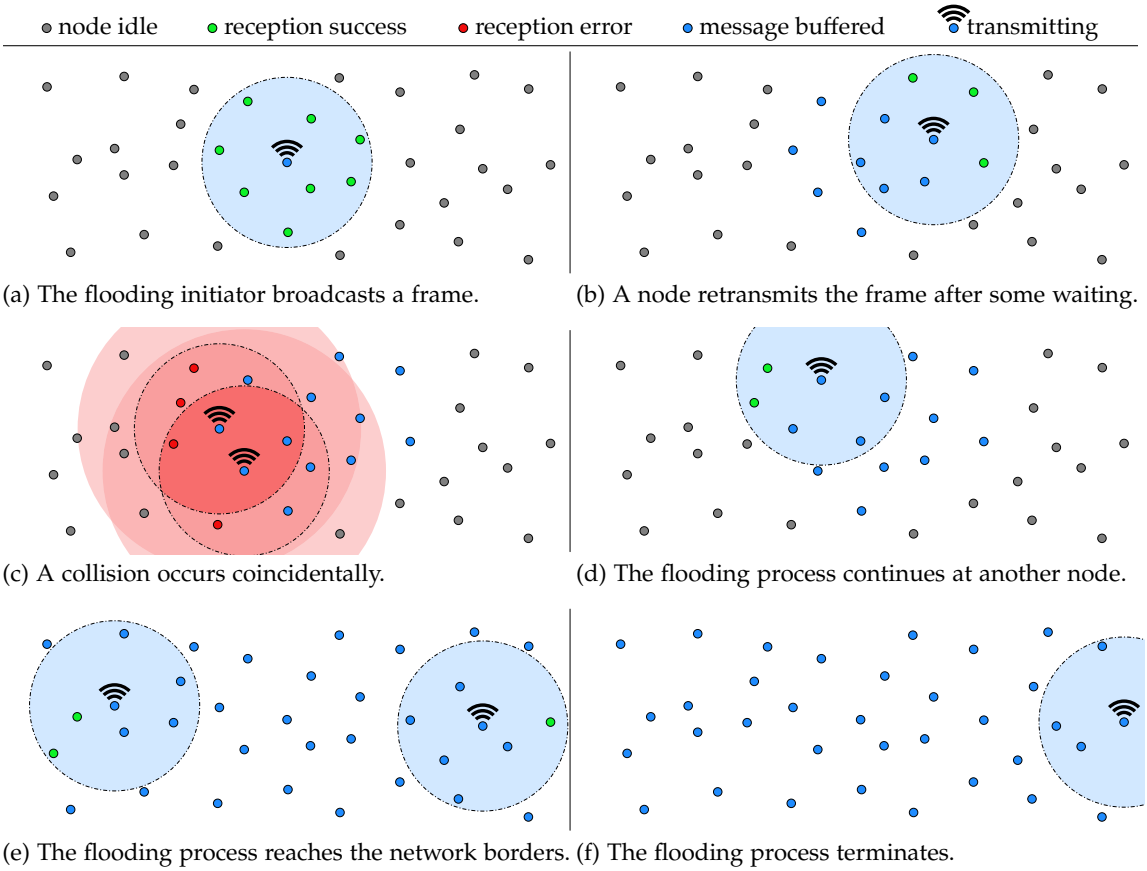
With URLLC, 5G cellular networks shall provide real-time access of data across mobile devices to carry out control algorithms and interactive tasks. To meet the targeted user plane latency of 1 ms in combination with a reliability of $1 - 10^{-5}$ [3], a base station must serve packets through advanced PHY techniques and tight scheduling [14, 90, 126]. The integration of URLLC into base stations essentially cuts down the travel path of time-sensitive data exchanged between mobile devices.

The proposed dURLLC paradigm is an extension of the outlined trend and reduces the travel path of time-sensitive data even further. In particular, such data are not routed through a cellular infrastructure, but mobile devices distribute them autonomously in an area of flexible size around the origin of the data. As time-sensitive data are often relevant for multiple interactive agents in a shared environment, CCT-based network flooding is a viable approach for data dissemination. Further, an infrastructure-less approach is not in conflict with cellular infrastructures. A base station could coordinate the high-level scheduling for CCT-based network flooding in different regions, while the actual flooding processes would take place in parallel. If the infrastructure fails or if there is no cellular coverage, the mobile devices could perform a transition [8] into a decentralized high-level MAC mechanism and coordinate their flooding schedules autonomously.

1.2 Data Dissemination via Concurrent Cooperative Transmission

The dURLLC paradigm proposed in this thesis is driven by CCT with distributed devices in variable-sized areas. This section outlines how this paradigm works in different scenarios and compares it with conventional data dissemination with a MAC based on CSMA/CA, such as the IEEE 802.11 distributed coordination function (DCF) [27, 104]. CCT essentially offers two fundamental advantages over CSMA/CA:

- Spectral resources are better utilized by leveraging interference.
- Waiting times for transmissions are minimized to speed up message propagation.



(a) The flooding initiator broadcasts a frame.
(b) A node retransmits the frame after some waiting.
(c) A collision occurs coincidentally.
(d) The flooding process continues at another node.
(e) The flooding process reaches the network borders.
(f) The flooding process terminates.

Figure 2: Conventional network flooding with a MAC based on CSMA/CA

1.2.1 *Network Flooding*

Network flooding is the process of data dissemination from one node to all others through relaying over multiple hops. Network flooding is used, for instance, in WMN routing protocols for route discovery [41, 55, 56, 98, 132] and in wireless sensor networks (WSNs) for data dissemination [134, 197], time synchronization [73, 146, 255] and routing [75, 83].

Figure 2 illustrates conventional network flooding with a CSMA/CA-based MAC. In Figure 2a, the initiator broadcasts a message that shall be flooded to all other nodes. Due to the broadcast characteristic of the wireless channel, the message is received by multiple neighbors that are marked green. Each recipient relays the message, but due to the collision avoidance feature of CSMA/CA, they wait a random amount of time before transmitting, e.g., up to 9207 µs with IEEE 802.11g [104, Section 10.3.3 and Table 17-21]. In Figure 2b, the first node retransmits the message after some waiting and reaches nodes in the 2-hop neighborhood of the initiator. Nodes that have already received the broadcast message are marked blue. However, it can happen that two nodes draw the same waiting time and retransmit the message simultaneously, as shown in Figure 2c. The receivers typically cannot recover the data from the superimposed signals in such a collision, as indicated with red color. Further, the interference range of a transmission exceeds the communication range, so the nodes that are in communication range of only one of the two transmitters cannot decode the transmission from their neighbor.

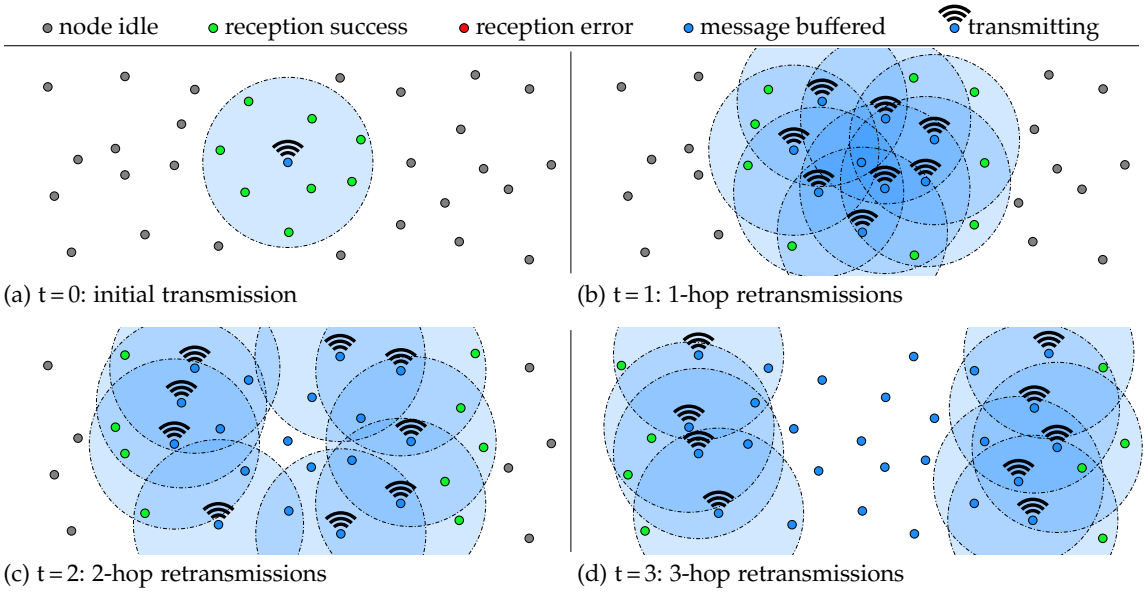
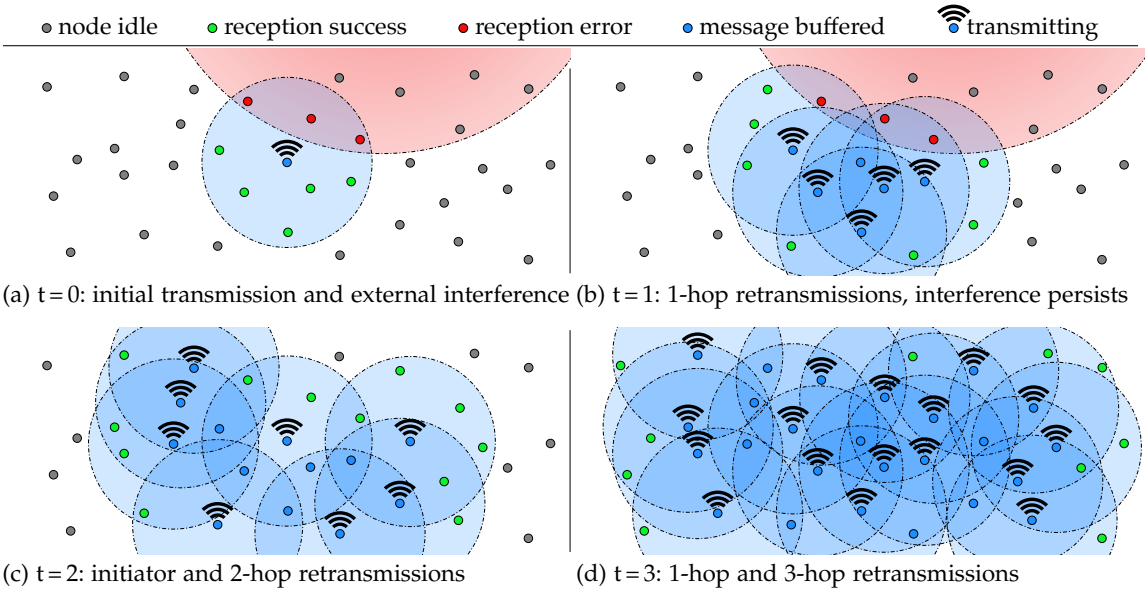


Figure 3: CCT-based network flooding

Still, a collision may eventually be recovered by a transmission from another node that has not relayed the message yet, as shown in Figure 2d. In Figure 2e, the flooding process reaches the network borders left and right after some more time. Since each node retransmits the message, the flooding process continues after all nodes have received it, as illustrated in Figure 2f. The overall flooding process may take a rather long time and use a considerable amount of spectral resources. Hence, it may cause a higher level of medium contention and more collisions, leading to broadcast storms [171, 226].

Figure 3 illustrates how CCT-based network flooding can substantially enhance the latency per hop and the utilization of the wireless medium. In Figure 3a, a node broadcasts a message for flooding as in the previous example. However, Figure 3b shows that all of the 1-hop neighbors retransmit the message immediately after the reception, i.e., without backing off, so that their signals superimpose. Then, all the 2-hop neighbors recover the message successfully and instantly continue with a retransmission in Figure 3c. The 3-hop neighbors retransmit the message accordingly in Figure 3d, so that the flooding process reaches the network borders. The flooding process may stop at this point if it is configured to propagate only 3 hops. Note that CCT-based network flooding does not assume collisions as in Figure 2c of the previous example. Instead, the receivers are assumed to recover the data from a CCT with mutually interfering signal components. To this end, special PHY techniques are required, which are the subject of this thesis.

Another important aspect is that latency and reliability remain conflicting goals at the wireless PHY. The reliability in terms of the frame reception rate (FRR) can generally be increased by using lower modulation rates and coding schemes with more redundancy. However, these measures increase the transmission duration and thus the latency per hop. Another method effective against spatio-temporal distortions is the use of repetitive transmissions, i.e., each node retransmits a flooding frame upon each reception up to a certain number of times. With this, a flooding frame is immediately forwarded by the nodes that receive it successfully upon their first reception opportunities, whereas other nodes get additional chances to join the CCT-based flooding process shortly afterwards.



(a) t = 0: initial transmission and external interference (b) t = 1: 1-hop retransmissions, interference persists

(c) t = 2: initiator and 2-hop retransmissions (d) t = 3: 1-hop and 3-hop retransmissions

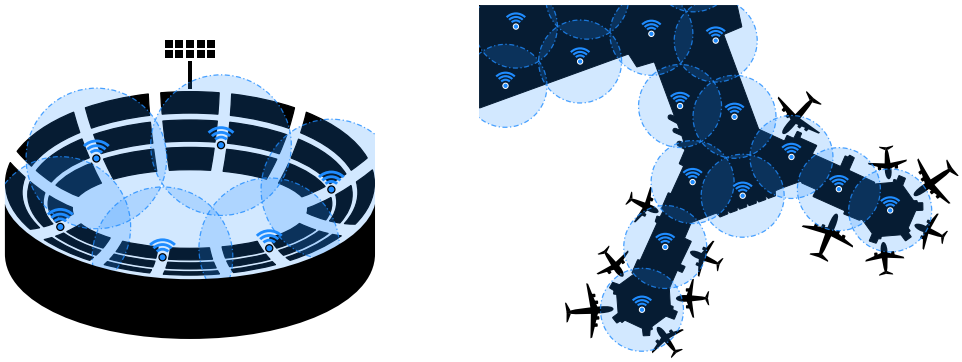
Figure 4: Repetitive transmissions can improve the reliability of CCT-based network flooding. Top: an external interferer distorts three nodes. Bottom: The flooding process recovers more quickly and persistently as each node retransmits the message twice.

Figure 4 illustrates CCT-based network flooding while an external interferer impedes receptions temporarily at the nodes marked red. A node initiates a flooding process in Figure 4a, receives its own message in Figure 4b and transmits it another time in Figure 4c. When the interference is gone in Figure 4c, all three previously jammed nodes receive the message. In Figure 4d, the transmitters from Figure 4b transmit for the second time and the receivers from Figure 4c transmit for the first time. As in the previous example, the flooding process reaches the borders left and right with the 3-hop retransmissions.

1.2.2 *Extensive Broadcasting*

CCT may be employed for extensive broadcasting in networks with a coverage exceeding the communication range of a single AP. Figure 5 illustrates examples of such scenarios. In the stadium scenario in Figure 5a, the APs cooperatively broadcast video content for the visitors of a live sports event, such as slow-motion replays. In the airport scenario in Figure 5b, the APs cooperatively broadcast information for travellers and workers. By transmitting data frames via CCT from all APs in dedicated time slots, the wireless spectrum is better utilized, the coverage is enhanced and medium contention is mitigated in comparison to conventional broadcasting with a CSMA/CA-based MAC.

Further applications for extensive broadcasting comprise automation scenarios in which a controller encodes commands in an aggregated or abstract format, so that multiple recipients are served with the same messages. While an aggregated format may contain several commands in one message, an abstract format may describe an overall target state of a system, such that the receivers have to determine by themselves how they contribute to the specified goal. For example, a traffic control system for autonomous vehicles in a city could broadcast general traffic guidance messages, so that each car derives its own driving behavior from these messages as a function of its own context.



(a) Stadium: live video and slow-motion replays (b) Airport: information for travellers and workers

Figure 5: CCT-based extensive broadcasting scenarios: area-specific information for humans and machines is transmitted concurrently by multiple access points.

1.3 Challenges and Goals

Data dissemination through CCT has several advantages over conventional broadcasting with a CSMA/CA-based MAC, as shown in Section 1.2. However, CCT-based network flooding and extensive broadcasting necessitate that a receiver can decode the resulting superimposed signals. Glossy [73] attains such receptions with IEEE 802.15.4 by achieving transmitter time synchronization better than 0.5 µs. Still, the IEEE 802.15.4 PHY is based on DSSS and offers a data rate of only 250 kbit/s in the 2.4 GHz band [73, 107].

To reduce the latency and to increase the capacity of CCT-based network flooding in comparison to the IEEE 802.15.4 PHY, higher data rates are required as they do not only increase the throughput but also reduce the transmission duration of a frame. Since the propagation speed of CCT-based flooding is governed by the duration of each transmission, higher data rates raise the prospect of a significant latency reduction. Hence, the overarching goal of this thesis is to explore the feasibility of CCT with broadband communications through OFDM [243]. In particular, to work towards practical solutions, CCT shall be studied with IEEE 802.11, the primary non-cellular broadband communication standard for mobile devices [12, 104]. In fact, variations of IEEE 802.11 are going to be used in future intelligent transportation systems (ITSs) [262], which makes the standard a reasonable choice as a basis for advancing research.

1.3.1 *Analysis*

Since OFDM is fundamentally different from DSSS, Glossy's approach to facilitate CCT through transmitter time synchronization better than 0.5 µs [73] may not apply for OFDM. On the one hand, DSSS uses a single carrier and encodes each bit with a sequence of chips to improve robustness [93]. On the other hand, OFDM is a multi-carrier scheme that copes with frequency-selective fading of broadband channels by encoding data on orthogonal narrow-band subcarriers in parallel [243]. Therefore, to make CCT work also with higher data rates, one goal is to understand under what conditions a superimposed OFDM signal can be decoded and how these conditions differ from the ones for DSSS. Specifically, the IEEE 802.11 standard [104] comprises both DSSS and OFDM PHYs that shall be analyzed and compared under these aspects.

1.3.2 *Fulfilling the Requirements*

The requirements to enable CCT with OFDM-based IEEE 802.11 PHYs concern certain properties of the signals superimposing at a receiver which fall into three categories. First, there are properties specific to the data frame, such as the frame length and the data rate according to the employed modulation and coding scheme (MCS). Second, physical effects and the environment manipulate signals, e.g., through noise, interference and multi-path propagation effects. Third, certain properties are specific to each sender, such as its transmission power, the exact time at which its transmission starts and its carrier frequency and oscillator jitter. To generate signals with multiple transmitters whose superposition can be decoded with an IEEE 802.11 receiver, the transmitters have to achieve a certain level of synchronization in time and frequency in accordance with the other parameters. Therefore, another goal is to develop wireless synchronization techniques that work cooperatively between nodes, ideally without explicit coordination and without feedback to save time and spectral resources. In CCT-based network flooding, the transmitter synchronization can, for instance, draw on frame receptions from the previous hop, similar to Glossy [73]. In addition to synchronizing with each other while transmitting, the nodes shall also contribute to the stability of a flooding process by maintaining the operational parameters within specific ranges over multiple hops.

1.3.3 *Challenging the Existing Requirements*

The targeted requirements for transmitter synchronization are based on the assumption that the CCTs are decoded by state-of-the-art receivers. However, it may not always be possible to meet these requirements. For instance, a node may fail to synchronize accurately during a CCT frame reception due to poor signal quality. Further, CCT-based extensive broadcasting mechanisms cannot employ the same synchronization techniques as CCT-based network flooding mechanisms due to the lack of common frame receptions. While this problem could be solved through wired synchronization, this is an expensive and inflexible solution. Hence, to push the limits of the technical feasibility, another goal is to challenge the requirements imposed at the receiver's end. In particular, decoding technique enhancements shall make a receiver more robust to a poor synchronization between concurrent transmitters. With this, scenarios shall be supported in which precise synchronization between multiple transmitters fails or cannot be achieved.

1.3.4 *Testbed Validation*

While mathematical modeling is crucial to formulate problems and solutions, it still makes idealizing assumptions to a certain extent, neglecting the intricacies and subtleties of the physical world. Especially in complex systems with cooperating devices, deficiencies of electronic components and interference can have negative ramifications. To provide practical solutions, it is indispensable to validate mechanisms in testbed experiments. Hence, a methodological goal is to use software-defined radios (SDRs) for prototyping. In particular, novel designs and algorithms shall be developed in software and be validated both in simulations and with transmissions via SDRs. Further, the compatibility with commercial IEEE 802.11 hardware shall also be tested.

1.3.5 *Prototype Design*

Transferring signals with SDRs in a physical environment and processing the received in-phase quadrature (IQ) samples in software allows for an assessment of mechanisms at specific operating points under realistic conditions. Yet, this technique captures only isolated transmissions and does not reflect how a system responds to its own CCTs. Therefore, another goal is to devise a prototype with selected mechanisms on an SDR with a field-programmable gate array (FPGA). The prototype design shall allow for a performance evaluation of multiple nodes generating successive CCTs in real-time. To facilitate a range of applications, a low-level MAC protocol shall be designed that provides an interface to configure the behavior of the nodes through the flooding frame that they process. With this, the flooding initiator shall be empowered to tune the latency, the reliability and the area coverage without overhead. Each node shall be able to detect and to discard duplicate receptions of a frame so that each flooding process is terminated. Further, the prototype shall monitor the transmissions of other nodes and be able to sustain a flooding process when the next-hop neighbors fail to retransmit a message. Even though making changes to the IEEE 802.11 MAC, the prototype shall retain PHY compatibility so that commodity devices can receive the messages.

1.4 Contributions

In order to address the aforementioned challenges and goals, this work comprises the following contributions in the research area of CCT with IEEE 802.11 for dURLLC.

1.4.1 *Analysis*

The contributions start with an analysis that explores the conditions under which CCTs with the IEEE 802.11 DSSS PHY and the OFDM-based IEEE 802.11 PHYs can respectively be decoded by state-of-the-art receivers. The analysis particularly identifies the conditions under which the interference becomes destructive, so that the received signal is canceled, which marks an edge case for the decodability. The analysis leads up to theoretical results that are validated in simulations and in testbed experiments.

On the one hand, the IEEE 802.11 DSSS PHY is found to be robust to interference when the transmitters of a CCT have a slight time offset (TO) with respect to each other. However, when the transmitters are synchronized in time, their carrier frequencies must also be synchronized. On the other hand, the OFDM-based IEEE 802.11 PHYs are found to require the transmitters of a CCT to be synchronized in both time and frequency. The time synchronization must undercut the clearance of the guard interval (GI) that remains after subtracting the channel delay spread (CDS). The frequency synchronization requirement depends on the frame length, the MCS and the signal-to-noise ratio (SNR). The analysis also takes the phase tracking mechanism and the channel equalization into account to identify conditions that lead to symbol errors on specific subcarriers, which can be corrected to a certain amount through forward error correction (FEC). For example, to successfully decode an IEEE 802.11 Non-HT frame with a MAC service data unit (MSDU) of 112 bytes at 24 Mbit/s, the carrier frequency offset (CFO) between the concurrent transmitters must not exceed ± 2 kHz in the 2.4 GHz band.

1.4.2 *Synchronization through Frame Receptions*

To generate coherent interference (CI) in CCT-based network flooding, this work presents techniques that leverage common frame receptions to synchronize devices wirelessly in time and frequency. IEEE 802.11 OFDM receivers generally synchronize in time and frequency to the transmitter during a frame reception by means of the frame preamble to mitigate intersymbol interference (ISI) and intercarrier interference (ICI), respectively [104]. The conventional preamble-based CFO estimate is enhanced in this work by leveraging the pilot subcarriers, which are obtained after applying a fast Fourier transform (FFT) on the received OFDM symbols. This technique reduces the impact of noise on the CFO estimate, especially for frames with a long duration. In testbed experiments, the absolute error of the enhanced CFO estimate is typically below 10 Hz for frames with an MSDU of 1508 bytes, whereas the preamble-based CFO estimate typically deviates by hundreds of hertz and more, depending on the SNR. The enhanced CFO estimation technique is also integrated into the real-time prototype described below.

1.4.3 *Real-Time Prototypes*

To facilitate CCT-based network flooding in real-time, this work comprises a prototype that is based on the Wireless Open Access Research Platform (WARP) v3 SDR [246]. The WARP prototype can retransmit a received frame with precise synchronization in time and frequency with respect to the received frame itself. Thus, multiple WARP prototype nodes can generate a CCT with CI by executing the same procedure upon a common frame reception, which in turn may also arrive as a CCT.

The WARP prototype extends the IEEE 802.11 reference design for WARP v3 [247] by signal processing algorithms in the FPGA and by a low-level MAC protocol running on a central processing unit (CPU). The frequency synchronization mechanism comprises an enhanced CFO estimator integrated into the PHY receive (Rx) core [9] as well as a frequency shifter integrated into the PHY transmit (Tx) core [9]. Testbed experiments show that the $[10\,\%, 90\,\%]$ quantiles of the phase error are within $\sim [-0.021\,\pi, 0.012\,\pi]$ at the end of a frame with up to ~ 100 OFDM symbols. The $[10\,\%, 90\,\%]$ quantiles of the relative CFO are within $\sim [-11.8, 14.6]$ Hz at 220 OFDM symbols.

The low-level MAC protocol fields are integrated into the IEEE 802.11 MAC header, allowing commercial devices to receive and inject flooding frames from and into a WMN. The WARP prototype processes the low-level MAC protocol in real-time, so that the behavior of a WMN during a flooding process is defined by the flooding frame itself. The initiator of a flooding process can tune the latency, the reliability and the number of hops through the low-level MAC protocol together with the MSDU size and the MCS. In particular, the reliability of a flooding process can be tuned by making each node retransmit a flooding frame multiple times. Further, a WARP prototype node can retry to reach its next-hop neighbors if it does not detect an expected retransmission from them. The highest possible PHY data rate with a 20 MHz wide channel can be achieved with the HT PHY at MCS 7 and is 65 Mbit/s. The lowest possible latency per hop is 44.2 µs, achievable with the Non-HT PHY at MCS 5 or higher (see Table 2). By allowing for certain performance tradeoffs, the low-level MAC protocol can enable dURLLC applications.

Various testbed experiments are conducted with two groups of nodes that forward a flooding frame to each other in a ping-pong fashion, allowing a limited number of nodes to generate a large number of successive CCTs. With five nodes per group, flooding processes with 10 000 successive CCTs can be reliably triggered with all MCS indices and MSDU sizes. Commercial devices often have a better FRR for CCTs from five nodes than for transmissions from a single node, indicating that the WARP prototype nodes successfully keep up their joint synchronization in time and frequency over time.

The feasibility of the URLLC requirement is assessed, which is defined by the 3GPP as the delivery of a 32-byte MSDU with a reliability of $1 - 10^{-5}$ at a latency of 1 ms [3]. Under non-line-of-sight (NLOS) conditions and with a single retransmission per node, the URLLC requirement can be met with an IEEE 802.11 Non-HT PHY with up to MCS 2, for which the latency per hop is 80.2 µs. Further, under line-of-sight (LOS) conditions and with a single retransmission per node, the URLLC requirement can be met with an IEEE 802.11 Non-HT PHY with up to MCS 4, for which the latency per hop is 60.2 µs. CCT-based network flooding is also shown to be feasible in an office environment through circular forwarding of a flooding frame between four groups of WARP prototype nodes.

Another prototype based on the Asus RT-AC86U wireless router is created through firmware modifications to showcase the feasibility of CCT also with commodity hardware. This prototype leverages registers of its IEEE 802.11 chip to estimate the CFO during frame receptions and to tune its carrier frequency through a voltage-controlled oscillator (VCO). Further, the Asus RT-AC86U prototype leverages the acknowledgment (ACK) engine of its IEEE 802.11 chip in order to retransmit a received flooding frame with accurate timing.

Testbed experiments show that the URLLC requirement is also met with the Asus RT-AC86U prototype, at MCS 0 and MCS 2 under NLOS conditions and up to MCS 6 under LOS conditions. Moreover, this prototype even allows assessing the feasibility of CCT with CI with up to four multiple input multiple output (MIMO) spatial streams with the IEEE 802.11 HT PHY at a channel bandwidth of 40 MHz in the 5 GHz band. In an experiment in which two groups of nodes generate CCTs with a 32-byte MSDU in a ping-pong fashion, a LOS receiver that accepts receptions from any of both groups has an FRR of 99.5 % at a PHY data rate of 324 Mbit/s. Further, with a fixed PHY data rate, the FRR is typically higher when employing a larger number of MIMO spatial streams, so that the URLLC reliability requirement can be fulfilled with less transmissions.

1.4.4 *Time-Variant Zero Forcing*

Another contribution of this work is time-variant zero-forcing (TVZF), an equalization technique that relaxes the requirement for accurate frequency synchronization in a CCT. When the transmitters are not synchronized in frequency, their CFOs introduce different phase drifts whose effects superimpose at the receiver, causing time-variant interference. TVZF analyzes, reconstructs and equalizes these interference patterns by means of the received superposition of signal components to enable the decoding of CCT frames that would otherwise lead to collisions and frame loss with state-of-the-art receivers. With this, TVZF allows for a recovery of frames when the frequency synchronization fails during CCT-based network flooding. Furthermore, TVZF facilitates dURLLC scenarios in which a transmitter synchronization through a common frame reception is infeasible, such as the CCT-based extensive broadcasting scenario (see Section 1.2.2).

This work presents a TVZF receiver design for the IEEE 802.11 Non-HT PHY that is prototyped in Matlab and that comprises several processing steps to reconstruct the time-variant interference of a CCT. First, the TVZF receiver performs a symbol timing estimation by means of successive interference cancellation (SIC) to mitigate ISI. Second, it performs a Fourier analysis of the symbol traces on the four pilot subcarriers by means of a technique called iterative interference cancellation (IIC) in order to estimate certain parameters of the detected CCT signal components. Third, it estimates the channel coefficients of each detected CCT signal component by means of an interpolation-based technique under the constraint of the measured superposition of the channel coefficients. Finally, the TVZF receiver reconstructs a vector of time-variant channel coefficients and equalizes the received superposition of the CCT signal components.

The TVZF receiver for the IEEE 802.11 Non-HT PHY is evaluated in simulations and testbed experiments with WARP SDRs [246] that are connected to a Matlab workspace via WARPLab [250]. The testbed experiments comprise combined parameter sweeps of the TOs and the CFOs of up to four concurrent transmitters in LOS and NLOS, respectively. For TOs of up to the duration of the GI and for CFOs within $[-5, 5]$ kHz, the TVZF receiver has an average FRR of 97 %, 78 % and 49 % for two, three and four concurrent transmitters, respectively. In another testbed experiment with CCTs from three commercial devices that are not synchronized in frequency, the TVZF receiver attains an FRR of up to 92 %. Moreover, simulations show that TVZF equalization enhances the reception performance for a wide range of relative signal power levels in a CCT.

1.5 Outline

This thesis comprises four parts: the introduction, the contributions on generating CCTs with coherent interference (CI) through transmitter synchronization in time and frequency, the contributions on decoding CCTs with incoherent interference and the discussion.

The first part presents the motivation and the goals of this work. Chapter 1 highlights the importance of low-latency wireless communications in cyber-physical systems and proposes dURLLC as a new communication paradigm that shall be enabled through CCT-based network flooding and extensive broadcasting. Subsequently, the challenges, the goals and the contributions of this thesis are outlined. Chapter 2 discusses related work on concurrent transmissions, synchronization techniques and mechanisms that might interoperate well with CCT-based solutions. In addition, it provides background information on the IEEE 802.11 standard that is relevant in the context of this thesis.

The second part presents the contributions of this work that address CCT with CI through precise transmitter synchronization in time and frequency. Chapter 3 presents an analysis of the exact conditions under which CCTs with the IEEE 802.11 DSSS and OFDM PHYs can be decoded, respectively. Chapter 4 presents a prototype that is based on the WARP v3 [246] and that enables CCT-based network flooding in real-time through reception-based synchronization in time and frequency. Chapter 5 presents a prototype that is created through firmware modifications of the Asus RT-AC86U wireless router and that allows for reception-based pre-synchronization in frequency and for generating retransmissions in real-time. Both prototypes are validated in various testbed experiments to characterize their synchronization accuracies. Further, various testbed experiments are conducted to assess the feasibility of generating successive reception-triggered CCTs.

The third part presents the contributions of this work that address CCT with incoherent interference to support dURLLC scenarios in which accurate synchronization may fail. Chapter 6 presents TVZF, an equalization technique that enables a receiver to deal with incoherent interference. A receiver design for the IEEE 802.11 Non-HT PHY is presented that extracts the required information from a superimposed signal with an unknown number of concurrent transmitters to perform TVZF equalization. The TVZF equalization technique is integrated into a custom Matlab receiver for IEEE 802.11 Non-HT data frames. Various simulations and testbed experiments by means of WARP SDRs are conducted to assess the characteristics and the performance of the TVZF receiver.

The fourth part comprises a discussion of the proposed dURLLC paradigm that is facilitated by means of CCT-based communications and provides concluding remarks. Chapter 7 discusses the characteristics of CCT signals and of CCT-based network flooding, the core technologies that enable CCT-based communications and technology extensions that can potentially enable even more advanced dURLLC applications and scenarios. Chapter 8 summarizes the conclusions of this thesis.

2 BACKGROUND AND RELATED WORK

This chapter discusses related work on the proposed decentralized URLLC (dURLLC) paradigm that should be enabled through concurrent cooperative transmission (CCT) in network flooding (see Section 1.2.1) and extensive broadcasting (see Section 1.2.2). Section 2.1 starts with related work on mechanisms that attempt to mitigate the impact of collisions or that intentionally draw on interference from distributed transmitters to enhance network performance by means of CCT. Section 2.2 covers fundamentals and related work on synchronization with a focus on practical techniques for wireless synchronization of distributed devices. Section 2.3 presents extended mechanisms that are based on CCT-based flooding or that might be compatible with it. Specifically these works might benefit from the significant latency reductions and throughput gains offered by the CCT techniques for IEEE 802.11 [104] presented in this thesis, compared to Glossy [73], which is designed for IEEE 802.15.4 [107]. Section 2.4 provides background information on the IEEE 802.11 physical layers (PHYs) and on fundamental processing techniques.

2.1 Concurrent Transmission

Wireless communications is fundamentally different from its wired counterpart due to the broadcast characteristic on a shared medium, i.e., a transmission occupies the wireless spectrum not only for its intended recipient but also for all other nodes in vicinity of the sender. When multiple nodes in an area start sending at the same time, their transmissions collide, i.e., the signals superimpose at nearby receivers. This undesired condition leads to distorting interference, which is often harsher than noise. This section discusses how wireless networks deal with such collisions, how collisions can be overcome in some cases through the capture effect and how synchronization can be leveraged to generate useful or coherent interference for CCT.

2.1.1 *Collisions and Medium Access Control*

The majority of wireless medium access control (MAC) protocols reduce collisions by means of randomized channel access, as introduced with ALOHA [6, 125, 194]. IEEE 802.15.4 [107, Section 6.2.5.1] and the distributed coordination function (DCF) of IEEE 802.11 [104, Section 10.3] employ carrier sense multiple access with collision avoidance (CSMA/CA) [27, 265], which works in two phases. First, a node senses the medium and refrains from a transmission if it finds the medium to be busy. Second, if the medium becomes free after a busy phase, a node waits a random amount of time before attempting a transmission in order to probabilistically avoid collisions with other nodes that employ the same MAC mechanism. Still, if at least two nodes start transmitting at the same time by chance, these nodes will not be able to detect the transmissions from the other nodes, respectively, since their own transmission will drown out any remote signals that are much weaker due to path loss and attenuation. Hence, as opposed to

carrier sense multiple access with collision detection (CSMA/CD), which is employed in wired networks such as Ethernet [102], a wireless collision lasts as long as the longest involved transmission, which reduces the network performance [27]. However, a receiver could send a collision notification in the form of a correlatable sequence that can be detected by the senders, allowing them to abort their transmissions immediately [205]. Other classes of MAC algorithms comprise, for instance, centralized control schemes, like the IEEE 802.11 point coordination function (PCF) [103, Section 10.4], reservation-based MAC schemes [140, 266] and distributed queueing (DQ) [125, 251]. In contrast to avoiding collisions, CCT leverages interference, which, however, may also require coordination at a higher level, as discussed in Section 7.3.2.

2.1.2 *Capture Effect*

In a wireless collision, the superimposing signals usually arrive with different power levels at a receiver due to signal propagation effects such as path loss and attenuation. Depending on the received signal strengths (RSSs) and the relative timings of the signals, a receiver may be able to capture the strongest signal and decode it in the presence of other interfering transmissions [194]. The conditions, under which this *capture effect* occurs, also depend on the PHY, the modulation and coding scheme (MCS) and the receiver design. In ALOHA-based networks with frequency modulation (FM) radios, the strongest signal should exceed the next strongest one by at least 1.5 – 3 dB [194]. With a frequency-shift keying (FSK) PHY, the capture effect occurs with high probability (>90 %) if the signal-to-interference-plus-noise ratio (SINR) exceeds $\sim$6 dB [216]. In IEEE 802.15.4 networks with a direct-sequence spread spectrum (DSSS) PHY, the capture threshold is at an SINR of $\sim$4 dB [81, 216]. However, the capture threshold may vary by a few decibels in practical settings as a function of the employed devices [216].

Depending on the order of the arriving signals, collisions can be categorized into *stronger-first* and *stronger-last* collisions [245]. While a receiver can typically exploit the capture effect if the strongest signal arrives first, it might not be able to do so otherwise if it has already synchronized to a weak signal, so that a later signal causes a collision. A receiver can resolve this problem by continuously searching for frame preambles after starting a reception process so that it can eventually resynchronize to a stronger signal and restart decoding [201, 238, 245]. The CC2420 radio [40] that is used by Glossy [73] is found to resynchronize to a stronger transmission if this transmission starts no later than a preamble duration after the start of a weaker signal [257, 258].

In IEEE 802.11 networks, the capture threshold depends to a large extent on the employed PHY. An analysis of the IEEE 802.11 DSSS and the IEEE 802.11b complementary code keying (CCK) PHYs indicates that the reception of a signal can be impeded by an interfering transmission that is 2 dB stronger, while the CCK PHY with 5.5 or 11 Mbit/s is just marginally more susceptible than the DSSS PHY with 1 or 2 Mbit/s [240]. This finding implies that a receiver may leverage the capture effect starting with a threshold of slightly above -2 dB. Further, practical experiments with IEEE 802.11 DSSS show that a signal that exceeds an SINR differential of about 5 dB typically leads to a capture [78, 239], even if the first sender is weak [117]. In contrast to the DSSS and CCK PHYs of IEEE 802.11, in IEEE 802.11a, which is based on orthogonal frequency-division multiplexing (OFDM), the capture threshold varies from -1 dB to 25 dB, depending on the actual scenario and

the employed data rate [127]. Generally, the transmission power of individual nodes in combination with the capture threshold are key to optimize the overall performance in wireless multi-hop networks (WMNs) [217] and infrastructure-based networks [87].

The capture effect is exploited in wireless sensor networks (WSNs) for low-latency flooding with Flash Flooding [138] and Glossy [73]. While Glossy's main contribution is to achieve useful interference through synchronization, it still relies on the capture effect if the synchronization fails [73]. Further, Chaos combines in-network processing (INP) with the capture effect to achieve a global consensus among all network nodes [124]. Even though Chaos builds on top of Glossy, it cannot leverage useful interference through synchronization since different nodes transmit different messages.

Because of the susceptibility to interference of the OFDM-based IEEE 802.11 PHYs, the mechanisms presented in this thesis are not designed to leverage the capture effect. Instead, to enable CCT under conditions of strong interference, the prototypes presented in Chapter 4 and in Chapter 5 employ synchronization techniques to achieve CCTs with coherent interference (CI), which can even improve a receiver's SINR. Furthermore, the equalization with time-variant zero-forcing (TVZF) enhances a receiver's robustness to co-channel interference beyond the capture threshold, as presented in Chapter 6.

2.1.3 *Interference Cancellation*

In addition to exploiting the capture effect for decoding data in the presence of interfering transmissions, interference cancellation is a technique that can enhance the reception performance further. With successive interference cancellation (SIC) [233], a receiver first decodes the strongest signal while treating the others as noise. Then, it re-encodes the data and subtracts the re-generated signal from the received waveform to cancel it so that the SINRs of the remaining signals are increased. If the receiver is then able to decode the next strongest signal, it applies SIC further, ideally until all mutually interfering signals are decoded [233]. SIC and similar interference cancellation techniques are used for multi-user detection (MUD) in cellular networks based on code-division multiple access (CDMA), which employ spread spectrum technologies such as DSSS [11, 93, 176].

Iterative interference cancellation is an enhanced MUD technique for CDMA that feeds the calculated bit probabilities back to the inputs of a soft decoder [11, 192, 235]. This thesis also introduces a technique called iterative interference cancellation (IIC) in Chapter 6 as a component of TVZF equalization, which also improves its outputs through multiple iterations, but it differs from these aforementioned works. Essentially, IIC performs a spectral estimation of OFDM pilot subcarriers rather than decoding raw baseband signals that encode payload data with a spreading code. In doing so, IIC refines the estimates of carrier frequency offsets (CFOs) and channel coefficients of multiple concurrent transmitters. Further, IIC can also be seen as a combination of SIC [233] and parallel interference cancellation (PIC) [177, 231], due to its successive addition of estimates while cancelling the known signal components in parallel.

Besides cellular networks, interference cancellation can also enhance the reception in WMNs based on IEEE 802.15.4 [88] and in IEEE 802.11 networks [206]. However, the OFDM-based IEEE 802.11 Non-HT PHY poses practical challenges with different signal strengths and data rates [206]. Further, ZigZag decoding applies SIC piecewise on sections of overlapping OFDM frames to decode multiple colliding transmissions [85].

This principle can be further improved through iterative soft-decision decoding [222] and works for up to four concurrent transmitters with the IEEE 802.15.4 PHY [118]. However, in contrast to TVZF equalization presented in Chapter 6, ZigZag decoding and its variants require that the colliding transmissions start with certain time offsets (TOs) to separate them [85]. Further, Strider [86] demonstrates that the resilience to collisions can be significantly enhanced through rateless codes. Strider also draws on SIC and is shown to work with a custom OFDM PHY inspired by IEEE 802.11 in testbed experiments with Universal Software Radio Peripheral (USRP) software-defined radios (SDRs) [229]. In contrast, the mechanisms of this thesis comply with IEEE 802.11 PHY formats.

2.1.4 *Full-Duplex Radios*

A common assumption in wireless communications is that a radio can either transmit or receive but not do both simultaneously, as its own transmission would be much stronger than any signals from remote transmitters that have undergone path loss and attenuation. However, there are various full-duplex prototypes in the research community that achieve self-interference cancellation through a combination of analog radio frequency (RF) and digital baseband processing [22–24, 54, 64, 108, 199], which theoretically doubles the link capacity. While most full-duplex radios have separate transmit (Tx) and receive (Rx) chains to reduce self-interference through careful antenna placement [54, 64, 108, 199], it is even possible to concurrently transmit and receive with the same antenna [22–24]. Full-duplex connections can be established by specialized MAC mechanisms in real-time to integrate the technology with IEEE 802.11 networks [108, 199, 221].

Since a full-duplex radio starts relaying almost immediately after the beginning of a reception, a message can even be relayed over multiple hops with low delay in a so-called cut-through connection [50–52, 253]. However, cut-through links are only end-to-end and there is just one relay at each hop, respectively. To utilize full-duplex technology also for CCT-based flooding with multiple transmitters at each hop, the transmissions would have to interfere in a way that the next-hop neighbors can decode the signals, e.g., by means of synchronization. FastForward [22] is a forerunner in this direction, as it achieves low-latency self-interference cancellation within the OFDM guard interval (GI) of IEEE 802.11, so that its transmitted signal combines beneficially with the original transmission at the receiver. Delay-and-forward [96] is another work that demodulates and re-modulates received symbols to remove noise, but it also delays symbols before their retransmission, so that they are aligned with the second next received symbol, respectively. While this concept requires a specialized receiver [96], it possibly also works for network flooding due to its accurate timing. Still, full-duplex technology also poses challenges in practical networks due to an increased level of interference [10, 170, 221].

2.1.5 *Useful Interference with IEEE 802.15.4*

Besides drawing on the capture effect and interference cancellation, distributed network nodes can also align concurrent transmissions of identical signals through precise time synchronization. With this, the received signal ideally resembles a transmission from a single device with a high signal-to-noise ratio (SNR).

Backcast [66] demonstrates that up to a dozen recipients can synchronously send an acknowledgment (ACK) frame in response to a broadcast or an anycast message with IEEE 802.15.4 [107], which is based on DSSS [93]. The superposition of these ACKs can be decoded correctly, so that it can be interpreted as a logical *OR* of ACKs [66]. With more than six concurrent transmitters, the frame reception rate (FRR) stays at a high level only if the ACKs are generated with precise timing in hardware, which improves the reception beyond the benefit of the capture effect [66].

Glossy [73] further expands on the idea of generating decodable interference with the DSSS-based IEEE 802.15.4 PHY [107] to facilitate multi-hop data dissemination and implicit time synchronization in WSNs. Analyses show that useful interference can be achieved in IEEE 802.15.4 networks if the TO between concurrent transmissions stays below 0.5 µs [73, 237]. Glossy employs sophisticated techniques to meet this timing requirement accurately, which are necessary due to several limitations introduced by the two-part hardware platform on which it is based [73, Section 5]. First, a retransmission is triggered in software by a microcontroller unit (MCU) after having received an interrupt from the radio. Second, the MCU and the radio are clocked by different oscillators, so that events in the two domains may happen at the fraction of a clock cycle of the other domain, respectively. Third, the MCU's clock can deviate by up to 20 % from its nominal frequency, which can affect the software execution time. Despite these impediments, 91 % of the retransmissions meet the targeted software delay, within the quantization step size of 125 ns according to the radio's sampling rate [73, Section 5.3].

However, Glossy has to deal with effects that hamper the alignment of superimposing signals. Most notably, small TOs introduced by the software implementation of Glossy may accumulate over multiple hops and ultimately impede the adherence to the timing requirement [73, 237, 257, 258]. Further, the reliability in terms of the FRR depends on the frame length, even in the case of perfect time synchronization with two senders: while the FRR is 95 % for short frames with 8 bytes, it is about 73 % for long frames with 128 bytes [73, Section 7.1.1, Figure 11]. The authors of Glossy argue that the FRR decreases with an increasing frame length also for non-concurrent transmissions [73], but another analysis reveals that useful interference cannot be guaranteed for long frames if the radios of the transmitters have large clock frequency offsets with respect to each other [257, 258]. The reason is that a CFO introduces a phase drift in the receiver's baseband signal during downconversion, i.e., a continuously changing phase shift [215]. Similarly, a sampling frequency offset (SFO) compresses or stretches a baseband signal temporally, which causes a continuously changing TO and thus also a phase drift [215]. Hence, the symbols may not add up constructively throughout an entire frame even if they do so at the beginning [257, 258]. In fact, if two devices send with the same power and with slightly different carrier frequencies, a beating effect occurs, i.e., alternating periods of constructive and destructive interference, which can be addressed, for instance, by means of forward error correction (FEC) with Reed-Solomon codes [174].

CCT with Glossy achieves a power gain only if the difference of the received signal strength indicators (RSSIs) of individual signal components does not exceed 3 dB, i.e., the signals must ideally arrive with the same power to increase the received signal power further [236]. Also, the FRR decreases rapidly when the different signal components have different propagation delays [236]. TriggerCast [236] addresses these issues and enhances Glossy with the goal to achieve useful interference more often than Glossy in order

to gain a higher FRR. It introduces a calibration method called *chip level synchronization* to compensate for propagation delays and clock frequency offsets, which, however, also creates overhead and has to be executed periodically in changing environments. Additionally, TriggerCast employs a method called *link selection and alignment* to make only the nodes participate in message forwarding that are likely to increase the SINR of the superimposed signal. In testbed experiments, TriggerCast improves the RSSI and the FRR with an increasing number of concurrent transmitters [236]. Particularly in dense networks, the reception performance can be improved by letting only selected nodes retransmit messages, while the other nodes only overhear the channel, as shown in simulations [237] and in testbed experiments [236].

While Glossy and some related works in the WSN literature [73, 236, 237, 257, 258] refer to the resulting effect of a CCT with accurate time synchronization of below 0.5 µs as *constructive interference*, this chapter uses the term *useful interference* for this effect since this thesis uses the term *constructive interference* more accurately to describe the alignment of the instantaneous phase values of concurrent transmitters. In fact, the time synchronization of concurrent transmitters only avoids intersymbol interference (ISI) at the beginning of a frame, but it does not ensure that the respective signals' phases are aligned at the receiver. Further, a phase drift introduced through a CFO or a SFO between concurrent transmitters makes their respective phase relations change over time, often leading to destructive interference at a later position of the data frame [174, 257, 258]. Hence, the radio clocks of multiple Tx or Rx chains must be synchronized in frequency to achieve *coherence*, i.e., to preserve their phase relations [4, 15, 153, 154, 162, 164]. While TriggerCast's *chip level synchronization* compensates various time delays, it possibly accomplishes also coherent interference through its oscillator calibration procedure, even though its authors do not explicitly claim this achievement [236].

The generation of coherent interference is essential for multi-antenna communication systems and is discussed in more detail in Section 2.1.6. Works that build on Glossy to provide extended functionalities for WSNs are discussed in Section 2.3.

2.1.6 *Coherent Interference*

Generating coherent signals through multiple RF chains is a fundamental precondition in multi-antenna communication systems, such as multiple input multiple output (MIMO) and beamforming [119]. These systems increase the network capacity by transferring more data in parallel and by steering radio waves in certain directions. In the basic scenario, multiple antennas are attached to a common device, so that the RF chains can be driven by a shared clock. This section outlines techniques that rely on coherent interference, which can be extended to operate also on distributed devices with uncoupled oscillators. Synchronization methods that facilitate the generation of coherent signals also with distributed devices are further discussed in Section 2.2.

2.1.6.1 *MIMO and MU-MIMO*

MIMO is a multi-antenna technique that draws on spatial diversity, spatial multiplexing or a combination of both [32, 119]. While the spatial diversity gain depends on the number of antennas per device for both the Tx and the Rx, respectively, the spatial multiplexing gain is a function of the rank of the sum correlation matrix of the scatterer clusters of

the wireless channel [32], but it can only be fully leveraged if a corresponding number of antennas is available for both the Tx and the Rx [15]. However, there is an imbalance between the numbers of antennas that can be attached to a base station (BS) or an access point (AP) and mobile devices, as the latter have only limited space to accommodate antennas. Recent mobile phones support up to 4x4 MIMO [49], whereas BSs have tens of antennas and are expected to be scaled to hundreds of antennas with massive MIMO in fifth generation (5G) cellular networks [30, 49]. Hence, state-of-the-art technologies such as fourth generation (4G) Long-Term Evolution (LTE), 5G New Radio (NR) [157] and IEEE 802.11ac/ax [104, 105] employ multi-user MIMO (MU-MIMO), where multiple mobile devices form virtual antenna arrays to be served simultaneously [39]. With this, a BS or an AP can send individual data through so-called spatial streams via multiple antennas to multiple mobile devices in the downlink [39, 219], while its RF chains operate coherently by means of shared clock signals [211]. In addition to forming virtual antenna arrays with multiple mobile devices, also multiple BSs can be combined to form even larger virtual antenna arrays in cooperative multi-point (CoMP) systems to tackle the problem of inter-cell interference [147, 148], which requires synchronization also between the BSs [210]. LTE CoMP comprises, for instance, coordinated scheduling, coordinated beamforming and joint processing between multiple BSs [147].

While MU-MIMO has also been studied extensively for the uplink direction, i.e., with multiple mobile devices transmitting concurrently to a BS [31, 39], such works assume that the distributed transmitters interfere coherently, which, however, requires additional frequency synchronization in practice due to frequency offsets of the devices' local oscillators [29]. In fact, accurate frequency synchronization of distributed devices is considered to be technically challenging due to its overhead, which has impeded the inclusion of MU-MIMO uplink, for instance, in the CSMA/CA-based IEEE 802.11ac standard [104, 264]. However, MU-MIMO uplink is part of the more recent IEEE 802.11ax standard, in which the AP is in charge of resource management tasks, such as spectrum allocation and transmission scheduling [105, 264].

MU-MIMO uplink is supported in 3rd Generation Partnership Project (3GPP) 5G cellular networks since release 15 of the specifications [112, Chapter 1.20], which is enabled through phase tracking reference signals in the downlink [112, Chapter 3.7.6] and in the uplink [112, Chapter 7.5.4] directions. In 5G cellular networks, MIMO and Tx diversity transmissions must meet a TO within 65 ns [1, Section 6.5.3], whereas BSs with overlapping coverage must meet a TO within 3 µs [1, Section 6.5.3] [112, Chapter 13.2]. In addition, a BS must be frequency-synchronized with an accuracy better than 0.05 parts per million (ppm) [1, Section 6.5.1] [112, Chapter 3.7.6], whereas a mobile device must be frequency-synchronized with an accuracy better than 0.1 ppm with respect to the carrier frequency of the BS [2, Section 6.4.1] [112, Chapter 3.7.6].

2.1.6.2 *Beamforming and Beamnulling*

Beamforming, also referred to as *spatial filtering*, is a multi-antenna technique that steers transmitted signals into desired directions and that increases a receiver's sensitivity in certain spatial directions [19]. A beamforming device controls phase shifters in its RF chains such that the different signal copies interfere constructively in one or more spatial directions, which works for both Tx and Rx. In non-line-of-sight (NLOS) environments with multi-path effects, a beamformer needs channel state information (CSI) feedback

to steer transmissions and receptions towards the respective devices by equalizing the frequency-selective fading on its different RF chains [100]. Beamforming is combined with MU-MIMO in 5G NR so that multiple users can be served simultaneously via the same resource blocks if they are spatially separated [112, Section 1.21]. Further, hybrid beamforming is a promising technique for massive MIMO, as it allows for a significant training overhead reduction [156].

In contrast to concentrating transmit energy towards specific users through constructive interference, beamnulling is a complementary technique that makes signals interfere destructively at specific users. With this, the network capacity can be further increased through parallel data transfers that interfere with other receivers as little as possible. Concurrent CSMA [198] allows secondary users to access the wireless channel in an IEEE 802.11 network during an ongoing data transfer of a primary user. The secondary users minimize their interference in the direction of the primary receiver, which yields a network throughput increase by a factor of two to three [198]. Further, a multi-antenna receiver can also apply beamnulling to reduce its exposition to interference [70] or jamming signals [25]. Beamnulling can also be referred to as nullforming [33, 34, 120].

Beamforming also works with distributed devices, which requires time and carrier frequency synchronization, carrier phase alignment and CSI feedback [33, 34, 110, 163].

2.1.7 *Cooperative Relaying with OFDM*

Message relaying allows extending the communication range of a wireless transmitter, which is particularly effective with cooperative transmissions, as a receiver can combine multiple copies of a message to harness SNR gains [133]. CCT is a special form of a cooperative transmission in which multiple nodes transmit a message concurrently, allowing them to use spectral resources sparingly by using the same frequency channel. However, synchronization in time and frequency is required to obtain a signal that resembles a transmission from a single transmitter with multi-path effects [114, 133].

CCT allows for both Rx diversity and Tx diversity, as a node can opportunistically become a relay upon the reception of a message and as multiple relays can form a virtual antenna array to retransmit a received message in a multiple input single output (MISO) fashion [18, 92, 109, 121, 123]. With this, CCT can be employed in WMNs for both directed message delivery and network flooding to improve throughput, latency and reliability [133]. Virtual antenna arrays that form through the successful reception of a message to disseminate the message further via CCT in a WMN are also referred to as opportunistic large arrays (OLAs) [202]. OLAs are studied under a variety of aspects, such as range extension [202], energy savings [94], scalability [212] and throughput [115].

This section presents works that address CCT-based relaying with an OFDM PHY since the mechanisms and prototypes presented in this thesis work with signals compliant with OFDM-based IEEE 802.11 PHYs. Section 2.1.7.1 presents practical systems for CCT-based relaying with OFDM, enabling communication range extension and opportunistic routing. Section 2.1.7.2 presents works for CCT-based relaying over multiple hops, which deal with the synchronization of successive virtual antenna arrays. Section 2.1.7.3 presents related work on CCT-based network flooding with OFDM.

2.1.7.1 *Practical Two-Hop Relaying with OFDM*

SourceSync [186] is a pioneering work in the field of relaying via CCT that provides practical solutions to facilitate application scenarios that are also addressed by this thesis and that is also based on an OFDM-based IEEE 802.11 PHY [104]. SourceSync allows for opportunistic routing by combining receiver diversity and sender diversity, i.e., it draws on multiple nodes being able to receive a transmission from the source and on the destination being able to receive a CCT from multiple relays [186]. In addition, SourceSync facilitates last-hop diversity, i.e., multiple APs send data to a client via CCT, which resembles the application scenario of extensive broadcasting in Section 1.2.2. SourceSync is evaluated by means of a prototyping platform with a field-programmable gate array (FPGA) and achieves a throughput gain of 57 % in last-hop diversity with two APs and 70-100 % in opportunistic routing with three relays [186].

However, SourceSync [186] deviates from the IEEE 802.11 standard [104] as follows. To synchronize in time, it compensates for different response times and propagation delays between the transmitters and the receiver through a periodic exchange of probes and by signaling measured misalignments through the ACKs sent by the receiver [186]. These probes are also used to estimate the CFOs of the transmitters with respect to the receiver so that the devices can synchronize in frequency through baseband processing. Further, SourceSync does not comply with an IEEE 802.11 PHY, so its transmissions cannot be decoded by commodity Wi-Fi hardware. First, SourceSync employs a lead sender to manage channel access and to trigger co-senders through a sync header. Second, SourceSync employs orthogonal training symbols through interleaving, i.e., the training symbols for channel estimation from the transmitters are separated from each other in the preamble to avoid interference from the respective other senders. Third, the pilot subcarriers are also allocated in an interleaved fashion across OFDM symbols to the senders so that their phases can be tracked by the receiver without interference from the other senders, respectively [186]. This indicates also that the frequency synchronization is rather inaccurate since residual CFOs occur and must be compensated. Fourth, SourceSync leverages a space-time block code (STBC) to mitigate the impact of deep fading subcarriers due to destructive interference [186].

As opposed to SourceSync [186], the mechanisms presented in this thesis enable the same scenarios with standard-compliant IEEE 802.11 PHY formats [104] and do not require explicit signaling for coordination or synchronization, nor any additional FEC. In particular, the prototype presented in Chapter 4 achieves synchronization in time and frequency by means of a single frame reception, which may even arrive as a CCT from multiple previous-hop neighbors. Further, the TVZF equalization presented in Chapter 6 allows receivers to decode CCTs from nodes with poor frequency synchronization. This is useful to enable the extensive broadcasting scenario presented in Section 1.2.2, but TVZF can also facilitate the opportunistic routing scenario of SourceSync [186] and low-latency network flooding. Moreover, TVZF does *not* require an interleaved allocation of training symbols or pilot subcarriers and works with interfering signals instead.

The communication range of mobile devices can be extended via CCT-based relaying also in cellular networks, as shown for a MU-MIMO uplink scenario where multiple source nodes concurrently send to a BS via a distributed amplify-and-forward relay [135]. The BS triggers a CCT from the source nodes by means of a sync header, which the source nodes utilize for time and frequency synchronization through baseband processing [135].

The system [135] employs a custom OFDM PHY and is evaluated in testbed experiments with USRP SDRs [229] and GNU Radio [84]. Similarly, distributed amplify-and-forward relaying via CCT also works in IEEE 802.11 networks, where a single source node can trigger the procedure itself [16]. Amplify-and-forward relaying, however, amplifies noise besides the useful signal, which makes it unsuitable for multi-hop networks. Hence, the methods presented in this thesis employ decode-and-forward relaying, i.e., if the received data have a correct frame check sequence (FCS), they are re-generated for retransmission to avoid an accumulation of noise.

Besides multiple relays, a single relay may also transmit a message concurrently with a source node to a destination node [167, 168]. To this end, the source node has to transmit a message twice, once to the relay and once with the relay to the destination. To synchronize the transmissions of the source node and of the relay in frequency, a preamble with periodic symbol copies [203] can be employed in conjunction with pilot tones, providing sufficient accuracy in higher SNR regimes [167, 168]. To synchronize both transmissions in time, hardware timers can be employed to trigger both nodes relative to a common event like a frame reception [167, 168]. The prototype presented in Chapter 4 draws on such techniques for time and frequency synchronization, too, and is developed on the successor Wireless Open Access Research Platform (WARP) v3 [246]. However, the solutions and prototypes of this thesis are designed for multiple CCT relays and for standard-compliant OFDM-based IEEE 802.11 PHY formats [104].

2.1.7.2 *Practical Multi-Hop Relaying with OFDM*

Another line of work [44, 45, 48, 79] investigates time and frequency synchronization techniques to efficiently relay data over multiple hops in a cascaded distributed MIMO (DMIMO) scenario. In this scheme, the nodes form a corridor-like topology with a fixed number of relays at each hop, so that each of these clusters of nodes relays the received data to the respective next cluster via CCT. The relays within each cluster leverage data frame receptions as a reference for synchronization in time and frequency [44, 45, 79]. To this end, each receiver estimates the synchronization parameters TO [44, 45, 79] and CFO [44, 79] from each transmitter of the previous hop, respectively. This procedure is facilitated through orthogonal preambles, e.g., the transmitters send the long training sequence (LTS) from IEEE 802.11a [79, 104] on different frequencies, which requires an assignment of the respective senders in each cluster to a frequency channel, so that a receiver can separate their respective training symbols [44, 45, 79]. Each relay weights and combines these estimates to obtain a single estimate for both the TO and the CFO, respectively, which are used for synchronization through baseband processing. Then, the synchronized nodes of a cluster relay the data via CCT to the next hop without any further interaction [44, 45, 79]. The evolution of TOs and CFOs over multiple hops is analyzed through simulations and ping-pong testbed experiments with USRP SDRs [229] and GNU Radio [84]. The root mean square (RMS) transmit time spread consistently stays below 200 ns in 90 % of the cases [44, 45, 79]. In addition to the narrow-band binary FSK (BFSK) PHY, which allows for non-coherent demodulation [44, 45, 79] and which operates at only 64 kbps at a sampling rate of 1 megasamples per second (MS/s) [45], a follow-up work proposes a sophisticated preamble structure to allow for a weighted combination of the synchronization parameters also with an OFDM PHY [44, 48].

While the aforementioned line of work [44, 45, 48, 79] has in common with this thesis to propagate synchronization parameters over multiple hops, there are also important differences. As opposed to a pre-defined corridor structure with a fixed number of nodes per hop, this thesis addresses extensive broadcasting and network flooding with a flexible number of nodes joining CCTs opportunistically. Further, the solutions of this thesis do not draw on an orthogonal preamble allocation and thus also do not require any assignments of nodes to time or frequency slots. Instead, the synchronization parameters TO and CFO are estimated by means of superimposed CCT signals to maintain compatibility with the IEEE 802.11 [104] PHY formats. In contrast to prototyping with GNU Radio [84], which incurs waiting between transmissions for data processing [44, 45, 79], this thesis comprises prototypes capable of real-time processing in the FPGA of WARP v3 SDRs [246] in Chapter 4 and through firmware modifications of commodity hardware in Chapter 5. Moreover, as opposed to only ten rounds of ping-pong transmissions for analyzing the temporal behaviour of the synchronization parameters [44, 45, 79], this thesis comprises experiments with runs of 10 000 successive CCTs with IEEE 802.11 data frames to study the progression of performance metrics over time and to demonstrate their stability.

2.1.7.3 *CCT-Based Network Flooding with OFDM*

CCT-based network flooding, such as with Glossy [73], is a special case of cooperative relaying where multiple nodes relay a message to several other nodes in multiple turns to cover an entire area. An analysis of network flooding with CCT using OFDM shows that the median SNR is improved by 3 dB, 7 dB and 9 dB for 2, 5 and 9 cooperative relays in an urban environment, respectively [68]. However, besides SNR improvements, there is also a risk for SNR deterioration with CCT due to destructive interference. This risk can be reduced with an increasing number of cooperative relays and it can be further reduced with phase dithering and delay diversity [68]. While measured channels from an urban environment are employed for the analysis [68], the work is still based on a simplified system model that does not address frequency synchronization, i.e., it assumes that all signal components of a CCT are coherent. Further, it assumes that the OFDM GI can accommodate the TOs and the channel delay spread (CDS) of a CCT [68], which may not hold in practical systems if the time synchronization fails or if the TOs accumulate over multiple hops. In contrast, this thesis analyzes the impact of these factors and also takes them into account in practical testbed experiments.

2.1.8 *Single Frequency Networks*

The extensive broadcasting scenario introduced in Section 1.2.2 is related to methods and technologies for wireless media distribution. In radio and television broadcasting, multiple stations may send the same signal on the same frequency channel to combine their coverage. The stations in such a single frequency network (SFN) are synchronized in time and frequency by means of cables, but the different signal copies can still arrive with different propagation delays at a receiver. Therefore, the OFDM-based Digital Video Broadcasting – Second Generation Terrestrial (DVB-T2) standard, for instance, allows for a GI ranging from 7 µs up to 2636 µs [69, Tables 65 and 67]. With this, different copies of the signal can be ensured to overlap within the length of the GI in different installation settings, so that intersymbol interference (ISI) is avoided at the receivers.

A CCT from an SFN effectively resembles a transmission from a single station. Similarly, 3GPP cellular networks allow for Multimedia Broadcast and Multicast Service (MBMS) via Multicast/Broadcast over SFN (MBSFN) since 4G LTE [136, 137]. With these means, neighboring synchronized cells can disseminate the same data to multiple users.

Broadcasting with multiple antennas can also be adopted at smaller scales with a distributed antenna system (DAS). A DAS consists of a single device with multiple synchronized RF chains to connect antennas via long cables in different locations [172]. With this, it can coherently broadcast the same data in an extended area, but it can also carry out unicast and MU-MIMO transmissions [113].

Co-BCast [184] is a broadcasting system for video streaming and multicast services at big events that employs multiple unsynchronized APs. Co-BCast effectively tackles the interference caused by TOs and CFOs in CCTs through FEC with a low-density parity check (LDPC) code, as shown in testbed experiments with WARP v3 SDRs [246]. In simulations of a stadium scenario, Co-BCast achieves a broadcasting throughput of $\sim 100\,\mathrm{Mbit/s}$ using four distinct 20 MHz channels. As opposed to employing a more robust FEC, this thesis addresses the interference caused by TOs and CFOs through TVZF equalization, as presented in Chapter 6.

2.2 Synchronization

Electronic devices are typically equipped with local oscillators to generate clock signals. Due to manufacturing tolerances and physical effects, however, an oscillator's frequency can deviate from its nominal frequency and fluctuate. Synchronization enables uncoupled nodes to establish a common notion of time and to compensate for frequency offsets, e.g., to comply with time schedules and frequency allocations. This section presents synchronization principles and techniques to facilitate coherent CCT.

2.2.1 *Fundamentals*

Synchronization has the goal to align chronometric properties of temporal processes. Given that temporal processes in technical systems are typically monitored and driven by means of a clock, these chronometric properties comprise the *frequency* and the *phase* of a clock as well as an absolute *time reference*. The frequency of a clock is defined as the number of clock cycles per second, which, for instance, may be the cycles of a signal generated by a local oscillator. In addition, the phase of a clock describes the fraction of a cycle at a particular point in time and is often given as an angular entity.

Two clocks that are synchronized in frequency count the exact same number of cycles between the occurrences of two events, in regard to their phases at the time of the first event, respectively [147, Chapter 8.1.1]. With this, the angular difference of the phases of the two clocks is fixed, even though their phases do not necessarily have to be aligned. Clocks that are synchronized in frequency are said to be *coherent* and are sometimes also referred to as *syntonized* [37, 139, 147].

When two clocks are synchronized in time, their phases are set to the same value and they start counting cycles at the same point in time. A clock may additionally be given an absolute time reference, such as a timestamp, so that it starts counting cycles with respect to this reference. With this, the two clocks display the same time in terms of elapsed

clock cycles and phase at the point in time when the time synchronization takes place. If the two clocks are also synchronized in frequency, their phases keep their alignment over time, so that they complete each of the following clock cycles at the same time. If the two clocks are not synchronized in frequency, their phases drift relative to each other, i.e., one clock runs faster than the other. To keep the time offset between unsyntonized clocks small, they must be synchronized in time periodically [147, Chapter 8.1.1].

2.2.2 *Practical Considerations*

Electronic devices are typically equipped with one or several *local oscillators* that generate clock signals to drive hardware components. In particular, a wireless radio employs a sampling clock to process baseband signals with an analog-to-digital converter (ADC) and a digital-to-analog converter (DAC) as well as an RF clock to upconvert baseband signals to the carrier frequency and vice versa. Different clock signals may run at different frequencies, but they can be sourced from the same local oscillator through a digital clock scaler or an analog phase-locked loop (PLL), which maintain a fixed relationship between the phase of their output and the phase of their reference signal, respectively [95].

The quality of an oscillator is typically described by its accuracy in terms of parts per million (ppm) of its nominal frequency [147, Chapter 8.1.1]. Cheap oscillators can have frequency offsets of $\sim \pm 100$ ppm with respect to their nominal frequency [147, 163, 258], whereas temperature-compensated crystal oscillators (TCXOs) can have accuracies better than 1 ppm [53, 147]. Similarly, a clock can also be said to be synchronized in frequency to a certain level of uncertainty if its frequency offset with respect to the nominal frequency or a reference does not exceed a specified range [147, Chapter 8.1.1].

Phase noise is a dynamic phase variation in clock generation circuits [58] that can be separated into short-term variations, i.e., phase jitter and time jitter, and long-term variations, i.e., phase drift, time drift and frequency variations [95, 153]. These effects are induced by voltage variations, temperature changes and thermal noise [58, 95, 153]. Hence, systems that rely on coherence have to perform frequency synchronization periodically or continuously, depending on the clock frequency variations, the jitter, the synchronization method and the required accuracy [153].

The IEEE 802.11 specifications require that the carrier frequency and the symbol clock frequency are derived from the same oscillator [104, Section 17.3.9.5]. Furthermore, the IEEE 802.11 specifications require a node operating in a 20 MHz channel to meet the center frequency with an accuracy of ± 20 ppm [104, Section 17.3.9.5]. With this, the CFO between two devices can be as high as 40 ppm, i.e., ~ 100 kHz in the 2.4 GHz band and ~ 236 kHz in the 5 GHz band. The IEEE 802.11 standard also specifies a procedure for clock drift adjustments that is used for the time-synchronized transmission of beacon frames in mesh mode [104, Section 14.13.2.2.3]. Furthermore, a receiver synchronizes temporarily in time and frequency to a transmitter by means of the IEEE 802.11 frame preamble [203, 215] to avoid ISI and intercarrier interference (ICI) [13, 183]. As this procedure is employed and extended in this work, it is outlined in Section 2.4.

2.2.3 *Shared Clock*

The generation and scaling of clock signals on a single device is typically accomplished by clock distribution integrated circuits (ICs), which may share a common local oscillator as a reference. The WARP v3 SDR [246], for instance, which is used for prototyping and testbed experiments in this work, employs two local oscillators and a clock module that facilitates flexible distribution of clock signals also across different devices [248]. The first oscillator is connected via low-voltage differential signaling (LVDS) to the Xilinx Virtex 6 FPGA and is used to generate various clock signals for its digital circuitries. Furthermore, a TCXO is connected to two AD9512 clock distribution ICs, which generate a sampling clock and an RF clock for the radios, respectively. There are two RF chains integrated on each WARP v3 board and two additional RF chains can be attached through an FPGA mezzanine card (FMC) module, so that up to four RF chains can be operated coherently by one WARP v3 SDR [248]. There are two different clock modules for WARP v3, one of which can share the sampling clock and the RF clock separately across different WARP v3 boards via micro-miniature coaxial (MMCX) wires [249], so that multiple WARP v3 SDRs can generate coherent radio waves. While commodity devices follow similar principles for clock generation and scaling, they do not offer much flexibility in terms of clock sourcing and distribution.

Argos [211] is a base station architecture based on WARP v3 [246] that addresses various practical challenges to demonstrate that the number of antennas can be flexibly scaled in practical MIMO systems. Argos defines a hierarchy for the WARP SDRs to distribute data and computational load and to apply beamforming weights at the antennas [211]. To align concurrent transmissions and receptions in time, a dedicated SDR sends a sync pulse via general-purpose input/output (GPIO) pins to the other SDRs [211]. This wired time synchronization functionality is readily provided by WARP v3 [246]. The Argos prototype is made up of 16 WARP SDRs and has 64 antennas [211]. Besides Argos, BigStation [252] is a similar BS architecture for MU-MIMO applications with 12 antennas attached to 3 Sora MIMO kits that are synchronized by an external clock source [252]. As opposed to the MIMO and beamforming scenarios enabled by Argos and BigStation, the CCT-based methods presented in this work are more lightweight, i.e., they work with one antenna per device and perform time and frequency synchronization without complicated signaling.

2.2.4 *Cooperative Wireless Synchronization*

In addition to distributing sync pulses and clock signals via cables, synchronization can also be achieved by recovering time and frequency references from wireless signals. Similar to wired clock distribution, wireless synchronization for distributed devices is often designed in a hierarchical topology. The main node, which may be the receiver in a multi-user transmission [17, 35, 36, 162, 227], broadcasts a clock reference derived from its local oscillator, while the secondary nodes derive their local clocks either directly from this reference [17, 35, 227] or they periodically synchronize a local clock [36, 162].

When the main node continuously broadcasts a sinusoidal beacon, the secondary nodes can generate their local clocks by means of a PLL from the received signal [17, 35, 227]. Further, the main node coordinates concurrent transmissions from the secondary nodes

by sending triggers [17] or it may encode a time reference in its beacon signal, which is used for time synchronization by the secondary nodes in addition to the beacon's phase [227]. To solve the problem that secondary nodes may have different distances from the main node in a distributed beamforming scenario [17], the propagation delays between the main node and the secondary nodes can be pre-compensated by tracking the beacon signal in a closed feedback loop [163], i.e., the secondary nodes feed back the received beacon signal on a different frequency using a multiple-access scheme [227]. Alternatively, the beacon signal from the main node and the feedback from the secondary nodes can also be exchanged in a time-division duplex (TDD) scheme [162].

In the previously described systems, node mobility can lead to CFOs due to the Doppler effect and distort the carrier coherence. This problem can be overcome by a two-source carrier synchronization scheme, where a secondary sinusoidal beacon is generated by the secondary nodes on a separate frequency channel, which in turn is used by these nodes themselves to generate the carrier frequency for the actual data transmissions [35]. Alternatively, the main and the secondary nodes can successively send short reference beacons in a TDD fashion, allowing each node to derive combined frequency and phase estimates by means of the beacons from all other nodes, respectively [36]. As this method is effective against the impact of mobility, the time span during which the synchronized nodes operate coherently mainly depends on the jitter of their local oscillators [36, 153]. Since the feedback from the secondary nodes is not consumed by the main node, these schemes can be categorized as an open-loop synchronization [110, 163].

While the synchronization systems described in this section are considered in isolation, a more realistic scenario like a WMN may also introduce co-channel interference due to communications by other nodes. This problem can be solved, for instance, by performing synchronization and channel estimation concurrently by multiple pairs of transmitters and receivers using Zadoff-Chu sequences [111].

2.2.5 *Practical Wireless Synchronization Systems*

Besides theoretical concepts and simulations, some works demonstrate the feasibility of wireless synchronization for multi-user transmissions also practically. This section outlines such works and compares their synchronization accuracies with each other as well as with the Global Positioning System (GPS) and synchronous Ethernet (SyncE).

2.2.5.1 *Characterization of the Synchronization Systems*

JMB [187] enables *joint multi-user beamforming* with a main AP and several secondary APs that synchronize in phase and frequency prior to CCTs. JMB employs Ethernet to distribute data to all APs and to let the main AP coordinate CCTs with the secondary APs, whereas synchronization is performed wirelessly through a synchronization header that is followed by pilot symbols for CFO and CSI estimation transmitted by both the main and the secondary APs in an interleaved fashion. JMB accomplishes frequency synchronization of all APs by compensating the CFOs of the secondary APs through baseband processing. The CFO estimates are obtained by averaging over samples taken across multiple frame transmissions, which increases the accuracy as the CFOs change only slowly over time [187]. While these CFO estimates are sufficiently accurate to track the relative phase drifts between the secondary APs and the main AP over the duration

of a frame, respectively, they would still be too inaccurate to track phase drifts on a time scale of multiple frames [187]. Thus, each secondary AP estimates its phase offset (PO) relative to the main AP by means of the synchronization header prior to each CCT and adjusts its phase accordingly [187]. However, this phase synchronization is *not* equivalent to a time synchronization, as it does not compensate for propagation delay differences between the APs, in contrast to some works discussed in Section 2.2.4 [35, 36, 163, 227]. Still, this method suffices as long as a CCT is covered within the GI of the OFDM PHY. In fact, a TO of a sender translates to a frequency-dependent phase shift of its CSI. Further, CSI can be refreshed according to the coherence time of the wireless channel, which is at the order of hundreds of milliseconds in indoor environments [89, 187]. The throughput of JMB scales linearly with the number of APs in testbed experiments with USRP SDRs [229] and GNU Radio [84] with up to ten APs and ten clients [187]. JMB's median and 95th percentile phase misalignments of frames with 1500 bytes payload and with a channel bandwidth of 10 MHz are 0.017 rad and 0.05 rad, respectively [187].

MegaMIMO 2.0 [89] is a real-time distributed MIMO system for IEEE 802.11 that builds on top of JMB [187]. Besides synchronization, it addresses other problems arising in practical distributed MIMO, such as CSI updates, power control and the architecture.

AirSync [15] employs a main AP sending an out-of-band reference signal with several OFDM pilot tones to synchronize co-located secondary APs for distributed MU-MIMO transmissions. The secondary APs track their POs with respect to the main AP and simultaneously shift the phases of their transmitted complex-valued baseband signals accordingly to maintain phase coherence in CCTs [15]. AirSync is implemented in the FPGA of WARP SDRs [246] and works in a testbed with four devices acting as APs and four devices acting as clients [15]. AirSync achieves time synchronization within the OFDM GI and a carrier phase coherence within ± 2.37 degrees and ± 4.5 degrees in terms of standard deviation and 95th percentile, respectively [15, Section 6].

AirShare [4] broadcasts a reference beacon consisting of two tones generated by PLLs from the same local oscillator to enable distributed nodes to derive coherent clocks from this signal. Thus, AirShare is similar to some works discussed in Section 2.2.4 [17, 227]. However, AirShare addresses only phase coherence [4] but not phase alignment, i.e., it neglects the compensation of propagation delays that is employed for distributed beamforming by the works discussed in Section 2.2.4 [35, 36, 163, 227]. AirShare is implemented on USRP SDRs [229] and employs SourceSync [186] for coordination in testbed experiments [4]. AirShare achieves a median CFO of 0.4 Hz and a 95th percentile CFO of 1.24 Hz between pairs of nodes in the 2.4 GHz band. For an IEEE 802.11 Non-HT frame with 1500 bytes at 6 Mbit/s, this translates to an accumulated median phase error of 0.005 rad and a 95th percentile phase error of 0.016 rad [4, Section 4.1].

Stitch [139] is an architecture for embedded wireless network devices that provides flexibility in the clocking of subsystems and that allows for accurate time and frequency synchronization across devices. To this end, Stitch employs a single local oscillator and utilizes an FPGA to flexibly synthesize the clocks as required so that all subsystems are synchronized in frequency with fixed scaling factors.

PlaStitch is a practical realization of Stitch with a narrow-band radio for frequency synchronization and an ultra-wideband (UWB) radio for time-of-flight (ToF) estimation, both of which are driven by a shared 1 ppm voltage-controlled TCXO (VCTCXO) [139]. In contrast to works that realize frequency synchronization through digital baseband

System	CFO (Hz)	CFO (ppb)	PO (deg)	PO (rad)	Metric	Sync
GPS [130]	0.53/3.30	**0.22/1.37**	(0.38/2.41)	(0.007/0.042)	90%/max	ambient
SyncE [142]	2.41	**1**	(1.76)	(0.031)	bound	wired
JMB [187]	(1.33/3.92)	(0.55/1.63)	0.97/2.86	**0.017*/0.05***	med/95%	per frame
AirSync [15]	N/A	N/A	**2.37*/4.5***	0.041/0.079	std/95%	continuous
AirShare [4]	**0.4/1.24**	0.17/0.51	0.29/0.92	**0.005/0.016**	med/95%	continuous
PlaStitch [139]	8.5	**3.54**	(6.24)	(0.11)	RMSE	1/1200 Hz
PreSync [180]	0.13	**0.055***	(0.096)	(0.0017)	bound	20 Hz

Table 1: Accuracy of wireless synchronization systems: Carrier frequency offset (CFO) conversions between Hz and ppb are provided for Wi-Fi channel 1 in the 2.4 GHz band, i.e., 2.412 GHz. The accumulated phase offsets (POs) apply for an IEEE 802.11 Non-HT frame with a 1500-byte PSDU at 6 Mbit/s on a 20 MHz wide channel, i.e., after 2 024 µs. Original values from the references are bold, inferred values are enclosed in parentheses. Values marked with an asterisk are obtained differently but are valid for comparison without scaling.

processing [15, 89, 187], PlaStitch tunes its VCTCXO by means of two DAC channels at a combined resolution of 20 bits [139]. The 40 MHz VCTCXO has an RMS error (RMSE) of 3.54 parts per billion (ppb), i.e., 0.00354 ppm [139] in a lab experiment in which a PlaStitch node is resynchronized every 20 minutes over 24 hours.

Another line of work [178–180] presents distributed beamforming with USRP N2X0 SDRs [229] and GNU Radio [84] by continually pre-synchronizing a distributed antenna array consisting of three nodes through periodic in-band signaling. When the nodes of the distributed antenna array receive a message from a distant node, they leverage channel reciprocity to adjust their precoding coefficients such that beamforming is achieved in the direction of the distant node while sending a message back to it [179, 180]. The system is also demonstrated to work for nullforming [178]. Both the main and the secondary nodes employ extended Kalman filters (EKFs) and perform periodic signaling with an update interval of 50 ms [180]. With this, the frequency synchronization is better than 0.05 Hz at a center frequency of 915 MHz [180], which translates to 0.055 ppb.

2.2.5.2 *Comparison of the Frequency Synchronization Accuracy*

Table 1 provides a comparison of the wireless synchronization systems discussed in this section. Note that these works use different statistical metrics in their respective evaluations, such as *median* (med), *standard deviation* (std), *95th percentile* (95 %) and *RMSE*. Systems for which only the typical upper bound is given in the respective references are denoted with the metric *bound*. GPS and SyncE are listed for comparison in Table 1 and are discussed in Section 2.2.6.5 and in Section 2.2.6.3, respectively.

Furthermore, there are differences in the operation and the evaluation methodology of the systems. In particular, JMB [187] employs a channel bandwidth of 10 MHz as opposed to the channel bandwidth of 20 MHz that is more commonly used in IEEE 802.11a/g/n networks and that is also employed in this work. Thus, JMB's symbol duration doubles

in comparison to a 20 MHz wide channel, so that its phase error accumulated over a frame transmission with 1500 bytes is also doubled. However, as the phase misalignment of JMB is calculated over *all* OFDM symbols of a frame, the given numbers should quite accurately be half of the phase misalignment accumulated at the last OFDM symbol. Assuming that the initial PO is negligible, these two effects compensate one another. Hence, the characteristics of JMB [187] are given without scaling in Table 1. Further, AirShare [4] employs a 10 MHz reference clock, but the duration of a frame with 1500 bytes is given as $\sim$2 ms [4], which actually corresponds to a 20 MHz wide channel. Assuming that clock scaling is applied, the original values of AirShare [4] are given without scaling in Table 1. Also note that the accumulated PO values provided by AirShare are inferred from CFO measurements under the assumption of a one-shot synchronization at the beginning of a frame transmission. In contrast, AirSync [15] continuously tracks the phase of a main AP, so that its accuracy is independent of the actual frame length. Hence, the CFO of AirSync cannot be inferred from the given PO measurements. The system name in the last row of Table 1 is given as *PreSync* [180] for brevity, even though its authors do not use this name. Moreover, the inferred values in Table 1 are based on the assumption that a node is perfectly synchronized in phase at the beginning of an IEEE 802.11 frame with a 1500-byte PHY service data unit (PSDU), so that the phase drift caused by the CFO departs from an initial PO of zero.

2.2.5.3 *Other Wireless Synchronization Systems*

In addition to the prototypes listed in Table 1, there are practical works for which the respective metrics are not given or cannot be derived. Pulsar [63] is a platform for wireless synchronization through UWB that achieves a time synchronization accuracy of 5 ns. Pulsar employs a chip-scale atomic clock (CSAC) which provides excellent long-term stability with a drift of 1 µs in 1.2 days. However, a CSAC is relatively expensive and may not be able to drive the phase of a PLL directly. Hence, Pulsar employs additional subsystems for phase recovery that generate a phase-aligned 1 pulse per second (PPS) output and a 10 MHz clock reference [63].

Furthermore, open-loop wireless synchronization for distributed beamforming works also in practice with an implementation using USRP X310 SDRs [154, 229]. In this system, a primary node transmits a beacon with two sinusoidal tones, which facilitates precise frequency synchronization also under conditions of mobility [154].

The feedback in distributed beamforming can be reduced to a single bit transmitted by the main node, indicating whether the SNR of a multi-user transmission improved or worsened with respect to the prior reception, while the secondary nodes randomly make small changes to their respective phases before each transmission [165]. If the SNR improved, the secondary nodes keep their new phase values, whereas they discard them otherwise. This closed-loop synchronization scheme [163] manages to minimize the feedback even for large numbers of devices, but it requires near-perfect frequency synchronization [165]. This one-bit feedback algorithm still works in practice with a prototype [185] based on USRP SDRs [229] and GNU Radio [84]. Synchronization is achieved with a wireless pilot signal of 5 ms duration and an update rate of 15 Hz that is tracked through an EKF by the secondary nodes [185].

2.2.6 *Wired, Ambient and Optical Synchronization*

Besides time and frequency synchronization via wireless signaling, there are various techniques and protocols for synchronization through wired connections, ambient signals and even optical pulses. These methods do not meet the design criteria for wireless CCT with an opportunistic selection of nodes due to limited flexibility, limited coverage or limited availability, but they are briefly outlined below for the sake of completeness.

2.2.6.1 *Network Time Protocol (NTP)*

The Network Time Protocol (NTP) is a wide-spread protocol for time synchronization over local area networks (LANs) and the Internet that is supported by various computer operating systems and embedded devices [155]. NTP is organized in a hierarchy of time sources, the topmost of which are disciplined, for instance, by GPS or other radio reference signals like the German DCF77, whereas time sources further down in the hierarchy are synchronized to time sources higher up in the hierarchy and are hence typically less accurate [155]. An NTP client and an NTP server exchange messages with timestamps to compensate the client's time offset with respect to the server's clock [155]. The accuracy of NTP varies from a fraction of a millisecond to tens of milliseconds, depending on the intermediate network's jitter [155], which makes it unsuitable for wireless communications [142].

2.2.6.2 *Precision Timing Protocol (PTP)*

The Precision Time Protocol (PTP) is standardized as IEEE 1588 [101] and distributes a time reference in a hierarchy of network nodes with a grandmaster clock at its root [101]. PTP can employ various network technologies as transport, such as Ethernet [102] and User Datagram Protocol (UDP) over Internet Protocol (IP) version 4 or 6 [101]. PTP compensates propagation delays by means of message timestamps, which can be generated by software or by hardware [101]. With hardware-based timestamps, PTP achieves an accuracy better than a microsecond, which makes it suitable for cellular networks, but which also requires specialized network equipment [142]. PTP supports time synchronization better than a nanosecond [101], which can be achieved by combining hardware-based timestamps with SyncE [57]. The latest IEEE 1588 standard [101] includes the *White Rabbit* protocol, which is based on SyncE, to facilitate a time synchronization accuracy in the sub-nanosecond range in conventional fiber networks [82].

2.2.6.3 *Synchronous Ethernet (SyncE)*

Synchronous Ethernet (SyncE) enables frequency synchronization via IEEE 802.3 [102] and is defined by the Telecommunication Standardization Sector of ITU (ITU-T) in cooperation with IEEE. The SyncE standard is fragmented into three parts: the architecture [189], timing and clock characteristics [190] and the synchronization through messages [191]. Technically, SyncE transfers a frequency reference from a main node by embedding a clock signal into IEEE 802.3 PHY signals, which requires hardware support by specialized network equipment. Applications for SyncE comprise, for instance, MBSFN [136] and CoMP [147, 148] in LTE [137] cellular networks [91]. The accuracy of SyncE is better than 1 ppb [142] and can be enhanced further by using multiple network paths [193].

2.2.6.4 *Power Line Network*

The electromagnetic fields emitted from power line networks can also serve as a reference for frequency synchronization, allowing for an accuracy better than 10 ppb in a low-power implementation [37]. Besides passively overhearing the utility frequency, a power line network can also be leveraged by actively injecting a clock reference signal at a higher frequency, e.g., at 10 MHz [254].

2.2.6.5 *Global Positioning System (GPS)*

The NAVSTAR Global Positioning System (GPS) is a first-generation global navigation satellite system (GNSS) that draws on clock synchronization and propagation delay measurements by means of satellite transmissions with an atomic clock source [175]. With this, GPS enables a receiver to localize its position and to synchronize in time and frequency with an atomic clock. GPS can be employed for synchronization of BSs in cellular networks [142, 147]. However, GPS signals do not penetrate buildings, so that their use for indoor applications is limited. Further, GPS signals are susceptible to jamming [142]. The mean time synchronization accuracy of GPS in a line-of-sight (LOS) experiment over 18 hours is 7 ns with a standard deviation of 37.1 ns and a maximum absolute deviation of 171 ns [130]. The frequency synchronization is accurate to within 0.218 ppb in 90 % of the measurements and deviates maximally by 1.37 ppb [130].

2.2.6.6 *Atomic Optical Clocks*

State-of-the-art atomic optical clocks offer a frequency accuracy on the order of 10^{-18} and can be synchronized via optical pulses with an accuracy better than a femtosecond over several kilometers [59]. Optical synchronization also works under conditions of mobility and can suppress Doppler shifts, as demonstrated with a retroreflector mounted on a quadcopter [21]. Nevertheless, optical atomic clocks and their synchronization mechanisms are still to be considered as research prototypes, but they might become an attractive choice for synchronization of wired and wireless networks in the future.

2.2.7 *Unsyntonized Concurrent Transmissions*

Most systems that leverage concurrent transmissions from distributed nodes rely on time and frequency synchronization so that the superimposed signals can be processed with conventional minimum mean square error (MMSE) or zero-forcing equalization by a receiver. This section presents works that break with this fundamental principle to explore new ways of wireless communications.

ForkPHY [264] addresses MU-MIMO uplink transmissions in an IEEE 802.11 network with one AP and several clients. While this scenario is supported by IEEE 802.11ax [105], the synchronization creates communication overhead and is considered complex [264]. ForkPHY eliminates this communication overhead by employing a spatial filter at the multi-antenna AP to separate MU-MIMO uplink transmissions that are synchronized neither in frequency nor in time. To detect the individual superimposed signals and to estimate their respective symbol timings and CFOs, ForkPHY additionally employs an interference-alleviation filter, which works best when initialized with the spatial filtering

coefficients from a previous reception, but which can also be constructed otherwise [264]. ForkPHY is validated in testbed experiments with USRP SDRs [229] and GNU Radio [84]. The TVZF equalizer presented in Chapter 6 relaxes the synchronization requirements for IEEE 802.11 signals, too, but it only allows for a TO within the OFDM GI and it only works with identical signal copies. However, as opposed to ForkPHY, TVZF works already with a single Rx antenna, which makes it suitable also for small mobile devices. Further, TVZF does not leverage repetitive receptions from the same nodes, i.e., it treats each frame reception separately to support scenarios with opportunistic transmitters, such as CCT-based network flooding.

SAM [220] is another work that enables an IEEE 802.11 AP to deal with asynchronous MU-MIMO uplink transmissions from multiple clients. In contrast to ForkPHY, SAM follows a cross-layer approach that involves both the PHY and the MAC, and it requires concurrent transmissions to arrive in a staggered fashion so that the preambles can be processed successively. At the PHY, SAM uses interference nullifying in combination with interference cancellation to recover and to decode all the superimposed signals. At the MAC, SAM extends carrier sense multiple access (CSMA) to allow stations to start their uplink transmission one after another until a maximum number of streams is reached that is limited by the number of antennas at the AP. SAM is validated to work in a testbed based on the Sora platform with two AP antennas [220]. As opposed to SAM, the TVZF equalization presented in Chapter 6 does not necessitate TOs between concurrent transmissions and works with superimposed preambles with identical signal copies to facilitate opportunistic CCT.

Instead of synchronizing nodes prior to distributed beamforming, the relative phase drifts between unsyntonized transmitters and receivers can also be integrated by the transmitters into precoding matrices with CSI predictions, which improves the feedback efficiency [33, 34]. To this end, the receivers feed back CSI estimates along with phase drift predictions, which they obtain by tracking the phases of training sequences from the transmitters with a Kalman filter [34]. The transmitters generate time-varying precoding matrices that incorporate the predicted phase drifts [34]. To mitigate suboptimal local tracking, the statistical coupling of phase and frequency offsets can also be exploited by unifying the tracking across receivers [33]. Still, this variant only provides gains when the feedback latency is low [33]. Note that the TVZF technique presented in Chapter 6 also draws on time-varying channel coefficients to compensate for CFOs in CCTs. However, TVZF is designed for MISO scenarios rather than spatial filtering and works on the receiver side without feedback.

Diversity gains from unsyntonized distributed transmitters can also be leveraged by means of on-off keying (OOK) and by repeating each symbol multiple times [213, 214]. With this, the phase alignments of CCTs continuously change, so that the symbols add constructively at least at some point in time. For the symbol transmissions subject to constructive interference, the *on* state of the OOK encoding is better distinguishable from the *off* state also without equalization [213]. Still, transmitting each symbol 10 times [214] might introduce a larger overhead than synchronization and spatial filtering.

Further, there are some works that consider multiple unsyntonized relays forwarding a message from a source to a destination via CCT [20, 129, 232, 234]. One approach to facilitate the decoding at the receiver is *delay diversity*, where the relays introduce small delays, i.e., the receiver gets multiple copies of each symbol at different times [20, 232, 234].

However, this approach still assumes symbol-level time synchronization and the receiver must know the delays of the respective relays [232, 234]. Further, the receiver must know the CFOs and the channel coefficients of the relays [232], which can be estimated, for instance, through orthogonal training symbols [234]. Delay diversity is demonstrated to work in simulations for single-carrier modulation with a narrow-band and flat-fading channel [232, 234] as well as for an OFDM PHY [20]. Besides delay diversity, the impact of multiple CFOs in a CCT can also be mitigated through an extension of the OFDM GI, which, however, significantly diminishes the spectral efficiency [129]. Further, this method requires knowledge of the symbol timings, the CFOs and the channel coefficients of the different transmitters [129]. A Viterbi equalizer can also mitigate the impact of multiple CFOs, but it requires the receiver to have perfect CSI knowledge, too [256]. In contrast to these works, the TVZF equalizer presented in Chapter 6 comprises the estimation of the TOs, CFOs and channel coefficients of multiple transmitters in a CCT and does not require any predefined TOs. Further, TVZF does not leverage a custom preamble format with orthogonal training symbols but processes only superimposed IEEE 802.11 PHY signals without any modifications to the standard.

A critical component of TVZF is the spectral estimation of noisy signal vectors of limited length obtained from the pilot subcarriers to retrieve the frequencies, amplitudes and phases of an unknown number of complex exponentials. There are spectral estimation approaches like MUltiple SIgnal Classification (MUSIC) [38], forward-backward linear prediction [228] and eigenstructure methods [97] that may be applicable with some modifications. Still, the spectral estimation algorithm should work with short signals that comprise only a small fraction of an oscillation cycle and it should detect signal components with closely spaced frequencies. In addition, it should leverage four signals from the pilot subcarriers in combination, which expose the same frequencies but different amplitudes and phases. This thesis proposes IIC as a spectral estimator that draws on SIC [233] to address these constraints and to leverage the given problem structure.

2.3 Mechanisms for Integration with CCT

CCT-based network flooding allows for rapid data dissemination as well as for implicit time synchronization in WMNs, as demonstrated with Glossy [73]. In practical scenarios, CCT may serve as a primitive for higher-level mechanisms with more specific purposes. This section discusses extended mechanisms that either explicitly leverage CCT-based network flooding or that might benefit from an integration of CCT.

Section 2.3.1 presents various mechanisms for data dissemination and aggregation. Further, Section 2.3.2 presents routing mechanisms that draw on CCT to facilitate fast and robust route discovery and data delivery in WMNs. Moreover, Section 2.3.3 discusses multi-hop synchronization with an emphasis on CCT-based techniques that address the compensation of propagation delays. The prototype presented in Chapter 4 readily provides a low-level MAC protocol and is suited for an integration of such mechanisms, while the prototype presented in Chapter 5 could be extended similarly.

2.3.1 *Data Dissemination and Aggregation*

Low-Power Wireless Bus (LWB) [72] mimics a shared bus system to connect several mobile nodes with each other and to enable data transfers with one-to-many, many-to-one and many-to-many traffic. LWB employs a dedicated host to maintain and to distribute a global communication schedule. LWB draws on Glossy [73] to disseminate data quickly among all nodes and also leverages its implicit time synchronization feature for accurate transmission scheduling [72]. LWB is shown to perform similarly or significantly better than seven other combinations of routing and link-layer protocols in testbed experiments. With this, LWB is a versatile solution for multi-hop data exchange and distributed control algorithms that is robust to external interference, node failure and mobility [72].

Sparkle [257, 260] is another wireless interconnection system based on Glossy [73] that is specifically designed to serve multiple control loops with periodic traffic in parallel while taking their respective quality of service (QoS) demands into account. To this end, Sparkle comprises *WSNShape*, a topology control technique that exploits the capture effect to find reliable multi-hop paths between nodes. Further, Sparkle comprises *PRRTrack* to adaptively switch between different *WSNShape* modes in terms of transmission power and path counts. With this, Sparkle maintains the reliability of traffic flows in varying environments, while enhancing the overall energy consumption and latency [257, 260].

Splash [62] is a flooding protocol based on Glossy [73] that optimizes throughput by *tree pipelining*, i.e., it concurrently forwards data from different levels of a tree topology on different frequency channels. Splash also draws on transmission density diversity, opportunistic overhearing, channel-cycling and XOR coding to improve reliability [62]. Ripple [257, 259] is a flooding protocol that improves the throughput further by a factor of two to three in comparison to Splash [62]. Contrary to Splash [62], Ripple [257, 259] assigns channels to packets rather than to groups of nodes, which eliminates the overhead of initial channel assignments to nodes at certain tree levels. In addition, Ripple employs a Reed-Solomon code to improve reliability. To meet QoS demands, Ripple counterbalances throughput and reliability by adapting its transmission interval [257, 259].

The energy cost of network flooding protocols like Splash [62] and Ripple [257, 259] could be further reduced by tuning the transmission power of individual nodes, e.g., through a decentralized algorithm based on game theory [160, 161]. However, as opposed to drawing on maximum-ratio combining (MRC) to leverage successive receptions of a message [160], CCT combines signals through superposition at the PHY, which improves the utilization of spectral resources and thus the achievable throughput.

Chaos [124] is another protocol based on Glossy [73] that allows for network-wide agreement and data aggregation. To this end, Chaos leverages the capture effect to decode diverging data in concurrent transmissions. Further, Chaos employs INP, i.e., it processes received data decentrally at the nodes, which in turn propagate their locally obtained results further through the network [124].

Since all these protocols either leverage CCT-based network flooding or may achieve further performance gains with it, they can potentially be integrated with the prototypes of this work that are presented in Chapter 4 and in Chapter 5. However, OFDM is more susceptible to interference than DSSS, as discussed in Section 2.1.2. Therefore, the capture effect can possibly not be exploited as easily with an OFDM-based PHY.

2.3.2 *Routing*

As the communication range of mobile devices is limited due to power constraints and regulations, WMNs are a flexible solution for data delivery between devices that cannot reach each other directly [65, 116, 209]. Applications of WMNs are manifold and comprise, for instance, vehicular networks and intelligent transportation systems (ITSs) based on IEEE 802.11 [104, 262], WSNs based on IEEE 802.15.4 [107, 258], emergency communication systems like the Serval Project [80, 131, 207], Internet sharing via Freifunk [76, 152, 182] and traffic offloading in cellular networks through device-to-device (D2D) communications [169, 208]. Still, WMN routing algorithms like Ad-hoc On-demand Distance Vector (AODV) [56], Dynamic Source Routing (DSR) [98] and Optimized Link State Routing (OLSR) [55] are susceptible to mobility and interference [65], which limits the scalability of WMNs in terms of size, reliability and throughput [116, 209]. Further, reactive routing protocols rely on network flooding for route discovery [56, 98], which deteriorates the network capacity and which can lead to broadcast storms [171, 226].

CCT-based flooding has the potential to overcome the problems of conventional WMN routing algorithms like AODV [56], as shown with OLA Routing On-Demand (OLAROAD) [47, 224]. Similar to AODV [56], OLAROAD [224] also floods a route request (RREQ) from a source node to discover a route to a destination node if no route exists. However, with CCT, OLAROAD [224] can disseminate the RREQ faster and consume less spectral resources in comparison to CSMA/CA-based network flooding. After that, AODV [56] sends a unicast route reply (RREP) message from the destination back to the source along the hops over which the RREQ was received first by the destination. OLAROAD [224] also transmits a RREP message, but it does so via CCT and selects nodes for relaying and route construction that lie in the intersections of the corresponding decoding levels of the RREQ and RREP transmissions. Finally, conventional routing algorithms like AODV [56] and DSR [98] maintain unicast routes, that can easily break due to node mobility or node outage. On the contrary, OLAROAD employs a series of node groups in a corridor between the source and the destination, which improves the robustness against mobility and node outage due to the redundancy of relays [224]. With this, OLAROAD reduces the end-to-end delay in comparison to AODV due to less route refreshes and due to a hop count reduction through CCT, especially at a high node density [224]. In testbed experiments with USRP SDRs [229] and GNU Radio [84], OLAROAD shows significant improvements of the round-trip time (RTT) and the packet delivery ratio in a line topology with 18 nodes using a BFSK PHY [47]. Besides OLAROAD, OLA Concentric Routing Algorithm (OLACRA) [225] is a similar routing algorithm that is tuned towards energy savings in WSNs.

Sparkle [257, 260] employs *WSNShape* as a routing algorithm to find a corridor of nodes between a source and a destination for data transfers by means of Glossy [73]. In contrast to OLAROAD [224], *WSNShape* identifies different reliable paths from a source to a destination by exploiting the capture effect. Sparkle combines multiple such paths according to a path count parameter [257, 260]. Further, Sparkle offers precise control of the degree of redundancy per route by employing *PRRTrack*, which maintains a desired level of reliability by adapting the path count and the transmission power dynamically. With this, Sparkle can operate multiple routes in parallel while maintaining a global transmission schedule for periodic data traffic [257, 260].

2.3.3 *Multi-Hop Synchronization*

Time synchronization is key in distributed systems to run distributed algorithms and to consistently assign timestamps to events [67, 73, 77, 146]. Particularly the dURLLC paradigm introduced in Chapter 1 envisions interactions between distributed devices in the physical world, driven by multi-hop data dissemination and time synchronization. While the mechanisms of this thesis have an emphasis on CCT-based data dissemination, they can simultaneously be used for multi-hop synchronization in time and frequency. This section discusses different multi-hop synchronization methods, how they can be enhanced through CCT-based network flooding and how propagation delays can still be compensated when employing CCT-based techniques.

In contrast to synchronizing secondary nodes to a time reference from a master clock, the reference-broadcast synchronization (RBS) [67] establishes a relative notion of time between all receivers of a broadcast message, which suffices to obtain the relative timing of events at different nodes. RBS extends to multi-hop networks by converting the time bases of different broadcast domains that connect the locations of the respective events. In this way, RBS can also inject an absolute time reference into a network [67].

The Timing-sync Protocol for Sensor Networks (TPSN) [77] constructs a spanning tree with a master clock at its root and synchronizes pairs of nodes along the graph edges. In contrast to TPSN, the Flooding Time Synchronization Protocol (FTSP) [146] adapts its topology dynamically while a root node floods synchronization messages.

The aforementioned algorithms are sensitive to timestamping errors [67, 77, 146], which can be caused by the propagation delay [67, 146] or by a jitter in the interrupt handling of the hardware platform [67, 77]. Propagation delays can be eliminated through sender and receiver timestamps via ACK frames [77], but this method is infeasible for the approaches based on broadcast messages and not even meaningful if other error sources introduce larger TOs [67, 146]. The stability of time synchronization can be enhanced through frequency synchronization of the logical clocks by means of linear regression of a sequence of message timestamps [67, 146, 255]. In addition, post-facto synchronization can extrapolate TOs also for past events, which saves energy in sensor networks that need synchronization only sporadically [67, 77]. The PulseSync [128] protocol is based on FTSP but it reduces the global TO, i.e., the maximum absolute TO of any two clocks in the network, by flooding timestamped synchronization messages quickly through the network. Glossy [73] excels in low-latency flooding and improves the utilization of spectral resources through CCT rather than managing interference through scheduling.

A major disadvantage of network time synchronization based on flooding via CCT is that the nodes do not learn the network structure and thus cannot compensate for propagation delays, which limits the synchronization accuracy. Reverse Flooding [223] addresses this issue by letting each network node estimate its propagation delay with respect to the root node through additional messages. Each node periodically broadcasts a propagation delay request, which is answered by the previous-hop neighbors concurrently. This allows the requestor both to measure the propagation delay to its previous-hop neighbors and to receive from these neighbors their respective propagation delay with respect to the root node [223]. Since the values sent by the previous-hop neighbors may differ slightly, a special encoding for the DSSS-based IEEE 802.15.4 PHY is employed that allows the requestor to extract the range of values from a concurrent transmission.

The requestor adds this information to its propagation delay measurement to get an accumulated measure with respect to the root node. Still, a major drawback of Reverse Flooding is that it requires a global schedule for sending the request messages [223].

Cooperative Analog and Digital (CANDI) [46] is a network time synchronization protocol that compensates for propagation delays by means of two flooding phases. During the first phase, the root node disseminates a reference timestamp throughout the network via CCT-based network flooding. In this process, each node additionally estimates the propagation delay to its respective next-hop neighbors by overhearing their retransmission. Then, the root node initiates a second flooding phase in which each node forwards its estimate of the accumulated propagation delay of its next-hop neighbors with respect to the root node. To this end, a node combines its reception from its previous-hop neighbors during the second phase with its propagation delay estimate from the first phase. As the propagation delays estimated by each node may vary slightly, CANDI employs an analog technique called Semi-Cooperative Spectrum Fusion (SCSF) that allows a receiver to obtain a weighted average of the different values that are transmitted concurrently. Testbed experiments with USRP SDRs [229] and GNU Radio [84] using a BFSK PHY show that CANDI reduces the worst-case RMS timing error by up to 29.5 % [46]. While the prototype presented in Chapter 4 readily provides global time synchronization similar to Glossy [73], the accuracy of multi-hop time synchronization could potentially be further enhanced through an integration of a technique to compensate for propagation delays, such as CANDI [46].

Besides synchronization to a reference, distributed nodes can also collaboratively find an agreement on a common notion of time. The CoCo [42] algorithm finds a consensus on a virtual clock with time and frequency synchronization by flooding synchronization information via SCSF multiple times back and forth through the network, which is also effective for an OFDM PHY in simulations with ping-pong transmissions between two clusters of nodes [43]. Further, there are also biologically-inspired approaches that imitate the synchronized flashing of fireflies [188] by means of pulse-coupled oscillators [244].

2.4 IEEE 802.11 Standard

The mechanisms and prototypes presented in this thesis are designed to work with PHY signals compliant with the IEEE 802.11 standard [104]. This section provides background information on selected portions of IEEE 802.11 and respective processing techniques. Section 2.4.1 starts with an overview of the different DSSS-based and OFDM-based IEEE 802.11 PHYs. Section 2.4.2 outlines the PHY protocol data unit (PPDU) format of the Non-HT PHY used by IEEE 802.11g and also points out differences of IEEE 802.11n. Section 2.4.3 describes the frame detection and the symbol timing acquisition performed by a receiver to synchronize in time with an ongoing frame reception. Section 2.4.4 describes the carrier frequency acquisition performed by a receiver to synchronize in frequency with an ongoing frame reception. Section 2.4.5 describes the CSI acquisition performed by a receiver to equalize multi-path effects occurring on a wireless channel. Section 2.4.6 briefly explains the concepts of coherence bandwidth and coherence time. Section 2.4.7 outlines the MAC header format of IEEE 802.11. Section 2.4.8 explains how the short interframe space (SIFS) is defined in this thesis.

2.4.1 *Overview of the IEEE 802.11 PHYs*

The IEEE 802.11 standard [104] comprises various PHYs [181, Chapter 1.1]. The initial IEEE 802.11 standard from 1997 defines an infrared (IR) PHY, a frequency-hopping spread spectrum (FHSS) and a DSSS PHY [93] [181, Chapter 1.1], the first two of which, however, are no longer included since the 2016 revision of the standard [103]. The IEEE 802.11 DSSS PHY [104, Section 15], which offers 1 and 2 Mbit/s in the 2.4 GHz Industrial, Scientific and Medical (ISM) band, is still widely supported for backwards compatibility. IEEE 802.11b, introduced in 1999, enhances the DSSS PHY with CCK to increase the PHY data rate up to 11 Mbit/s in the 2.4 GHz ISM band [104, Section 16] [181, Chapter 1.1]. IEEE 802.11a, introduced in 1999, has an OFDM PHY operating in the 5 GHz band that offers PHY data rates from 6 Mbit/s to 54 Mbit/s [104, Section 17] [181, Chapter 1.1]. IEEE 802.11g, introduced in 2003, adopts the OFDM PHY of IEEE 802.11a for use in the 2.4 GHz ISM band, while maintaining interoperability and coexistence with legacy IEEE 802.11 DSSS and IEEE 802.11b devices [104, Sections 17 and 18] [181, Chapter 11.5.1]. IEEE 802.11n, introduced in 2009, has an OFDM PHY with various enhancements, including 20 and 40 MHz wide channels, operation in the 2.4 GHz and the 5 GHz bands, MIMO with up to four spatial streams and MAC functions to increase throughput and to save energy [104, Section 19] [181, Chapter 1.1]. IEEE 802.11ac, introduced in 2013, has an OFDM PHY for operation in the 5 GHz band and introduces enhancements such as up to 160 MHz wide channels, MIMO with up to eight spatial streams and downlink (DL) MU-MIMO [104, Section 21]. IEEE 802.11ax, introduced in 2021, has an OFDM PHY that operates from 1 GHz to 7 GHz with up to 160 MHz wide channels, offering up to eight MIMO spatial streams as well as DL and uplink (UL) MU-MIMO [105, Section 27].

IEEE 802.11p, introduced in 2010, adopts the OFDM PHY of IEEE 802.11a in order to facilitate dedicated short range communications (DSRC) in vehicular networks in a frequency band at 5.9 GHz with a channel bandwidth of 10 MHz [104, Section 17] [262]. IEEE 802.11bd is going to be the successor of IEEE 802.11p and adopts improvements of other IEEE 802.11 PHYs, such as FEC through LDPC from IEEE 802.11n [262].

The PHY of IEEE 802.11n is referred to as high throughput (HT) [104, Section 19], in accordance with the name of the High Throughput Task Group (TGn) that developed the amendment [181]. Correspondingly, the PHY of IEEE 802.11a, IEEE 802.11g and IEEE 802.11p is named Non-HT [104, Section 17] [181]. Similarly, the PHY of IEEE 802.11ac is referred to as very high throughput (VHT) [104, Section 21] [181]. Furthermore, the PHY of IEEE 802.11ax is referred to as high efficiency (HE) [105, Section 27].

This work primarily utilizes the Non-HT PHY in the 2.4 GHz band for various reasons. First, one goal of this thesis is to explore the feasibility of CCT with OFDM, for which the Non-HT PHY can serve as a generic representative. In particular, the PHY header format of the Non-HT PHY is the least complex out of the OFDM PHYs of IEEE 802.11, which makes prototyping easier. Second, the PHY header format of the Non-HT PHY is shorter than those of the HT or VHT PHYs, which results in the shortest frame duration for small payloads, which in turn minimizes latency. Third, another goal of this thesis is to maintain compliance with the IEEE 802.11 PHYs, the feasibility of which can best be explored with a basic format. Nevertheless, the solutions presented in this thesis are also applicable for higher bandwidths and work even with up to four MIMO spatial streams in a 40 MHz wide channel using the HT PHY, as presented in Section 5.4.4.

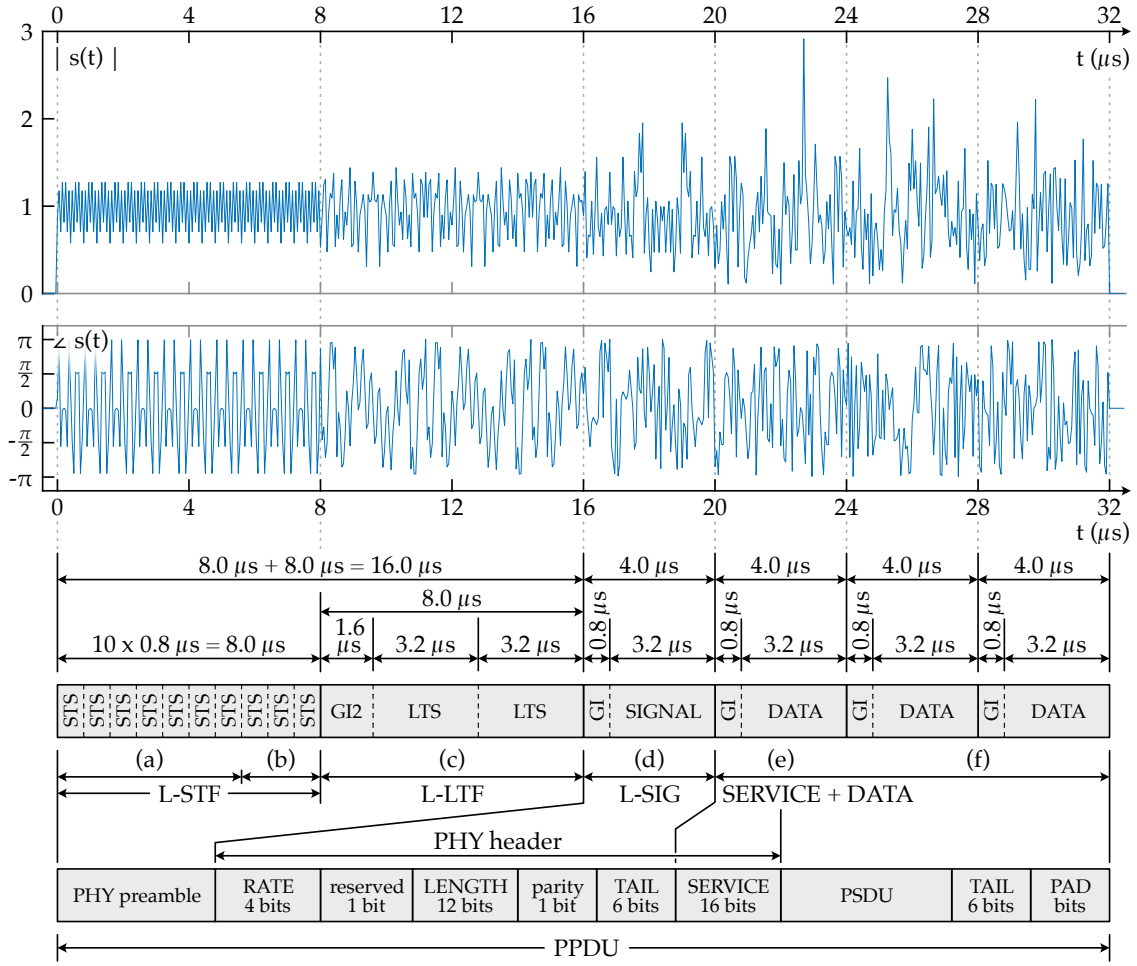


Figure 6: Non-HT PPDU format: A receiver utilizes the different sections for (a) signal detection, automatic gain control (AGC), diversity selection, (b) coarse symbol timing acquisition, coarse frequency offset acquisition, (c) fine symbol timing acquisition, fine frequency offset acquisition, channel estimation, (d) PHY parameter extraction, (e) descrambler initialization and (f) PSDU extraction [104, Section 17.3, Figures 17-1 and 17-4].

2.4.2 *Non-HT PPDU Format*

The main function of a PHY is the conversion of a PHY service data unit (PSDU) consisting of a sequence of bytes to and from a PHY protocol data unit (PPDU) for transfer over a wireless medium. In particular, the Non-HT PPDU format consists of a PHY preamble, a PHY header and the encoded PSDU. Figure 6 gives an overview of the Non-HT PPDU format [104, Section 17.3, Figures 17-1 and 17-4].

The PHY preamble consists of a legacy short training field (L-STF) and a legacy long training field (L-LTF). The term *legacy* indicates that the fields are part of a Non-HT PPDU, in contrast to the corresponding fields of the more recent PHY formats like HT or VHT. The L-STF consists of ten repetitions of a 0.8 µs long short training sequence (STS), which contains fixed values on 12 subcarriers modulated through binary phase-shift keying (BPSK). A receiver uses the L-STF to detect an incoming frame while listening on its RF interface, typically through auto-correlation. In addition, the L-STF is used by the automatic gain control (AGC) to adjust the Rx gain such that the resolution of the ADC

MCS	modulation	coding rate	PHY data rate (Mbit/s)
0	BPSK	1/2	6
1	BPSK	3/4	9
2	QPSK	1/2	12
3	QPSK	3/4	18
4	16-QAM	1/2	24
5	16-QAM	3/4	36
6	64-QAM	2/3	48
7	64-QAM	3/4	54

Table 2: MCS (i.e., RATE) parameters of the IEEE 802.11 Non-HT PHY (20 MHz channel spacing) [104, Section 17.3, Tables 17-4 and 17-6]

is optimally utilized without clipping. If multiple Rx chains are attached, a receiver may perform diversity selection, e.g., by means of SNR estimates of the respective L-STFs. The L-LTF consists of two copies of a long training sequence (LTS) that are prepended by a guard interval (GI) with a cyclic prefix (CP) [243] with the double length of the GI of the subsequent OFDM symbols. The LTS contains fixed values on 52 BPSK-modulated subcarriers. A receiver uses the L-LTF to acquire fine estimates of the symbol timing and of its CFO relative to the transmitter and to estimate the channel coefficients of all 52 subcarriers used by the DATA OFDM symbols [104, Section 17.3].

The PHY header consists of the legacy SIGNAL field (L-SIG), which is contained in a single OFDM symbol with BPSK rate 1/2, and the SERVICE field, which is located in the first 16 bits of the subsequent OFDM symbol for DATA. The generation of the L-SIG OFDM symbol is carried out like the generation of DATA OFDM symbols with the exception that scrambling is not applied, i.e., the bits undergo convolutional encoding, interleaving, modulation mapping, pilot insertion, inverse FFT (IFFT) and CP insertion into the prepended GI. The end of the SIGNAL field contains 6 TAIL bits to facilitate robust convolutional decoding [104, Section 17.3]. With this, a receiver can recover the L-SIG immediately after its reception, extract the PHY parameters and start decoding DATA while still receiving in-phase quadrature (IQ) samples of the incoming frame.

The RATE field encodes the combination of a modulation scheme and a coding rate that is used in the DATA OFDM symbols of the PPDU. Note that later IEEE 802.11 PHYs like HT and VHT use the term *MCS*, whereas the Non-HT PHY specification uses the term *RATE*. For convenience and in accordance with the Matlab wireless LAN (WLAN) toolbox [150], this thesis uses the term *MCS* to index the available PHY data rates also for the Non-HT PHY, as listed in Table 2 [104, Section 17.3, Tables 17-4 and 17-6].

The LENGTH field defines the number of octets (i.e., bytes) of the PSDU. In combination with the MCS, it can be used to calculate the number of DATA OFDM symbols.

The first seven bits of the SERVICE field are used by a receiver to initialize the descrambler. Note that transmitters shall initialize the scrambler to a pseudorandom nonzero state [104, Section 17.3.5]. However, to generate the same bit patterns across transmitters for CCT, the mechanisms of this work lock the scrambler to a fixed value.

The TAIL bits are used to return the convolutional decoder of a receiver to a zero state, which improves its robustness to decoding errors. The PAD bits fill the remainder of the last OFDM symbol and have no further relevance for a receiver [104, Section 17.3].

The HT-mixed format (HT-MF) and the VHT PPDU format both start with an L-STF, an L-LTF and an L-SIG for coexistence with legacy devices [104, Sections 19.3 and 2.3]. A receiver capable of HT and VHT PHYs can distinguish between Non-HT, HT-MF and VHT PPDUs by the symbols following the L-SIG. In HT-MF PPDUs, the L-SIG is followed by an HT short training field (HT-STF) and one or several HT long training field (HT-LTF) symbols, the exact number of which depends on the number of MIMO spatial streams employed by the selected MCS [104, Section 19.3, Table 19-13]. While an HT-MF PPDU allows for higher data rates than a Non-HT PPDU, its PHY preamble is longer, which impairs the latency with short PSDUs in CCT-based flooding, compared to Non-HT.

2.4.3 *Frame Detection and Symbol Timing Estimation*

To receive IEEE 802.11 frames incoming from an Rx RF interface, a receiver has to detect such frames in a stream of noisy complex-valued IQ samples. Further, it has to acquire an accurate symbol timing to prevent ISI during decoding. Both of these tasks can be performed through filtering, as outlined below.

Figure 7 shows a data frame with additive white Gaussian noise (AWGN) at 30 dB SNR, separated into (a) magnitude and (b) phase. The first sample of a frame is also referred to as the start of frame (SOF) in this thesis and its TO amounts to 4 µs with respect to the beginning of the IQ stream in the example, as visible in Figure 7a. Note that the term *frame* actually refers to a MAC protocol data unit (MPDU), i.e., a transmission unit of the link layer, but in the context of the PHY, this term may also be used to refer to an entire PPDU in accordance with the IEEE 802.11 standard [104].

To detect an L-STF in a stream of IQ samples, a receiver may leverage the repeated occurrence of the STS in the L-STF [203]. Figure 7c shows the decision metric of the `wlanPacketDetect()` function of the Matlab WLAN toolbox [150], which auto-correlates the received signal with a delayed version in a double sliding window and normalizes the result by the energy received in the auto-correlation window to become independent of the absolute power level. This function outputs a high signal level for the duration over which both sliding windows are filled with an entire STS. With this, a receiver can detect the presence of an IEEE 802.11 frame when a threshold is exceeded for a certain number of output samples within a certain time span [150], which is also effective in lower SNR regimes and under conditions of interference. Further, the first peak exceeding the threshold can be used as a coarse estimate of the SOF for the PPDU symbol timing.

To acquire a fine SOF estimate, a receiver may leverage the L-LTF, which consists of two repetitions of the LTS prepended by a long GI filled with a CP [243]. To this end, a receiver typically applies a matched filter [173] to search for the occurrence of the LTS in a portion of the received signal in which the L-LTF is expected according to the coarse symbol timing estimation. Figure 7d shows the magnitude of the output of a matched filter, computed as the convolution of the entire received signal with a time-reversed complex-conjugated LTS. This filter produces two magnitude peaks at a distance of 3.2 µs, indicating the positions of the two LTS symbols. With this, a receiver can accurately determine the SOF by means of the known symbol durations.

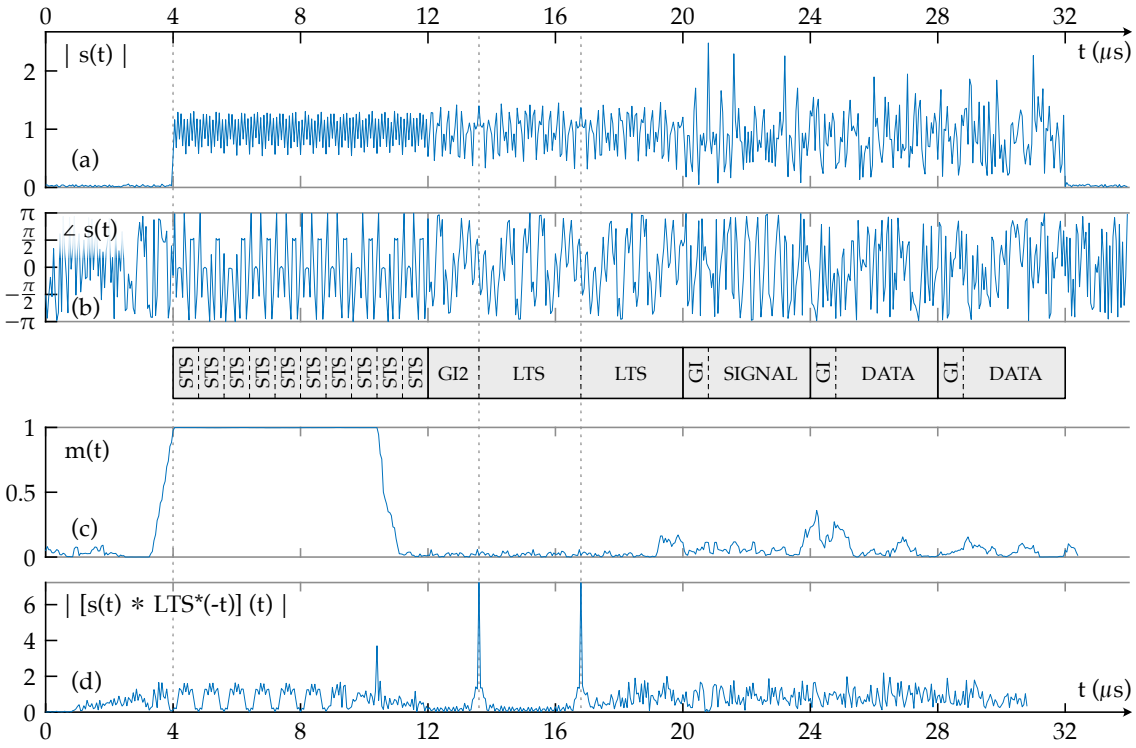


Figure 7: IEEE 802.11 Non-HT frame detection and symbol timing estimation: A Non-HT PPDU with AWGN at 30 dB SNR is shown as (a) magnitude and (b) phase. (c) The L-STF is detected through auto-correlation over a double sliding window. (d) The fine symbol timing is estimated with a matched filter searching for the LTS symbols of the L-LTF.

Note that the example in Figure 7 is generated with AWGN at a high SNR level. In practice, however, a received signal may additionally be subject to multi-path effects, i.e., a receiver gets multiple slightly delayed and superimposed copies of a signal. Thus, multiple magnitude peaks may appear in the output of the LTS-matched filter, necessitating a decision criterion in receiver implementations, e.g., selecting the highest peak pair exceeding a threshold. Further, the filter outputs shown in Figure 7 are aligned with the received waveform for convenience, whereas in real-time systems like in Chapter 4, their outputs are delayed with respect to the input signal according to the sliding window size or the impulse response length of the respective filter.

2.4.4 *CFO Estimation and Correction*

In order to avoid ICI, a receiver typically synchronizes in frequency to a transmitter before applying a fast Fourier transform (FFT) on the received OFDM symbols [13, 183]. With a CFO between a transmitter and a receiver, the phase of the received signal drifts, i.e., there is a continuous rotation of the phase over time [13, 159]. A receiver can estimate the phase difference $\Delta\phi$ between two successive identical symbols from the phase of the complex correlation between the two symbols in the frequency domain [159]. With this, the CFO estimate $\Delta\hat{f}$ can be obtained through the relation $\Delta\hat{\phi} = 2\pi\,\Delta\hat{f}\,T$, where T is the symbol duration, i.e., $T_{\text{STS}} = 0.8\,\mu\text{s}$ for an STS and $T_{\text{LTS}} = 3.2\,\mu\text{s}$ for an LTS. Note that the phase difference $\Delta\phi$ must not exceed $\pm\pi$ over the duration T to be unambiguous, i.e.,

the CFO must not exceed half the subcarrier spacing. Thus, with a Non-HT PHY and a channel bandwidth of 20 MHz, the CFO must be within ±625 kHz when estimated by use of the L-STF and within ±156.25 kHz when estimated by use of the L-LTF.

Next, an IEEE 802.11 receiver compensates for the estimated CFO over the entire frame duration, typically by shifting the frequency through baseband processing, i.e., by multiplying the IQ samples with a complex exponential [15, 79, 89, 135, 187]. With this, a receiver mitigates ICI when executing an FFT on the OFDM symbols of the PPDU.

2.4.5 *Channel Estimation and Equalization*

Wireless signals are subject to propagation effects, such as attenuation, reflection and scattering [200]. Thus, a receiver sees the superposition of multiple slightly delayed copies of a signal, arriving as LOS and NLOS components [200]. These signal copies add constructively or destructively at different frequencies, causing frequency-selective fading at a receiver [263]. An OFDM receiver compensates for such fading effects by means of channel equalization with coefficients describing the complex-valued gains at the respective subcarriers. These channel coefficients are obtained through training symbols [263]. Established equalization methods are zero-forcing and MMSE [218].

Prior to decoding an IEEE 802.11 Non-HT frame, a receiver obtains the channel coefficients for all 52 used subcarriers by performing an FFT over the LTS symbols and by reverting the fixed BPSK-modulated sequence. Receiver implementations usually combine the channel coefficients from the two LTS symbols, e.g., through averaging, to improve the channel estimate [263]. An IEEE 802.11 Non-HT frame contains training symbols only at the beginning of the PPDU, but the channel coefficients may still be adjusted during decoding, e.g., through residual CFO estimation via pilot tracking.

2.4.6 *Coherence Bandwidth and Coherence Time*

The coherence bandwidth is a frequency interval in which the channel transfer function is consistent [218]. The transfer function across the entire OFDM channel bandwidth is typically frequency-selective, whereas it is flat-fading for individual subcarriers [243]. The coherence bandwidth is inversely proportional to the channel delay spread (CDS), i.e., multi-path environments make the channel frequency-selective.

Similarly, the coherence time is the time span during which the channel coefficients do not change significantly. It depends on the mobility of a node with respect to its environment and is proportional to the inverse of the Doppler spread [218]. At 50 km/h, for instance, the Doppler spread is 111.1 Hz, resulting in a channel coherence time of 1.61 ms [109]. At this mobility, the channel can be considered to be slow-fading for Non-HT frames of up to 400 OFDM symbols, i.e., a PSDU of up to 1182 bytes at MCS 0.

2.4.7 *MAC Frame Format*

An IEEE 802.11 MAC protocol data unit (MPDU) consists of a MAC header, a frame body and an FCS [104, Section 9.2]. Note that the prototypes and mechanisms presented in this thesis work with broadcast data frames. Figure 8 depicts the general MAC frame format of an IEEE 802.11 MPDU, containing the following fields [104, Section 9.2.3]:

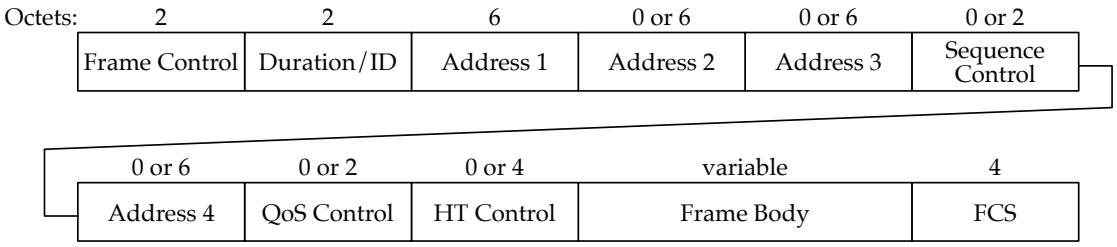


Figure 8: MAC frame format of an IEEE 802.11 MPDU [104, Section 9.2.3]

- The **Frame Control** field indicates, among other things, the frame type, e.g., whether a frame is a data frame or a control frame [104, Section 9.2.4.1].
- The **Duration/ID** field denotes the transmission duration in microseconds or an association identifier, depending on the frame type and settings [104, Section 9.2.4.2]. For non-QoS data frames addressed to a group address, e.g., broadcast frames, the duration field is set to zero [104, Section 9.3.2.1.5]. The WARP prototype presented in Chapter 4 deviates from the IEEE 802.11 standard in this respect by implementing a low-level MAC protocol in the Duration/ID field of broadcast frames, as described in Section 4.2.2. Still, commodity devices can receive such frames flawlessly.
- The **Address** fields designate the receiver and the transmitter of a frame as well as other network entities, depending on the respective frame type [104, Section 9.2.4.3]. In data frames, Address 1 denotes the receiver address (RA), Address 2 denotes the transmitter address (TA) and Address 3 denotes the BSS identifier (BSSID) if the To DS and From DS fields are both set to zero [104, Section 9.3.2, Table 9-30]. The low-level MAC protocol of the WARP prototype presented in Section 4.2.2 works with broadcast data frames, i.e., it sets Address 1 to `FF:FF:FF:FF:FF:FF`. Furthermore, it marks flooding frames by setting Address 3 to `B1:6F:10:0D:FE:ED`. Commodity devices that cannot change the Duration/ID field can inject flooding frames through a special format in Address 2 that encodes protocol parameters.
- The **Sequence Control** field is present in data frames and contains a sequence number and a fragment number, used to identify MAC service data units (MSDUs) and to re-assemble MSDU fragments [104, Section 9.2.4.4]. The prototypes of this work do not employ this field. In particular, the low-level MAC protocol of the WARP prototype presented in Section 4.2.2 defines its own sequence number to meet critical timing constraints. Still, this field may be utilized in future work.
- The **QoS Control** field is present in QoS data frames [104, Section 9.2.4.5]. It is present in the experiments presented in Chapter 5 but not specifically utilized.
- The **HT Control** field is present in QoS data and management frames in the HT and VHT formats [104, Section 9.2.4.6]. It is not specifically utilized in this work.
- The **Frame Body** field contains an MSDU of up to 2304 octets [104, Section 9.2.4.7]. The prototypes and mechanisms presented in this work do not use data frames with a fragmented MSDU or an aggregated MSDU (A-MSDU).
- The **FCS** field comprises a 32-bit cyclic redundancy check (CRC) that is calculated over all MAC header fields and the Frame Body field [104, Section 9.2.4.8].

IEEE 802.2 LLC header			SNAP extension		IPv4 header
DSAP 1 octet	SSAP 1 octet	Control 1 octet	OUI 3 octets	Protocol ID 2 octets	
`0xAA`	`0xAA`	`0x03`	`0x000000`	`0x0800`	

MSDU

Figure 9: IEEE 802.11 MSDU with LLC header and SNAP extension [104, 106]: the LLC header indicates the use of the SNAP extension. The OUI indicates an EtherType [102] value in the Protocol ID field. The encapsulated NET PDU is an IPv4 packet.

An IEEE 802.11 MSDU is a protocol data unit (PDU) of the logical link control (LLC) sublayer [106] to multiplex different network layer (NET) protocols on the data link layer [104, Section 5.1.4]. The LLC can employ the Subnetwork Access Protocol (SNAP) extension to use EtherType [102] values for specifying the encapsulated NET protocol. Figure 9 shows an LLC header with SNAP extension encapsulating an IPv4 packet, as used in this work. To indicate the use of the SNAP extension, the destination service access point (DSAP) and the source service access point (SSAP) addresses of the LLC header are set to `0xAA`. Further, the control field of the LLC header is set to `0x03` to indicate a connectionless and unacknowledged operation of the data link layer [106]. The subsequent SNAP extension consists of an organizationally unique identifier (OUI) field and a Protocol ID field. The OUI field is set to zero to indicate the use of an EtherType value in the Protocol ID field. The EtherType value `0x0800` designates IPv4.

2.4.8 *Short Interframe Space*

The IEEE 802.11 standard uses the term *interframe space (IFS)* to describe the time interval between two consecutive frames and defines different IFS durations that essentially prioritize the medium access by certain frame types over others [104, Section 10.3.2.3]. A short interframe space (SIFS) usually precedes high-priority frame types like ACK and clear to send (CTS) frames [104, Section 10.3.2.3.3], but it is also employed by the prototypes presented in Chapter 4 and in Chapter 5 when generating a retransmission of a received flooding frame. However, the IEEE 802.11 standard uses different definitions of a SIFS that depend on the frequency band in which a PHY operates. In particular, IEEE 802.11a employs the original Non-HT PHY in the 5 GHz band, for which a SIFS is defined as a duration of 16 µs [104, Section 17.4.4]. In contrast, since IEEE 802.11g adopts the Non-HT PHY for use in the 2.4 GHz band, it must ensure backwards compatibility with the IEEE 802.11 DSSS PHY, which defines a SIFS as a duration of only 10 µs. To unify the SIFS duration for different PPDU formats, a so-called *signal extension* is appended to Non-HT PPDUs in the 2.4 GHz band, which essentially is an idle period of 6 µs [104, Section 18.3.2.4]. Similarly, the HT PHY makes this distinction for PPDUs in the 2.4 GHz band and in the 5 GHz band, too [104, Section 19.4.4].

For the sake of consistency, this thesis uses the SIFS definition of 16 µs as specified for the Non-HT PHY [104, Section 17.4.4] in both the 2.4 GHz band and in the 5 GHz band, while the *signal extension* is not applied for experiments in the 2.4 GHz band.

Part II

CONCURRENT COOPERATIVE TRANSMISSION WITH COHERENT INTERFERENCE

This part presents the exploration of CCT with coherent interference from multiple uncoupled senders through synchronization in time and frequency. Chapter 3 presents a study of the feasibility of CCT with IEEE 802.11 PHYs, comprising mathematical analysis, simulations as well as testbed experiments. Then, two prototypes are presented that enable CCT through synchronization in time and frequency. First, Chapter 4 presents an FPGA-based design for the WARP v3 SDR with a low-level MAC protocol, synchronizing through individual frame receptions. Second, Chapter 5 presents a prototype created through firmware modifications of the Asus RT-AC86U wireless router that demonstrates the practical feasibility of CCT also with commercial hardware.

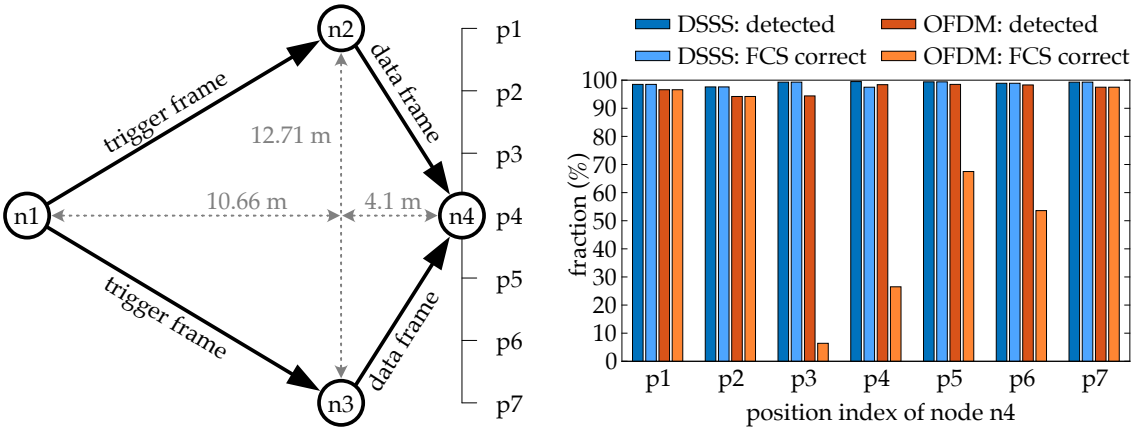
3 ANALYSIS

In a concurrent cooperative transmission (CCT), a receiver sees a signal that results from the superposition of multiple signals from different transmitters sending at the same time. A CCT signal arriving at a receiver ideally resembles a transmission from a single node passing through a multi-path channel, so that the receiver can process the signal as usual. However, the appearance of a CCT signal, as seen by a receiver, generally depends on various factors that can be attributed to the transmitters of the CCT, to the environment through which their respective signals pass as well as to the data frame itself. In the latter category, the employed physical layer (PHY) plays a central role in how the received signal is processed and hence in its decodability. Glossy [73] demonstrates that CCT is feasible with the IEEE 802.15.4 PHY [107], which offers a data rate of 250 kbit/s with direct-sequence spread spectrum (DSSS) technology [93]. To increase the throughput and to reduce the latency further in comparison to Glossy, the technical goal of this thesis is to enable CCT for broadband communications with orthogonal frequency-division multiplexing (OFDM) [243]. However, as OFDM is fundamentally different from DSSS, Glossy's requirement to enable CCT through time synchronization better than 0.5 µs between concurrent transmitters [73] (see Section 2.1.5) may not be applicable to OFDM. Thus, it must be understood how the superposition of multiple OFDM signals in a CCT is conditioned and how the various factors associated with the senders, the environment and the data frame respectively affect the decodability of such a superimposed signal. Accordingly, the following analysis elicits the requirements to facilitate CCT for DSSS and OFDM PHYs in general and for the IEEE 802.11 PHYs [104] (see Section 2.4) in particular.

3.1 Experimental Study with Time Synchronization

Accurate time synchronization is a necessary condition to make identical signals from different senders superimpose without intersymbol interference (ISI) in a CCT, as it aligns the boundaries of the corresponding PHY symbols at the beginning of a frame. Glossy [73] demonstrates that time synchronization better than 0.5 µs between concurrent transmitters is a sufficient condition to obtain decodable signals with the DSSS-based IEEE 802.15.4 PHY [93, 107] (see Section 2.1.5). This section presents an experimental study on the feasibility of CCT through time synchronization of concurrent transmitters with both the IEEE 802.11 DSSS PHY [104, Section 15.3] and the OFDM-based IEEE 802.11 Non-HT PHY [104, Section 17.3], the latter of which is also referred to as the IEEE 802.11 OFDM PHY (see Section 2.4). The goal of this experimental study is to explore the potential to enable CCT through time synchronization also for the IEEE 802.11 PHYs.

Figure 10a shows four nodes `n1`, `n2`, `n3` and `n4` in a line-of-sight (LOS) topology, where node `n4` takes seven positions from `p1` through `p7`. Each node is a Linksys WRT54GL router with a single antenna and with a customized firmware. Node `n1` repeatedly sends a trigger frame that makes nodes `n2` and `n3` concurrently send the same data frame a short interframe space (SIFS) after the trigger frame, i.e., after 16 µs (see Section 2.4.8).



(a) Topology of nodes

(b) CCT frame detection and reception rates

Figure 10: Experimental study on the feasibility of CCT through time synchronization with IEEE 802.11 DSSS at 1 Mbit/s and with IEEE 802.11 Non-HT (i.e., OFDM) at 6 Mbit/s

The data frame contains a MAC service data unit (MSDU) of 1024 bytes. Both nodes `n2` and `n3` are set to the same transmit power to ensure that their signal strengths are roughly equal at the receiver `n4` when it has an equal distance to both nodes. Node `n4` classifies the incoming data frames as *detected* and *received* at the seven marked positions, while *detected* means that the legacy SIGNAL field (L-SIG) in the Non-HT PHY header can be decoded (see Section 2.4.2) and *received* means that the frame check sequence (FCS), which is calculated over the medium access control (MAC) header and the frame body of the MAC protocol data unit (MPDU) (see Section 2.4.7), is correct.

The frame detection and reception rates at node `n4` are depicted in Figure 10b for both the IEEE 802.11 DSSS PHY at 1 Mbit/s and for the IEEE 802.11 Non-HT PHY at the lowest modulation and coding scheme (MCS), i.e., 6 Mbit/s (see Table 2). Each sample in the bar plot is calculated over several hundred transmissions. While the detection and reception rates of the DSSS frames are consistently high, the OFDM PHY achieves high detection rates but irregular reception rates that vary with the position of the receiving node `n4`. In particular, the OFDM reception rates are high at the outer positions `p1`, `p2` and `p7`, where the receiver `n4` is much closer to one of the two transmitters than to the other, respectively. In these locations, the signal from the close transmitter is stronger than the signal from the distant transmitter, so that the latter signal might cause rather weak interference and the capture effect can possibly occur (see Section 2.1.2).

In the center positions, the mutual interference between the two signals is rather strong since both transmitters send with the same power and their respective path losses are quite equal. Notably, at position `p3`, most OFDM frames can be detected whereas they cannot be completely decoded without error. On the one hand, the fact that at least a few OFDM symbols at the beginning of most frames are correctly received indicates that the concurrent transmitters `n2` and `n3` are accurately synchronized in time, so that their corresponding OFDM symbols are aligned with each other and ISI is prevented. On the other hand, the fact that the reception of the subsequent OFDM symbols is often inhibited indicates that interference between the two transmitters sets in shortly after the start of frame (SOF). This distortion cannot be due to a time offset (TO) between the two senders as the resulting ISI would affect all PHY protocol data unit (PPDU) symbols.

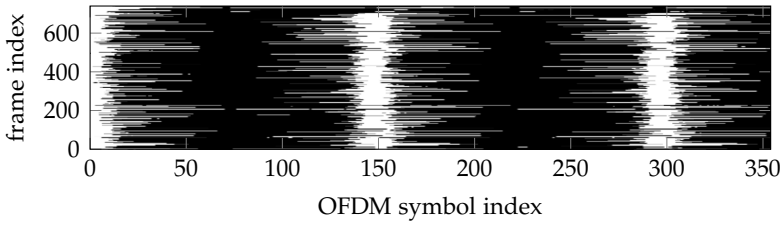


Figure 11: OFDM symbols (white: correct; black: error) of 740 CCT frames at position p3

However, the fact that the first symbols of the CCT frames are correct at position p3, while others are not, can be attributed to effects that make the phase relation between the superimposing signals change over time. In fact, a carrier frequency offset (CFO) between a transmitter and a receiver introduces a phase drift of the received baseband signal during downconversion, i.e., a continuous change of the phase over time [215]. In a CCT, this effect occurs for each transmitter. Thus, a receiver sees the superposition of multiple signals whose phases drift according to their respective transmitters' CFOs, i.e., the transmitters cause incoherent interference. When both signals are received with roughly the same power and the phases of both signals become inverted, a deep fade occurs due to destructive interference. Under this condition, the power of the received signal is very low, rendering the signal undecodable when it is drowned out by noise. Moreover, the phases of both signals continuously rotate relative to each other at the pace of the relative CFO between the two transmitters, which causes a beating effect, i.e., recurring periods of constructive and destructive interference.

The reception performance at position p3 is investigated further with an ALIX system that exposes the demapping of the OFDM symbols of an IEEE 802.11 Non-HT PPDU. Figure 11 shows which OFDM symbols are demapped correctly (white) and which OFDM symbols cause errors (black) for 740 CCT frames. Most frames are correctly received at the beginning but have OFDM symbol errors after a short time, followed by alternating phases of correct reception and OFDM symbol errors. This finding supports the explanation that the symbol errors are introduced by the CFO between the two transmitters n2 and n3. Further, correct reception occurs about every 150 OFDM symbols, which translates to a periodicity of roughly 600 µs, indicating a CFO of about 1660 Hz. This CFO makes the signal-to-interference-plus-noise ratio (SINR) oscillate between two levels that correspond to periods of useful interference and disturbing interference. Note, however, that the useful interference is not necessarily constructive and that the disturbing interference is not necessarily destructive since the phases of the senders are not ensured to be aligned at the start of the CCT. Instead, the periods of useful interference resemble the state of the phase relation between the transmitters at the beginning of the frame, which is rather random, so that equalization with the channel estimates obtained from the legacy long training field (L-LTF) (see Section 2.4.2) relocates the received symbols back to their original in-phase quadrature (IQ) constellation points. Conversely, the disturbing interference displaces the received symbols in the IQ plane, so that the symbols may be demapped incorrectly after equalization.

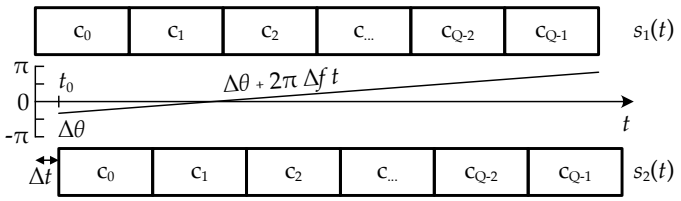


Figure 12: Trains of symbols: with respect to $s_1(t)$, the signal $s_2(t)$ has a TO Δt, an initial PO $\Delta\theta$ and a cumulative phase shift growing over time at the pace of the CFO Δf.

The experimental study shows that time synchronization is not sufficient to reliably enable CCT with an OFDM-based IEEE 802.11 PHY. The decodability of a CCT frame rather depends on several factors, such as the TO and the CFO between the transmitters as well as the frame length, the MCS and noise. In the following, an in-depth study of the combined effects of these factors is provided to foster a deeper understanding of CCT with both the IEEE 802.11 DSSS PHY and the OFDM-based IEEE 802.11 Non-HT PHY.

3.2 CCT System Model

This section introduces a PHY-independent CCT system model to examine the impact of the interference in a CCT with two transmitters. This model is subsequently employed for an analysis of CCT with the IEEE 802.11 DSSS PHY in Section 3.3 and for an analysis of CCT with the OFDM-based IEEE 802.11 Non-HT PHY in Section 3.4. A CCT with two transmitters is the most fundamental case, allowing the carrier frequency offset (CFO), the time offset (TO) and the phase offset (PO) between the concurrent transmitters to be expressed as single quantities. The presented concepts, however, can be generalized to a larger number of transmitters, as shown in Chapter 6. To specifically expose the effect of the interference in a CCT, an ideal channel without any further distortions is assumed. Furthermore, both transmitters are modeled to use the same power, which generates the strongest possible mutual interference and prevents the capture effect (see Section 2.1.2).

The data portion of a frame is composed of Q successive symbols c_q, $0 \leq q \leq Q-1$, as illustrated in Figure 12. These symbols each have a duration T_c, do not overlap, are defined in the interval $[0, T_c[$ and are shifted to their respective positions qT_c within a data frame. In the following, the most robust data rate of each PHY is considered, i.e., 1 Mbit/s for IEEE 802.11 DSSS and 6 Mbit/s for IEEE 802.11 Non-HT (see Table 2). On the one hand, each DSSS symbol has a duration of $T_c = 1\,\mu s$ and encodes a single bit of the PHY service data unit (PSDU) by modulating an 11-chip Barker sequence through differential binary phase-shift keying (DBPSK) [104, Sections 15.4.4.4 and 15.4.4.5] [93]. On the other hand, each OFDM symbol has a duration of $T_c = 4\,\mu s$ and encodes 3 bytes of the PSDU through binary phase-shift keying (BPSK) with coding rate 1/2 by employing 48 data subcarriers in parallel [104, Sections 17.3.2.3 and 17.3.2.4].

As the channel is assumed to be ideal, the absolute channel gains are equal for both transmitters. However, each sender introduces a phase shift during upconversion of the baseband signal to the radio frequency (RF) band, according to the phase of its local carrier frequency oscillator. Such a phase shift is typically random unless the local oscillator is disciplined for a specific purpose like beamforming (see Section 2.1.6.2). Similarly, each transmitter introduces a phase drift, i.e., a continuous phase change, according to its

CFO with respect to another reference, such as a receiver's local oscillator, the specified carrier frequency, or another transmitter. To study the effect of the interference of two superimposing and identical signals, the signal from the first transmitter is taken as a reference for the signal from the second transmitter. With this, the TO, the initial PO and the CFO between the two transmitters are denoted as Δt, $\Delta\theta$ and Δf, respectively:

$$\begin{aligned} s_1(t) &= 2\Re\left\{\sum_{q=0}^{Q-1} c_q(t - qT_c)\,\mathrm{e}^{j2\pi f_c t}\right\} \\ s_2(t) &= 2\Re\left\{\sum_{q=0}^{Q-1} c_q(t - qT_c - \Delta t)\,\mathrm{e}^{j\Delta\theta}\,\mathrm{e}^{j2\pi(f_c+\Delta f)t}\right\} \end{aligned} \tag{1}$$

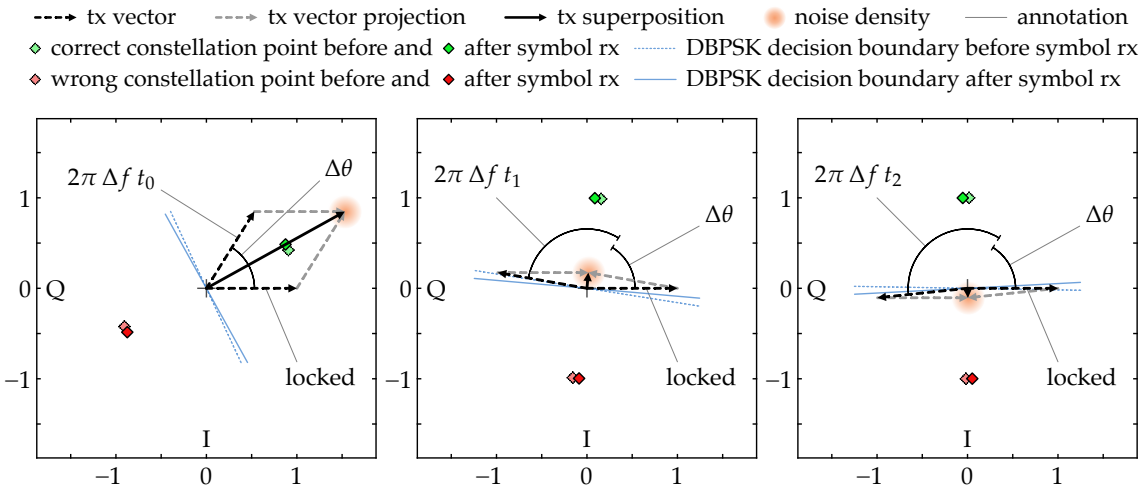
The complex envelope of the superposition of $s_1(t)$ and $s_2(t)$ is:

$$c_\Sigma(t) = \sum_{q=0}^{Q-1} c_q(t - qT_c) + c_q(t - qT_c - \Delta t)\,\mathrm{e}^{j\Delta\theta}\,\mathrm{e}^{j2\pi\Delta f t} \tag{2}$$

The TO Δt between the transmitters may result from imperfect time synchronization, but from a receiver's perspective, it can also be introduced through different propagation delays of the respective signals. In practice, wireless communications can achieve a time synchronization accuracy of about 20 ns [186]. However, in CCT-based network flooding, time synchronization errors might propagate and grow. The PO $\Delta\theta$ is the initial phase relation between the two transmitters. Due to the lack of coherence between uncoupled oscillators, $\Delta\theta$ can be modeled as a random variable uniformly distributed in $]-\pi, \pi]$. The CFO Δf results from manufacturing tolerances and jitter of the oscillators. As an oscillator's phase and frequency variations are typically rather slow (see Section 2.2.2), the PO $\Delta\theta$ and the CFO Δf can be assumed to be invariant for the duration of a frame. Note that a receiver likewise introduces an initial phase shift and a phase drift during downconversion of the received superimposed RF signal, according to its PO and its CFO with respect to the reference carrier frequency f_c, respectively. These phase changes, however, are not relevant to study the effects of the interference between the transmitters.

3.3 Analysis of IEEE 802.11 DSSS

An IEEE 802.11 DSSS PPDU consists of a 144-bit PHY preamble, followed by a 48-bit PHY header and a PSDU (i.e., an MPDU). The PHY preamble comprises a 128-bit SYNC field with scrambled 1s, followed by a 16-bit start of frame delimiter (SFD) [104, Section 15.3]. While clock recovery is achieved best when receiving all bits from the SYNC field, only the last eight bits are actually required to train the self-synchronizing descrambler and to detect the SFD. For this reason, commercial devices are able to detect a valid reception even if only a certain number of bits from the SYNC field are recovered. Therefore, a reception can be regarded as successful if all bits starting from a certain index $q_s \leq 120$ on are correctly received. The segment between the symbols with the indices q_s and $Q-1$ is subsequently referred to as the *sensitive* part of the frame. In the following, both cases with and without a TO between the two transmitters are considered, respectively.



(a) $t_0 = 0\,\mu s$: symbol index q_s (b) $t_1 = 14\,\mu s$: impending deep fade (c) $t_2 = 16\,\mu s$: abating deep fade

Figure 13: CCT interference events with IEEE 802.11 DSSS for $\Delta t = 0\,s$ and $\Delta f = 22.2\,kHz$: all symbols encode bits such that the polarity of correct and wrong constellation points is retained. (a) The sensitive part of the frame starts at the symbol with the index q_s. (b) An imminent deep fade may cause a symbol error of the next symbol due to noise. (c) After the deep fade, a phase jump by π occurs, enforcing a symbol error.

3.3.1 *IEEE 802.11 DSSS without TO*

With a TO $\Delta t = 0\,s$ between both signal components of $c_\Sigma(t)$, the corresponding symbols are exactly aligned with each other, so that no ISI occurs. Still, the amplitude and the phase of $c_\Sigma(t)$ may change over time due to the effects of the PO $\Delta\theta$ and the CFO Δf. A DBPSK receiver, however, is robust to a phase drift in a received signal, as each bit is encoded as a phase change of $\{0, \pi\}$ between successive symbols [104, Section 15.4.4.5]. With this, a DBPSK receiver can track the phase drift introduced by a CCT as long as the signal power is large enough for detection and as long as the effective phase drift of $c_\Sigma(t)$ introduces phase jumps smaller than $\pm\frac{\pi}{2}$ between successive symbols. However, these requirements are violated under the following conditions, as illustrated in Figure 13:

- As both signal components have the same power, they cancel each other when their phases become inverted with respect to each other, i.e., $c_\Sigma(t)$ is subject to a deep fade when $\Delta\theta + 2\pi\,\Delta f\,t$ reaches $\pm\pi$. Under this condition, the received signal contains only noise in practice, potentially causing a symbol error. Figure 13b shows such an event one symbol duration before the interference gets most destructive.
- When $\Delta\theta + 2\pi\,\Delta f\,t$ crosses $\pm\pi$, a phase jump by π occurs in $c_\Sigma(t)$. Even if the symbol affected by the deep fade is received correctly by chance, the phase jump leads to a symbol error of a subsequent symbol, as indicated in Figure 13c.
- The CFO between a transmitter and a receiver generates a phase drift in the received baseband signal. To cause a phase jump greater than $\pm\frac{\pi}{2}$ between two successive symbols of 1 µs duration, the CFO between transmitter and receiver must be at least $\pm 250\,kHz$. The IEEE 802.11 DSSS standard, however, only allows for carrier frequency deviations of up to 25 parts per million (ppm), i.e., up to $\sim \pm 60\,kHz$ in the 2.4 GHz band [104, Section 15.4.5.6], so that this event should not occur.

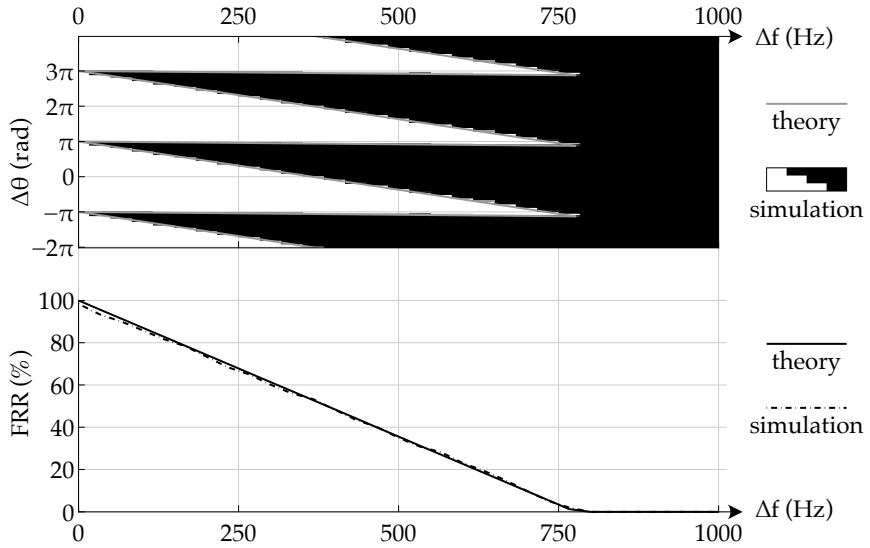


Figure 14: Decodability of a CCT with IEEE 802.11 DSSS for $\Delta t = 0\,\mathrm{s}$ and for a 147-byte PSDU: the top map shows which combinations of CFO and PO between two transmitters can be decoded (white area: decodable; black area: error; gray line: theoretical boundary), whereas the bottom graph shows the FRR after averaging out the PO $\Delta\theta$.

Thus, the relative phase shift between both signal components of $c_\Sigma(t)$ must not exceed $\pm\pi$ in the sensitive part of the frame, as the destructive interference would cause symbol errors due to noise and due to a phase jump by π after the deep fade. Setting $\Delta t = 0\,\mathrm{s}$ in Equation 2 and shifting the temporal reference to the q-th symbol yields:

$$c_\Sigma(t;q) \;=\; c_q(t)\left(1 + \mathrm{e}^{j\,\Delta\theta + j\,2\pi\,\Delta f\,(qT_c + t)}\right) \tag{3}$$

To avoid a phase inversion, two conditions must be met for $\Delta f \geq 0\,\mathrm{Hz}$ and $\forall\, k \in \mathbb{Z}$:

$$\begin{aligned} \Delta\theta + 2\pi\,\Delta f\,q_s\,T_c \;&>\; -\pi + k\cdot 2\pi \\ \Delta\theta + 2\pi\,\Delta f\,Q\,T_c \;&<\; \pi + k\cdot 2\pi \end{aligned} \tag{4}$$

The parameter combinations that satisfy these conditions for $k = 0$ are enclosed between two gray lines in the top map of Figure 14, representing the boundaries for a PSDU size of 147 bytes, i.e., for a PPDU with $Q = 1368$, and for $q_s = 120$. This result is validated through a simulation that generates CCT frames with parameter sweeps of the PO $\Delta\theta$ and the CFO Δf by means of a software encoder/decoder for IEEE 802.11 DSSS. The superimposed CCT signal is fed through an additive white Gaussian noise (AWGN) channel that maintains an excellent signal quality at a signal-to-noise ratio (SNR) of 30 dB. The noisy CCT signal is fed to the software decoder. Successful receptions are depicted as a white region in the top map of Figure 14 and match perfectly with the analytical result enclosed in gray lines. Further, the decodable region is point-symmetric to the origin as a sign inversion of the CFO Δf reverses the inequality relations in Equation 4. The maximum tolerable CFO $\Delta f_{\max}$ is at the intersection of both resulting boundaries. Using an infinitesimally small $\epsilon \in \mathbb{R}$ yields:

$$\Delta f_{\max} \;=\; \frac{1}{T_c(Q - q_s)} - \epsilon \tag{5}$$

However, a CCT frame with $\Delta f = \Delta f_{\max}$ is only decodable for one specific value of the PO $\Delta\theta$, while all other PO values lead to a phase inversion within the frame and thus to a decoding error. As the PO $\Delta\theta$ can be assumed to be uniformly distributed in $]-\pi, \pi]$ in practice, it can be averaged out to get the probability $P_r(\Delta f)$ of correct reception, or the frame reception rate (FRR), as shown in the bottom graph of Figure 14:

$$P_r(\Delta f) = \begin{cases} 1 - (\Delta f / \Delta f_{\max}) \text{ , } 0 \leq \Delta f \leq \Delta f_{\max} \\ 0 \qquad\qquad\qquad\qquad \text{, else} \end{cases} \tag{6}$$

3.3.2 *IEEE 802.11 DSSS with TO*

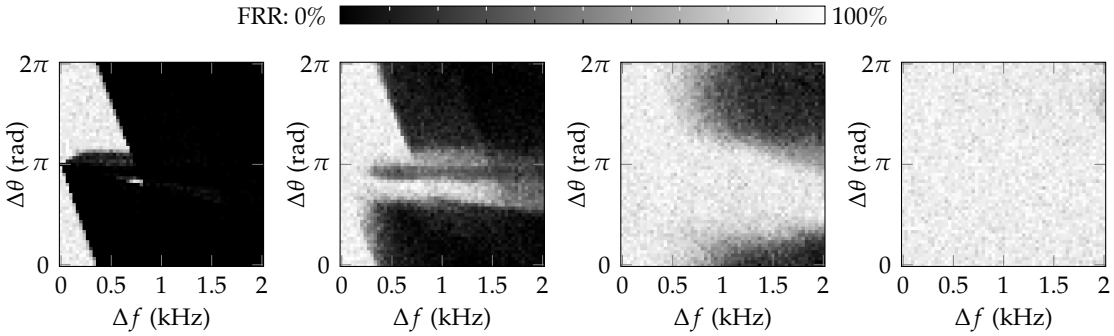
The IEEE 802.11 DSSS PHY applies an 11-chip Barker code as a spreading sequence, the phase of which is modulated through DBPSK for a data rate of 1 Mbit/s or through differential QPSK (DQPSK) for a data rate of 2 Mbit/s [104, Sections 15.4.4.4 and 15.4.4.5]. An IEEE 802.11 DSSS decoder employs a despreading circuit that detects the equidistant occurrences of the Barker code in a received signal through a matched filter. Further, it extracts the phase values at these positions and uses their relative phase shifts for demodulation according to the applied DBPSK or DQPSK modulation scheme.

Since the off-peak auto-correlation coefficients γ_k of a Barker code are by definition minimal, i.e., $|\gamma_k| \leq 1 \;\; \forall\, k \in \{x \,|\, x \in \mathbb{Z},\, 1 \leq x < 11\}$ [93], two IEEE 802.11 DSSS signals with an absolute TO Δt greater than the chip duration $T_p = \frac{1}{11}\,\mu\text{s} = 90.\overline{90}\,\text{ns}$ are almost uncorrelated to each other. In this case, a receiver can lock its despreading circuit to one of the two signal components of $c_\Sigma(t)$, while the other signal component disturbs the reception only marginally. Further, such a reception is mostly unaffected by other factors like the PO $\Delta\theta$ and the CFO Δf, as the time-shifted signals remain highly uncorrelated also under the influence of these factors. However, if the TO Δt is equal to a multiple of the symbol duration $T_c = 1\,\mu\text{s}$, the Barker codes within the two signal components are aligned with each other. In this case, the spreading sequences correlate highly, and since they encode different data, they cause distorting interference. Consequently, an even further increasing TO periodically leads to decoding errors at $\Delta t = k\,T_c$, $k \in \mathbb{Z} \setminus \{0\}$, while the reception is only marginally distorted in between.

3.3.3 *CCT Testbed Experiments with IEEE 802.11 DSSS*

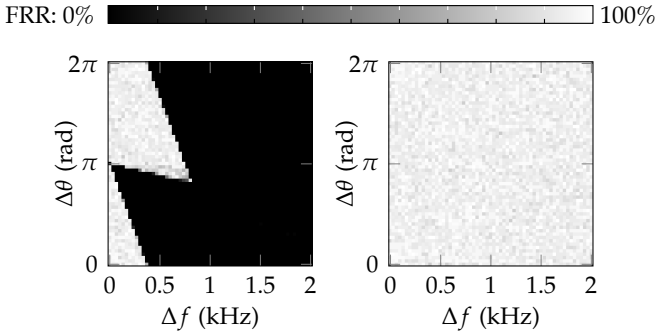
To validate the results of the analysis for the IEEE 802.11 DSSS PHY also in practice, the CCT frame reception capability of commercial devices is assessed in testbed experiments. To this end, IEEE 802.11 DSSS data frames are generated with the C-based software encoder that is also used in the simulation. The generated IQ samples are transferred to a Wireless Open Access Research Platform (WARP) v3 software-defined radio (SDR) [246] via Ethernet by means of WARPLab [250]. The WARP SDR transmits a data frame as a CCT with specific parameter combinations of the TO Δt, the PO $\Delta\theta$ and the CFO Δf. These factors can be precisely tuned in software as a WARP SDR drives its radio interfaces coherently through common RF and sampling clocks (see Section 2.2.3).

The CCT experiments are conducted automatically with complete parameter sweeps across all combinations of the factors Δt, $\Delta\theta$ and Δf within specific parameter ranges



(a) $\Delta t = 0\,\text{ns}$ (b) $\Delta t = 25\,\text{ns}$ (c) $\Delta t = 50\,\text{ns}$ (d) $\Delta t = 75\,\text{ns}$

Figure 15: FRR of AR928X for CCT of a 140-byte PSDU with IEEE 802.11 DSSS: $\Delta f \times \Delta\theta$



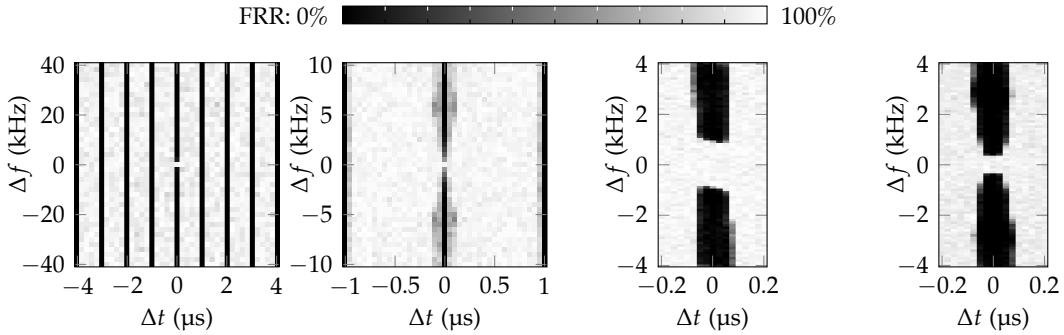
(a) $\Delta t = 0\,\text{ns}$ (b) $\Delta t = 25\,\text{ns}$

Figure 16: FRR of BCM4339 for CCT of a 140-byte PSDU with IEEE 802.11 DSSS: $\Delta f \times \Delta\theta$

and with constant step sizes. To facilitate the calculation of the FRR, 100 frames are transmitted for each parameter combination. The encoder writes the respective parameter values into the MSDU of each frame, besides a logical link control (LLC) header and padding data. With this method, a receiver can extract the parameter values of correctly received frames, so the FRR can be calculated for all covered parameter combinations. The MSDU length is 112 bytes and the PSDU length is 140 bytes (see Section 2.4.7).

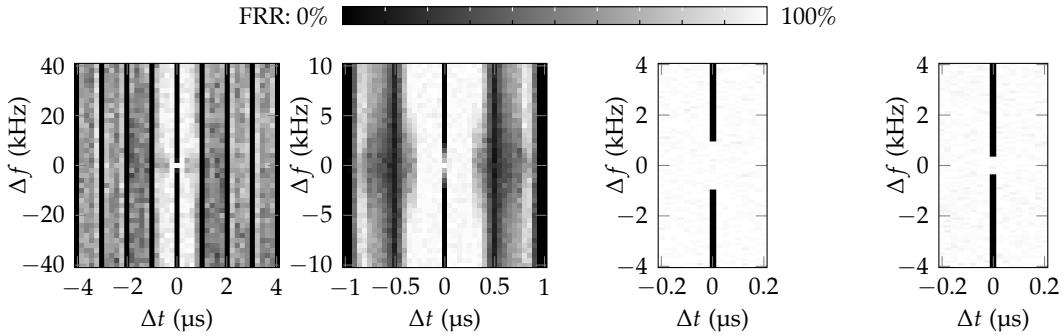
To reduce the chance of collisions with other devices, the CCT frames are transmitted on channel 14 [104, Section 15.4.4.3], the use of which is only allowed with a license in Germany. Additionally, each CCT frame is preceded with a request to send (RTS) frame to block other devices. Both signals of a CCT are mixed and transmitted via a single antenna to ensure equal power levels of both signals at multiple receivers. This causes the highest possible amount of interference, inhibiting the capture effect optimally (see Section 2.1.2). Several receivers are placed in an office environment and are operated in monitor mode to record CCT frame receptions for offline evaluation. The receivers are notebooks with an Atheros AR928X wireless network adapter and Nexus 5 smartphones with a Broadcom BCM4339 chip. The monitor mode is enabled on the Nexus 5 smartphones by means of the firmware patching framework Nexmon [204].

Figure 15 and Figure 16 show the FRRs of the two device types for the combined effects of the CFO Δf and the PO $\Delta\theta$ at different levels of the TO Δt, respectively. For $\Delta t = 0\,\text{s}$, the shapes of the decodable regions in Figure 15a and Figure 16a match with the results



(a) PSDU: 40 b; $\Delta\theta$ rand. (b) PSDU: 40 b; $\Delta\theta$ rand. (c) PSDU: 40 b; $\Delta\theta = 0$ (d) PSDU: 140 b; $\Delta\theta = 0$

Figure 17: FRR of Atheros AR928X for CCT with IEEE 802.11 DSSS: $\Delta t \times \Delta f$



(a) PSDU: 40 b; $\Delta\theta$ rand. (b) PSDU: 40 b; $\Delta\theta$ rand. (c) PSDU: 40 b; $\Delta\theta = 0$ (d) PSDU: 140 b; $\Delta\theta = 0$

Figure 18: FRR of Broadcom BCM4339 for CCT with IEEE 802.11 DSSS: $\Delta t \times \Delta f$

in Figure 14 obtained through analysis and simulations, including the value for $\Delta f_{\max}$. For larger values of Δt, the FRRs increase due to the minimal off-peak auto-correlation of the Barker code that is employed as spreading sequence by the IEEE 802.11 DSSS PHY, as explained in Section 3.3.2. Figure 16b shows that the BCM4339 chip achieves a high FRR for all parameter combinations of $\Delta\theta$ and Δf immediately at $\Delta t = 25\,\text{ns}$, whereas the AR928X adapter reaches this characteristic at $\Delta t = 75\,\text{ns}$, as shown in Figure 15d. The observed differences between AR928X and BCM4339 might be due to different sampling rates and signal processing algorithms. Still, different devices of the same type all exhibit almost identical characteristics in this experiment.

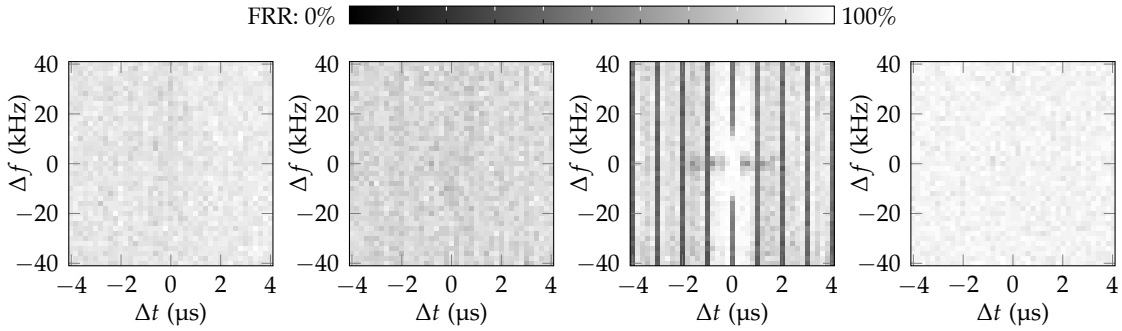
When the TO is a multiple of a DSSS symbol duration, i.e., $\Delta t = k\,T_c$, $k \in \mathbb{Z} \setminus \{0\}$, the spreading sequences of two differing symbols are aligned with each other at a time, leading to distorting interference, as explained in Section 3.3.2. This characteristic is confirmed through another testbed experiment that assesses the FRR over combined parameter sweeps of the TO Δt and the CFO Δf while the PO $\Delta\theta$ is drawn randomly. In Figure 17a and in Figure 18a, the described decoding errors are visible as vertical black lines. Figure 17b and Figure 18b provide a more detailed view of the center and show that the FRR decreases for an increasing absolute CFO at $\Delta t = 0\,\text{s}$, in accordance with the characteristic of the analysis and the simulation shown in the bottom graph of Figure 14. However, the darker regions between the vertical black lines in Figure 18a and

in Figure 18b are not explained by the model and would necessitate an analysis of the specific algorithms running on the BCM4339 chip, which are not disclosed by the vendor.

To validate the dependency of the FRR on the frame length and the CFO Δf at $\Delta t = 0\,\mathrm{s}$, another testbed experiment is conducted in which the PO is fixed to $\Delta\theta = 0$ and two different MSDU sizes of 12 bytes and 112 bytes are employed, corresponding to PSDU sizes of 40 bytes and 140 bytes, respectively. The experiment comprises parameter sweeps of the TO Δt and the CFO Δf with reduced step sizes of 25 ns for Δt and 100 Hz for Δf to increase the resolution in the center of the map. As the PO is fixed to $\Delta\theta = 0$, the second inequality of Equation 4 must be used to determine the condition of a phase inversion. For a PSDU size of 40 bytes, Equation 4 predicts a tolerable CFO of $|\Delta f| \leq 976\,\mathrm{Hz}$. Figure 17c and Figure 18c show the FRR for the AR928X adapter and the BCM4339 chip, respectively, and confirm that a CFO of $|\Delta f| \leq 900\,\mathrm{Hz}$ can be tolerated by both devices at $\Delta t = 0\,\mathrm{s}$, whereas a CFO exceeding $|\Delta f| \geq 1000\,\mathrm{Hz}$ leads to decoding errors. For a PSDU size of 140 bytes, Equation 4 predicts a tolerable CFO of $|\Delta f| \leq 381\,\mathrm{Hz}$. Figure 17d and Figure 18d confirm that both device types tolerate a CFO of $|\Delta f| \leq 300\,\mathrm{Hz}$ but fail to decode frames with a CFO of $|\Delta f| \geq 400\,\mathrm{Hz}$. Also note that the vertical black lines at $\Delta t = 0\,\mathrm{s}$ are much wider for the AR928X adapter than for the BCM4339 chip, which is also reflected in Figure 15 and Figure 16. However, these differences between the two device types are due to implementation details that are not disclosed by the vendors.

In the previous experiments, both CCT signal components are mixed and sent on a single antenna to ensure that their power levels are equal at all receivers. With this, the characteristics of the two different device types can be compared to the results from the analytical model and the simulation. In particular, for $\Delta t = k\,T_c$, $k \in \mathbb{Z} \setminus \{0\}$, the capture effect cannot be exploited since the spreading sequences of the two signal components of $c_\Sigma(t)$ are aligned with each other, encoding different symbols at a time. However, if the two signal components are transmitted with the same power over separate antennas, they propagate along different paths and may arrive at a receiver with slightly different power levels. In this case and for $\Delta t = 0\,\mathrm{s}$, the stronger signal component is not completely canceled by the weaker one when their phases become inverted. Furthermore, the phase jumps of the received signal are smaller than the worst case illustrated in Figure 13. With this, an IEEE 802.11 DSSS decoder may still be able to lock its despreading circuit to the stronger signal component and decode the symbols affected by the interference while tracking the phase of the received signal.

To validate the occurrence of the capture effect for CCTs with IEEE 802.11 DSSS in practical scenarios, another CCT testbed experiment is conducted in which two identical signals are transmitted with the same power via two separate antennas. Figure 19 shows the corresponding FRR maps of four devices for the combined parameter sweeps of Δt and Δf. Figure 19c exhibits vertical gray lines, indicating that this receiver is distorted at $\Delta t = k\,T_c$, $k \in \mathbb{Z}$. The other receivers do not exhibit this pattern, indicating that the two signal components arrive with slightly different power levels, so that these receivers can exploit the capture effect, respectively. The FRR map of another AR928X device looks similar to Figure 19b and is not shown. Likewise, the high reception rates for IEEE 802.11 DSSS in Figure 10b can also be attributed to the capture effect.



(a) AR928X #1 (b) AR928X #2 (c) BCM4339 #1 (d) BCM4339 #2

Figure 19: FRR for CCT of a 40-byte PSDU via two antennas with IEEE 802.11 DSSS: $\Delta t \times \Delta f$

3.4 Analysis of IEEE 802.11 Non-HT OFDM

This analysis considers two cases under which a CCT with the OFDM-based IEEE 802.11 Non-HT PHY can be decoded correctly. First, Section 3.4.1 explores the conditions under which no symbol errors occur in a CCT with otherwise excellent channel conditions. In this case, a frame can be decoded without utilizing forward error correction (FEC). Second, Section 3.4.2 investigates the decodability of a CCT when FEC is effective. To this end, the signal processing algorithms of a receiver are analyzed as a function of the CCT parameters in order to identify parameter regions with a moderate amount of symbol errors that can be corrected through FEC by means of the convolutional code of the IEEE 802.11 Non-HT PHY. Finally, Section 3.4.3 validates the analytical results in testbed experiments with WARP SDRs [246] and with commodity IEEE 802.11g devices.

3.4.1 *IEEE 802.11 Non-HT OFDM without Symbol Errors*

An IEEE 802.11 Non-HT frame starts with a preamble consisting of a legacy short training field (L-STF) and a legacy long training field (L-LTF), according to the format described in Section 2.4.2 [104]. The L-STF and the L-LTF each last 8 µs, so that the total duration of the PHY preamble amounts to the duration of four regular OFDM symbols. Contrary to the analysis of the IEEE DSSS PHY in Section 3.3, this analysis assumes that the *entire* IEEE 802.11 Non-HT PHY preamble is required for a correct frame reception. This is because a receiver employs the L-STF to detect an incoming frame on its RF interface and to perform other tasks like automatic gain control (AGC) (see Section 2.4.3). To model an IEEE 802.11 Non-HT PPDU as a sequence of OFDM symbols of duration T_c, this analysis describes the PHY preamble as four pseudo OFDM symbols with the indices $q = \{0, 1, 2, 3\}$ that prepend the L-SIG and the DATA OFDM symbols (see Section 2.4.2). Starting with the PHY-independent CCT system model from Section 3.2 and changing the temporal reference of Equation 2 to the q-th symbol yields:

$$c_\Sigma(t;q) \;=\; c_q(t) + c_q(t-\Delta t)\, e^{j\,\Delta\theta}\, e^{j\,2\pi\,\Delta f\,(qT_c+t)} \tag{7}$$

Applying a Fourier transform with the first signal component as the reference yields:

$$\mathcal{F}\{c_\Sigma\}(f;q) \;=\; C_\Sigma(f;q) \;=\; C_q(f) + C_q(f-\Delta f)\, e^{j\,\Delta\theta}\, e^{-j\,2\pi\,f\,\Delta t}\, e^{j\,2\pi\,\Delta f\,(qT_c+\Delta t)} \tag{8}$$

As the CFO Δf between both transmitters can be assumed to be very small compared to the subcarrier spacing, i.e., $\Delta f \ll 312.5\,\text{kHz}$, intercarrier interference (ICI) is subsequently neglected for the sake of simplicity: $C_q(f - \Delta f) \simeq C_q(f)$. Furthermore, f is replaced by the discrete frequency of an OFDM subcarrier with index n: $f = \frac{n}{NT_s}$, where T_s is the sample duration and N is the length of the fast Fourier transform (FFT). Note that with an RF channel bandwidth of 20 MHz, many systems operate at a sampling rate of 20 megasamples per second (MS/s), for which the corresponding values are $T_s = 50\,\text{ns}$ and $N = 64$. The spectrum of the received complex envelope can be rewritten as:

$$C_\Sigma(n;q) \simeq C_q(n)\left[1 + \mathrm{e}^{j\Delta\theta}\,\mathrm{e}^{-j2\pi\frac{n}{NT_s}\Delta t}\,\mathrm{e}^{j2\pi\Delta f\,(qT_c+\Delta t)}\right] \tag{9}$$

The spectrum $C_q(n)$ of the q-th OFDM symbol is multiplied by a factor $M(n;q)$ that incorporates the TO Δt, the PO $\Delta\theta$ and the CFO Δf between the two senders of the CCT:

$$M(n;q) = 1 + \mathrm{e}^{j\left(\Delta\theta - 2\pi\frac{n}{NT_s}\Delta t + 2\pi\Delta f\,(qT_c+\Delta t)\right)} \tag{10}$$

To avoid a phase inversion between both senders, which would lead to a deep fade and thus to symbol errors, the following conditions must be met for $\Delta f \geq 0\,\text{Hz}$ and $\forall\, k \in \mathbb{Z}$:

$$\begin{aligned} \Delta\theta - 2\pi\frac{n}{NT_s}\Delta t + 2\pi\Delta f\,\Delta t &> -\pi + k\cdot 2\pi \\ \Delta\theta - 2\pi\frac{n}{NT_s}\Delta t + 2\pi\Delta f\,(Q\,T_c + \Delta t) &< +\pi + k\cdot 2\pi \end{aligned} \tag{11}$$

Note that only the second summand of the exponent in Equation 10 depends on the subcarrier index n. This term describes a phase shift that is proportional to the subcarrier index and to the TO Δt. Thus, a TO $\Delta t \neq 0$ introduces a phase slope across all subcarriers in the second signal component of $C_\Sigma(n;q)$. As the above conditions must be met for all subcarriers simultaneously, the region of correct decoding is determined for $k = 0$ as:

$$\begin{aligned} -\pi + 2\pi\frac{n_\text{max}}{NT_s}\Delta t - 2\pi\Delta f\,\Delta t &< \Delta\theta \\ \Delta\theta &< \pi + 2\pi\frac{n_\text{min}}{NT_s}\Delta t - 2\pi\Delta f\,(Q\,T_c + \Delta t) \end{aligned} \tag{12}$$

The subcarrier indices n_max and n_min are the outermost OFDM subcarriers and take values $n_\text{max} = -n_\text{min} = 26$ for the IEEE 802.11 Non-HT PHY [104, Section 17.3]. By eliminating the PO $\Delta\theta$ and by using an infinitesimally small $\epsilon \in \mathbb{R}$, the maximum tolerable CFO is:

$$\Delta f_\text{max} = \frac{1 - \frac{n_\text{max}-n_\text{min}}{NT_s}\Delta t}{Q\,T_c} - \epsilon \tag{13}$$

As $\Delta f \geq 0\,\text{Hz}$ holds since Equation 11, the following condition must also be fulfilled:

$$\frac{n_\text{max} - n_\text{min}}{NT_s}\Delta t < 1 \tag{14}$$

Thus, there is a TO Δt_max that must not be exceeded in order to determine a maximum tolerable CFO Δf_max through Equation 13. In particular, Equation 14 ensures that the phase shift caused by the second summand of the exponent in Equation 10 is contained within $\pm\pi$ for all subcarriers. With this, Equation 14 is independent of the frame length.

With $n_{\max} = -n_{\min} = 26$, $N = 64$ and $T_s = 50\,\text{ns}$, according to the IEEE 802.11 Non-HT PHY with an RF channel bandwidth of 20 MHz at 20 MS/s, the maximum TO $\Delta t_{\max}$ is:

$$\Delta t_{\max} \approx 61.5\,\text{ns} \tag{15}$$

This result reveals that the time synchronization requirement for a CCT with two senders using the IEEE 802.11 Non-HT PHY is quite strict when symbol errors should be avoided. In fact, it allows only for a time synchronization error of ± 1 sample at a sampling rate of 20 MS/s. Note that this limitation solely results from the requirement that the phase difference between the outermost subcarriers should not exceed 2π for the second sender.

While a CCT without symbol errors is theoretically still possible for $\Delta t > \Delta t_{\max}$, it is unlikely to occur in practice. This is because when $\Delta t > \Delta t_{\max}$, the OFDM subcarriers are fanned out into all angular directions in the second signal component of $C_\Sigma(n;q)$, so that there is either at least one subcarrier on which a phase inversion occurs immediately or there are at least two subcarriers in between of which a phase inversion would occur. Thus, when the subcarriers drift due to a CFO, the time span until a deep fade occurs on any of the 52 subcarriers is very small. Still, a receiver may recover a frame through FEC in this case, as subsequently discussed in Section 3.4.2.

In contrast to these considerations on imperfect time synchronization, $\Delta f_{\max}$ reaches its maximum when both senders are perfectly synchronized in time, i.e., at $\Delta t = 0\,\text{s}$:

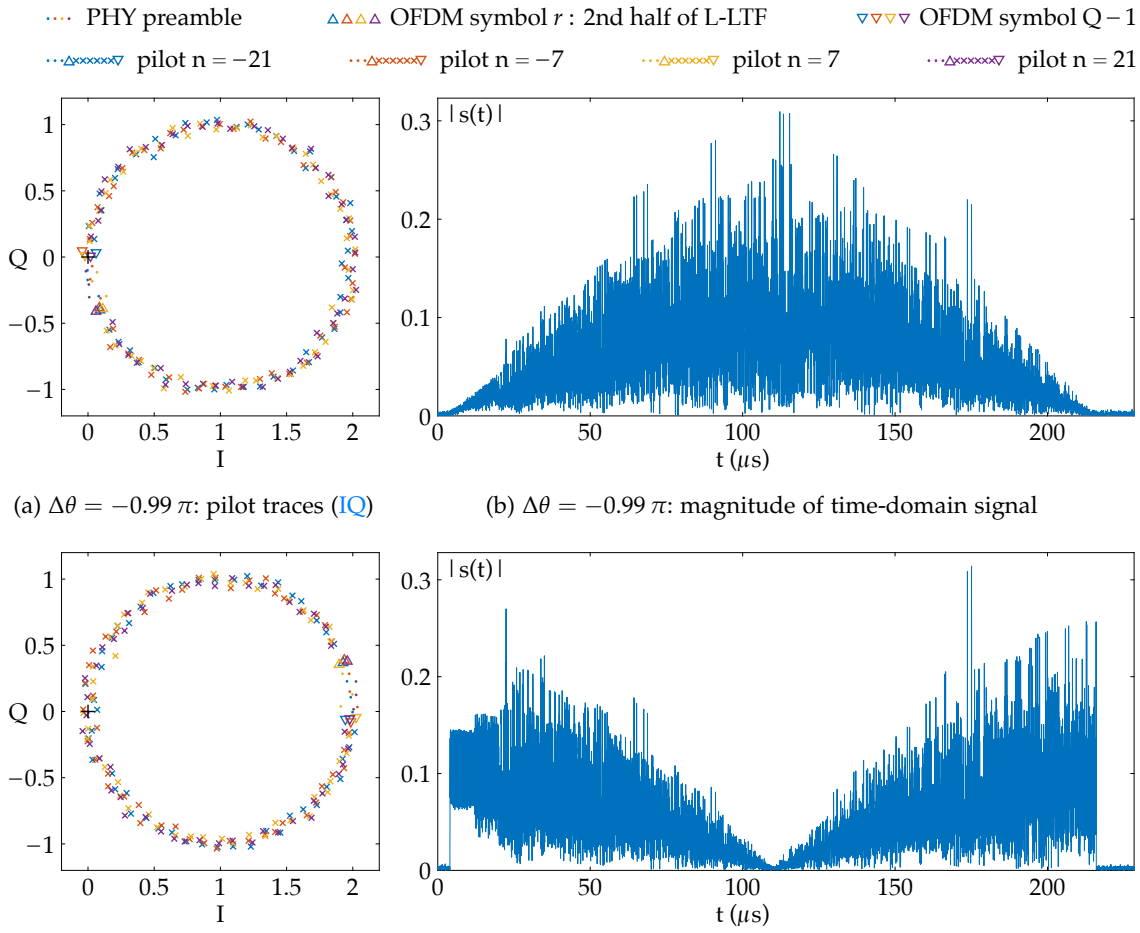
$$\Delta f_{\max}\Big|_{\Delta t = 0\,\text{s}} = \frac{1}{Q\,T_c} - \epsilon \tag{16}$$

Note that this result is similar to the result of the analysis of the IEEE 802.11 DSSS PHY in Equation 5 with the difference that the parameter q_s is not included in Equation 16 since all OFDM symbols are considered to be required for a correct frame reception.

Furthermore, when $\Delta f = \Delta f_{\max}$ and $\Delta t = 0\,\text{s}$, the conditions in Equation 12 hold only when $\Delta\theta$ is within an infinitesimally small interval at $\Delta\theta = -\pi + \epsilon$. This interval of valid values for $\Delta\theta$ grows towards $+\pi$ for smaller values of Δf. Note that a similar relation also exists in the analysis of the IEEE 802.11 DSSS PHY presented in Section 3.3. For $\Delta t = 0\,\text{s}$, the phases of all subcarriers are aligned with each other, so that a deep fade during a phase inversion between both senders affects all subcarriers simultaneously. Given that many symbol errors occur within an affected OFDM symbol in such an event, it is very unlikely that the FEC of a receiver can recover the frame without bit errors. Taking these considerations into account and under the assumption that the PO $\Delta\theta$ is uniformly distributed in $]-\pi, \pi]$ in practice, the probability $P_r(\Delta f)$ of a correct frame reception, i.e., the expected FRR, is in analogy to Equation 6:

$$P_r(\Delta f)\Big|_{\Delta t = 0\,\text{s}} = \begin{cases} 1 - (\Delta f / \Delta f_{\max})\,, & 0 \le \Delta f \le \Delta f_{\max} \\ 0 & \text{, else} \end{cases} \tag{17}$$

The probability of correct frame reception $P_r(\Delta f)$ reaches 100 % in case of both perfect time and perfect frequency synchronization. Still, this result is obtained from an idealized model that does not take noise into account. In practice, a CCT signal with $\Delta t = 0\,\text{s}$, $\Delta f = 0\,\text{Hz}$ and a little noise would attain a low SNR for values of $\Delta\theta$ close to $\pm\pi$, effectively reducing the average FRR. Note that the phase of the second transmitter can theoretically be aligned to the phase of the first transmitter, like in beamforming systems.



(a) $\Delta\theta = -0.99\,\pi$: pilot traces (IQ)

(b) $\Delta\theta = -0.99\,\pi$: magnitude of time-domain signal

(c) $\Delta\theta = 0$: pilot traces (IQ)

(d) $\Delta\theta = 0$: magnitude of time-domain signal

Figure 20: IEEE 802.11 Non-HT CCTs with $Q = 53$, $\Delta t = 0\,\text{s}$, $\Delta f = \lfloor \Delta f_{\text{max}} \rfloor$, AWGN at 30 dB SNR

Still, such a phase synchronization causes a significant amount of signaling overhead in practice (see Section 2.1.6.2) and is thus considered infeasible in the context of this work. Furthermore, the two signal components would typically arrive with slightly different power levels in practice, so that the interference caused through a phase inversion would not be completely destructive, which increases the chance of a correct frame reception.

Figure 20 illustrates by means of a Matlab simulation how the interference in a CCT with two senders affects an IEEE 802.11 Non-HT frame reception under the condition of perfect time synchronization, i.e., for $\Delta t = 0\,\text{s}$. The Matlab simulation generates an IEEE 802.11 Non-HT frame through a custom PHY implementation, applies the CCT parameters Δt, $\Delta\theta$ and Δf to a copy of the generated baseband signal, mixes both signals, applies AWGN at an SNR of 30 dB and feeds the noisy signal to the corresponding IEEE 802.11 Non-HT decoder. In this example, the frame contains an MSDU of 112 bytes, amounting to a PSDU of 140 bytes in accordance with the practical testbed experiments subsequently presented in Section 3.4.3. Further, with MCS 0 (i.e., with a PHY data rate of 6 Mbit/s, see Table 2), the total number of OFDM symbols is $Q = 53$, including the pseudo symbols for the PHY preamble. According to Equation 13, the CFO between both transmitters is set to $\Delta f = \lfloor \Delta f_{\text{max}} \rfloor = 4716\,\text{Hz}$.

In a first simulation example, the PO is set to $\Delta\theta = -0.99\pi$, a value approximately at the beginning of the small interval of valid values for $\Delta\theta$ in accordance with Equation 12. Figure 20a shows a scatter diagram with the symbol traces of the four pilot subcarriers after reverting the polarity of their fixed BPSK-modulated sequence [104, Section 17.3.5.10]. The PHY preamble is represented by four pseudo symbols with a regular OFDM symbol duration $T_c = 4\,\mu s$, shown as three dots and an upward pointing triangle, respectively. Particularly the first two pseudo symbols have to be considered as approximations of the L-STF location, as the L-STF does not employ these pilot subcarriers (see Section 2.4.2). Further, the fourth pseudo symbol corresponds to the second half of the L-LTF and is marked with an upward pointing triangle for each pilot subcarrier since it is considered as the reference for the phase tracking mechanism and for channel estimation in the following analysis in Section 3.4.2. The subsequent SIGNAL and DATA OFDM symbols are shown as crosses, while the last OFDM symbol is shown as a downward pointing triangle. The symbol traces of the pilot subcarriers start directly below the origin $(0 + 0j)$, i.e., the interference is destructive. They drift counterclockwise around the point $(1 + 0j)$, which corresponds to the contribution of the first signal component. In the middle of the frame, the pilot symbols are located around the point $(2 + 0j)$, corresponding to constructive interference, i.e., the phases of both transmitters are aligned. Finally, all four symbol traces end directly above the origin $(0 + 0j)$ without crossing it.

Figure 20b shows the waveform of the magnitude of the corresponding CCT frame. The signal power is apparently very low at the beginning and at the end of the frame, whereas it reaches its maximum in the middle of the frame. These signal power variations are due to the alternating effects of destructive and constructive interference. In the shown example, all OFDM symbols are correctly decoded except for the last one, which is subject to symbol errors on 14 out of 48 data subcarriers. Still, simulation runs with the same parameter settings without AWGN can be decoded without symbol errors. Furthermore, CCT frames with AWGN at 30 dB SNR can be consistently received without symbol errors when the CFO Δf is slightly smaller and when the PO $\Delta\theta$ is slightly larger. These findings indicate that the stated values of Δf and $\Delta\theta$ are indeed the edge case for the decodability of a CCT frame with the length $Q = 53$.

In a second simulation example, the PO is set to $\Delta\theta = 0$, which is well outside of the interval of valid values for $\Delta\theta$. Figure 20c shows that the symbol traces of the four pilot subcarriers start directly at the point $(2 + 0j)$, where the interference is most constructive. Then, they drift counterclockwise around the point $(1 + 0j)$, as in the first example. However, in this example, the symbol traces pass through the origin $(0 + 0j)$, where the frame becomes subject to a deep fade due to destructive interference. Finally, the symbol traces end directly below the point $(2 + 0j)$. In this example, all OFDM symbols are received without symbol errors except for the OFDM symbol with index $q = 26$, which has 19 symbol errors due to the deep fade, leading to bit errors and frame loss. Note that the concentration of symbol errors in a single OFDM symbol can be attributed to the exact time synchronization since the phases of all subcarriers are aligned with each other in accordance with the second summand of the exponent in Equation 10. This relation is also reflected in Figure 20d as a very low signal magnitude in the middle of the frame.

3.4.2 *IEEE 802.11 Non-HT OFDM with FEC*

Since the IEEE 802.11 Non-HT PHY employs a convolutional code for forward error correction (FEC), a receiver can potentially decode a corresponding CCT frame without bit errors also in CCT parameter regions that cause a moderate amount of symbol errors. In particular for $\Delta t > \Delta t_{\max}$ (see Equation 15), the 52 OFDM subcarriers of the second signal component of $C_{\Sigma}(n;q)$ are fanned out into all angular directions. While a CFO still causes constructive and destructive interference in alternation in this case, a deep fade occurs just on one or a few subcarriers at a time, in contrast to the case of perfect time synchronization discussed in Section 3.4.1. Thus, the deep fade events are spread across several OFDM symbols, opening up the chance for correct decoding through FEC.

The capability of a receiver to cope with symbol errors through FEC depends on two factors. First, a lower coding rate improves the robustness of a receiver against symbol errors, at the expense of a lower PHY data rate. The IEEE 802.11 Non-HT PHY employs the coding rates 1/2, 2/3 and 3/4 for the different MCS indices, as listed in Table 2. Second, the employed decoding algorithm and its configuration also have an influence on the error correction capability. While the IEEE 802.11 standard recommends the Viterbi algorithm for convolutional decoding [104, Section 17.3.5.6], this recommendation is not binding, so that commercial products may potentially implement also other algorithms. Particularly suboptimal decoding algorithms may achieve a better decoding complexity while they do not always provide the maximum-likelihood decision [158, Section 12.8]. Further, the Viterbi algorithm itself allows for variations in terms of hard-decision vs. soft-decision decoding [158, Section 12.3]. Moreover, due to the structure of convolutional coding, it is generally difficult to give an exact figure for the fraction of errors that can be corrected even with a specific decoding algorithm [158, Section 12.5].

Despite these complications, this analysis investigates how the CCT parameters PO $\Delta\theta$, CFO Δf and TO Δt as well as the number of OFDM symbols Q affect the decodability of a CCT frame when FEC is effective. To this end, this analysis identifies the conditions in a CCT that cause symbol errors at a receiver. First, the phase tracking mechanism is analyzed since it alters the phase values of received symbols as a function of the OFDM symbol index. Then, also other mechanisms that affect the phase values of received symbols are taken into account to systematically identify conditions that cause symbol errors throughout a CCT. Subsequently, Section 3.4.3 validates the results of this analysis through practical testbed experiments with commodity IEEE 802.11g devices.

3.4.2.1 *Analysis of the Phase Tracking Mechanism*

This section analyzes the effect of a CCT on the phase tracking mechanism, which is required for the following analysis in Section 3.4.2.2 to study the cause of symbol errors. The IEEE 802.11 Non-HT PHY has four pilot subcarriers at the indices -21, -7, 7 and 21 on which a transmitter sends a fixed BPSK-modulated sequence [104, Section 17.3.5.9]. These pilot subcarriers allow a receiver to track the phase of received OFDM symbols, i.e., to estimate the so-called common phase error (CPE) for each OFDM symbol. The term *common* reflects that the CPE applies equally for all subcarriers of an OFDM symbol [263]. A CPE may occur due to a residual CFO of a received baseband signal, even after CFO estimation and correction by means of the PHY preamble (see Section 2.4.4). A receiver usually corrects the CPE of each OFDM symbol before demapping and decoding.

To estimate the CPE of an OFDM symbol, receiver implementations typically multiply the contained pilot symbols with the complex conjugates of the corresponding channel coefficients from the L-LTF and extract the phase from the sum of these values [150, 263]. With a set of pilot subcarrier indices P and an OFDM symbol index r of the reference OFDM symbol for channel estimation, the estimated CPE of the q-th OFDM symbol is:

$$\hat{\psi}(q;r) \;=\; \arg\left(\sum_{n\in P} C_q(n)\cdot C_r^*(n)\right) \tag{18}$$

This procedure essentially improves the robustness of the CPE estimation against noise, as the contribution of each subcarrier is proportional to the square of its magnitude, unless its magnitude fades over time. Also note that in case of a fading channel, the multiplication of a received symbol $C_q(n)$ with $C_r^*(n)$ does not recover the magnitude of the value originally sent on subcarrier n, contrary to channel equalization methods for data subcarriers, like zero-forcing or minimum mean square error (MMSE) [218].

In the following analysis, the set of pilot subcarrier indices is $P = \{-21, -7, 7, 21\}$ in accordance with the IEEE 802.11 Non-HT PHY. Further, the reference OFDM symbol for channel estimation is assumed at index $r = 3$, pointing to the second half of the L-LTF. For a CCT with an ideal channel and with two senders, the OFDM symbols $C_q(n)$ in Equation 18 have to be replaced by the OFDM symbols $C_\Sigma(n;q)$ from Equation 9. By using the factor $M(n;q)$ from Equation 10 and by considering that the symbols $C_q(n)$ on a pilot subcarrier $n \in P$ are identical $\forall\; q \in \{0, 1, ..., Q-1\}$ after reverting the polarity of the fixed BPSK-modulated sequence [104, Section 17.3.5.10], Equation 18 can be rewritten as:

$$\begin{aligned}
\hat{\psi}(q;r) \;&=\; \arg\left(\sum_{n\in P} C_\Sigma(n;q)\cdot C_\Sigma^*(n;r)\right) \\
&=\; \arg\left(\sum_{n\in P} |C_r(n)|^2\cdot M(n;q)\cdot M^*(n;r)\right) \\
&=\; \arg\left(\sum_{n\in P} M(n;q)\cdot M^*(n;r)\right)
\end{aligned} \tag{19}$$

With this, the estimated CPE $\hat{\psi}(q;r)$ is expressed as a function of $M(n;q)$, which contains the CCT parameters $\Delta\theta$, Δf and Δt. The product in the sum can be rewritten as:

$$\begin{aligned}
\widetilde{M}(n;q,r) \;&=\; M(n;q)\cdot M^*(n;r) \\
&= 1 + \mathrm{e}^{j\,2\pi\,\Delta f\,(q-r)\,T_c} + \mathrm{e}^{j\left(\Delta\theta - 2\pi\frac{n}{NT_s}\Delta t + 2\pi\,\Delta f\,(qT_c+\Delta t)\right)} + \mathrm{e}^{-j\left(\Delta\theta - 2\pi\frac{n}{NT_s}\Delta t + 2\pi\,\Delta f\,(rT_c+\Delta t)\right)}
\end{aligned} \tag{20}$$

Adding the contributions of pilot subcarriers with oppositely signed indices yields:

$$\begin{aligned}
\widetilde{M}^{\pm}(n;q,r) \;&=\; \widetilde{M}(n;q,r) + \widetilde{M}(-n;q,r) \\
&= 2 + 2\cdot \mathrm{e}^{j\,2\pi\,\Delta f\,(q-r)\,T_c} \\
&\quad + \mathrm{e}^{j\,(\Delta\theta + 2\pi\,\Delta f\,(qT_c+\Delta t))}\cdot\left[\mathrm{e}^{j\,2\pi\frac{n}{NT_s}\Delta t} + \mathrm{e}^{-j\,2\pi\frac{n}{NT_s}\Delta t}\right] \\
&\quad + \mathrm{e}^{-j\,(\Delta\theta + 2\pi\,\Delta f\,(rT_c+\Delta t))}\cdot\left[\mathrm{e}^{j\,2\pi\frac{n}{NT_s}\Delta t} + \mathrm{e}^{-j\,2\pi\frac{n}{NT_s}\Delta t}\right] \\
&= 2 + 2\cdot \mathrm{e}^{j\,2\pi\,\Delta f\,(q-r)\,T_c} \\
&\quad + 2\cdot\cos\left(2\pi\frac{n}{NT_s}\Delta t\right)\cdot\left[\mathrm{e}^{j\,(\Delta\theta + 2\pi\,\Delta f\,(qT_c+\Delta t))} + \mathrm{e}^{-j\,(\Delta\theta + 2\pi\,\Delta f\,(rT_c+\Delta t))}\right]
\end{aligned} \tag{21}$$

The cumulative phase shift of the second signal component of $C_\Sigma(n;q)$ with respect to its phase at the reference OFDM symbol with index r is subsequently defined as:

$$\beta_r^q = 2\pi\,\Delta f\,(q-r)\,T_c \tag{22}$$

Substituting the OFDM symbol index q in Equation 21 according to Equation 22 yields:

$$\begin{aligned}
\widetilde{M}^{\pm}(n;q,r) &= 2 + 2\cdot \mathrm{e}^{j\,\beta_r^q} + 2\cdot\cos\left(2\pi\,\frac{n}{NT_s}\Delta t\right) \\
&\qquad \cdot\left[\mathrm{e}^{j\left(\Delta\theta+\beta_r^q+2\pi\,\Delta f\,(rT_c+\Delta t)\right)} + \mathrm{e}^{-j\,(\Delta\theta+2\pi\,\Delta f\,(rT_c+\Delta t))}\right] \\
&= 2\cdot\mathrm{e}^{j\,\frac{\beta_r^q}{2}}\cdot\left[\mathrm{e}^{-j\,\frac{\beta_r^q}{2}} + \mathrm{e}^{j\,\frac{\beta_r^q}{2}}\right] + 2\cdot\cos\left(2\pi\,\frac{n}{NT_s}\,\Delta t\right) \\
&\qquad \cdot\mathrm{e}^{j\,\frac{\beta_r^q}{2}}\cdot\left[\mathrm{e}^{j\left(\Delta\theta+\frac{\beta_r^q}{2}+2\pi\,\Delta f\,(rT_c+\Delta t)\right)} + \mathrm{e}^{-j\left(\Delta\theta+\frac{\beta_r^q}{2}+2\pi\,\Delta f\,(rT_c+\Delta t)\right)}\right] \\
&= \mathrm{e}^{j\,\frac{\beta_r^q}{2}}\cdot 4\cdot\left[\cos\left(\frac{\beta_r^q}{2}\right) + \cos\left(2\pi\,\frac{n}{NT_s}\,\Delta t\right)\right. \\
&\qquad \left.\cdot\cos\left(\Delta\theta + \frac{\beta_r^q}{2} + 2\pi\,\Delta f\,(rT_c+\Delta t)\right)\right]
\end{aligned} \tag{23}$$

This result reveals that the metric of the phase tracking mechanism for a pair of subcarriers with oppositely signed indices is a phase shift by $\beta_r^q/2$ scaled by a real-valued factor:

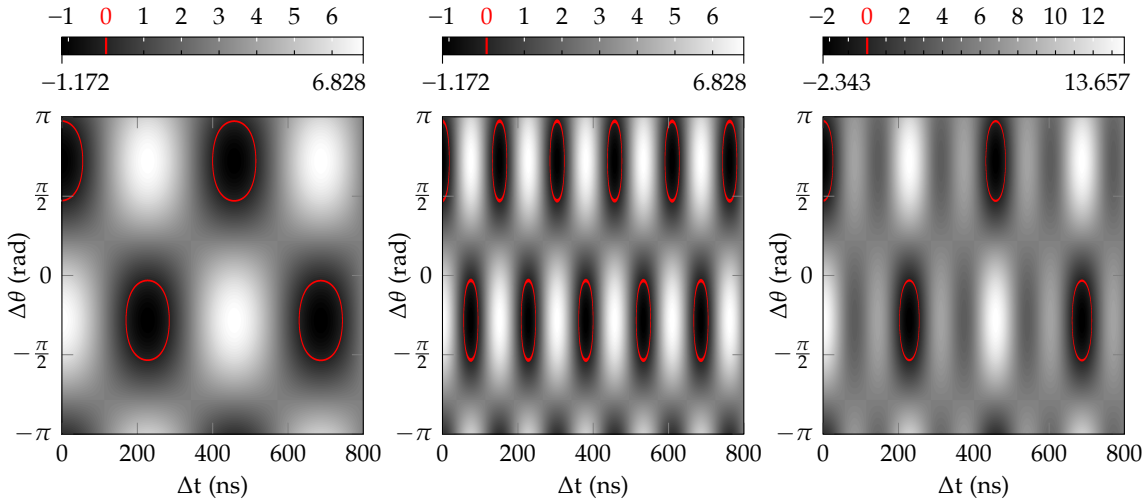
$$\widetilde{M}_{\mathbb{R}}^{\pm}(n;q,r) = 4\cdot\Bigg[\underbrace{\cos\left(\frac{\beta_r^q}{2}\right)}_{m_a(\Delta f,q;r)} + \underbrace{\cos\left(2\pi\frac{n}{NT_s}\Delta t\right)}_{m_b(\Delta t;n)}\cdot\underbrace{\cos\left(\Delta\theta+\frac{\beta_r^q}{2}+2\pi\,\Delta f\,(rT_c+\Delta t)\right)}_{m_c(\Delta\theta,\Delta f,\Delta t,q;r)}\Bigg] \tag{24}$$

Note that $\widetilde{M}_{\mathbb{R}}^{\pm}(n;q,r) \geq 0$ holds for $\beta_r^q = 0$, i.e., for the reference OFDM symbol with the index $q = r$. For larger values of q, the summand $m_a(\Delta f,q;r)$ in Equation 24 decreases from one at $\beta_r^q = 0$ to zero at $\beta_r^q = \pi$. Therefore, the real-valued factor $\widetilde{M}_{\mathbb{R}}^{\pm}(n;q,r)$ has a positive bias for $\beta_r^q \in]0,\pi[$, but it may still become negative for specific CCT parameter combinations due to the second summand $m_b(\Delta t;n)\cdot m_c(\Delta\theta,\Delta f,\Delta t,q;r)$ in the square brackets of Equation 24. Note that the CCT parameters $\Delta\theta$, Δf and Δt are listed as the primary arguments of the three components $m_a(\Delta f,q;r)$, $m_b(\Delta t;n)$ and $m_c(\Delta\theta,\Delta f,\Delta t,q;r)$ to facilitate the subsequent analysis.

For CCT parameter combinations that produce a positive real-valued factor $\widetilde{M}_{\mathbb{R}}^{\pm}(n;q,r)$, the phase of the metric $\widetilde{M}^{\pm}(n;q,r)$ is $\beta_r^q/2$ throughout the entire frame, i.e., the phase of $\widetilde{M}^{\pm}(n;q,r)$ drifts at half of the speed of the CFO Δf between the two transmitters. Contrary, for CCT parameter combinations with a negative $\widetilde{M}_{\mathbb{R}}^{\pm}(n;q,r)$, a deep fade occurs in $\widetilde{M}^{\pm}(n;q,r)$ on an OFDM symbol with an index from r to q, causing a phase jump by π after the deep fade, as represented by the negative sign of $\widetilde{M}_{\mathbb{R}}^{\pm}(n;q,r)$.

Since the IEEE 802.11 Non-HT PHY employs four pilot subcarriers with the indices $P = \{-21,-7,7,21\}$, the overall phase tracking metric given in Equation 19 can be rewritten as a phase shift by $\beta_r^q/2$ scaled by the sum of two real-valued factors:

$$\hat{\psi}(q;r) = \arg\left(\sum_{n\in P}\widetilde{M}(n;q,r)\right) = \arg\left(\mathrm{e}^{j\,\frac{\beta_r^q}{2}}\cdot\left[\widetilde{M}_{\mathbb{R}}^{\pm}(7;q,r) + \widetilde{M}_{\mathbb{R}}^{\pm}(21;q,r)\right]\right) \tag{25}$$



(a) $\widetilde{M}^{\pm}_{\mathbb{R}}(7;q,r)$ (b) $\widetilde{M}^{\pm}_{\mathbb{R}}(21;q,r)$ (c) $\widetilde{M}^{\pm}_{\mathbb{R}}(7;q,r) + \widetilde{M}^{\pm}_{\mathbb{R}}(21;q,r)$

Figure 21: Real-valued factor $\widetilde{M}^{\pm}_{\mathbb{R}}(n;q,r)$ of the phase tracking metric for an IEEE 802.11 Non-HT CCT with a 140-byte PSDU, i.e., $Q = 53$ at MCS 0 (see Table 2); fixed parameters: $q = Q-1$, $r = 3$ and $\Delta f = 1275\,\text{Hz}$ so that $\beta^q_r \approx \frac{\pi}{2}$; parameter sweep: $\Delta t \times \Delta\theta$

Figure 21 shows a slice of $\widetilde{M}^{\pm}_{\mathbb{R}}(n;q,r)$ over a combined parameter sweep of the TO Δt and the PO $\Delta\theta$ for $n = 7$ in Figure 21a, for $n = 21$ in Figure 21b and for the sum of both real-valued factors in Figure 21c. The CCT frame is considered to consist of $Q = 53$ OFDM symbols, including the pseudo symbols for the PHY preamble. This value for Q corresponds to an MSDU of 112 bytes, i.e., a PSDU of 140 bytes, at MCS 0 (see Table 2). Note that these parameters match with the example shown in Figure 20 and with the practical testbed experiments subsequently presented in Section 3.4.3. Further, the CFO is set to $\Delta f = 1275\,\text{Hz}$ so that the phase shift of the last OFDM symbol is $\beta^q_r\big|^{q=Q-1}_{r=3} \approx \frac{\pi}{2}$.

Figure 21a and Figure 21b confirm that $\widetilde{M}^{\pm}_{\mathbb{R}}(7;q,r)$ and $\widetilde{M}^{\pm}_{\mathbb{R}}(21;q,r)$ are positive for large ranges of parameter combinations of the TO Δt and the PO $\Delta\theta$, respectively. However, there are also regions enclosed in red lines for which $\widetilde{M}^{\pm}_{\mathbb{R}}(n;q,r)$ is negative. These red lines are equipotential lines of the function value 0. The repetitive pattern of high and low values is denser in the direction of Δt in Figure 21b than in Figure 21a because of $m_b(\Delta t;n)$ in the second summand in the square brackets in Equation 24.

The extrema of $\widetilde{M}^{\pm}_{\mathbb{R}}(n;q,r)$ can generally be determined through an analysis of its three components in the square brackets in Equation 24. Note that the summand $m_a(\Delta f,q;r)$ is constant when considering a CCT with a specific CFO Δf and a specific OFDM symbol index, e.g., $q = Q-1$ for an OFDM symbol count $Q = 53$. Thus, the extrema of $\widetilde{M}^{\pm}_{\mathbb{R}}(n;q,r)$ depend on $m_b(\Delta t;n)$ and $m_c(\Delta\theta,\Delta f,\Delta t,q;r)$ in the example. Since $m_b(\Delta t;n) \in [-1,1]$ and $m_c(\Delta\theta,\Delta f,\Delta t,q;r) \in [-1,1]$, $\widetilde{M}^{\pm}_{\mathbb{R}}(n;q,r)$ has a maximum where both components are 1 or where they are both -1. Conversely, $\widetilde{M}^{\pm}_{\mathbb{R}}(n;q,r)$ has a minimum where both components have extrema with inverse signs. The TOs Δt at extrema of $m_b(\Delta t;n)$ are:

$$\Delta t^{\max}_{m_b}(n) = \underset{\Delta t\,\in\,[0,800]\,\text{ns}}{\arg\max}\, m_b(\Delta t;n) = \left\{\Delta t \,\middle|\, 2\pi\frac{n}{NT_s}\Delta t = k\cdot 2\pi,\ k\in\mathbb{Z}\right\} \tag{26}$$

$$\Delta t^{\min}_{m_b}(n) = \underset{\Delta t\,\in\,[0,800]\,\text{ns}}{\arg\min}\, m_b(\Delta t;n) = \left\{\Delta t \,\middle|\, 2\pi\frac{n}{NT_s}\Delta t = \pi + k\cdot 2\pi,\ k\in\mathbb{Z}\right\}$$

The corresponding values for the pilot subcarrier pairs with indices $\{-7,7\}$ and $\{-21,21\}$ of the IEEE 802.11 Non-HT PHY are:

$$\begin{aligned} \Delta t_b^{\max}(7) &= \{0\,,\,457.1\}\,\text{ns} \quad \Delta t_b^{\max}(21) = \{0\,,\,152.4\,,\,304.8\,,\,457.1\,,\,609.5\,,\,761.9\}\,\text{ns} \\ \Delta t_b^{\min}(7) &= \{228.6\,,\,685.7\}\,\text{ns} \quad \Delta t_b^{\min}(21) = \{76.2\,,\,228.6\,,\,381.0\,,\,533.3\,,\,685.7\}\,\text{ns} \end{aligned} \tag{27}$$

The extrema of $m_c(\Delta\theta, \Delta f, \Delta t, q; r)$ are located at the following POs $\Delta\theta$:

$$\begin{aligned} \Delta\theta_{m_c}^{\max}(\Delta f, \Delta t, q; r) &= \underset{\Delta\theta \in\,]-\pi,\pi]}{\arg\max}\, m_c(\Delta\theta, \Delta f, \Delta t, q; r) \\ &= \left\{ \Delta\theta \,\middle|\, \Delta\theta + \frac{\beta_r^q}{2} + 2\pi\,\Delta f\,(rT_c + \Delta t) = k \cdot 2\pi,\ k \in \mathbb{Z} \right\} \\ \Delta\theta_{m_c}^{\min}(\Delta f, \Delta t, q; r) &= \underset{\Delta\theta \in\,]-\pi,\pi]}{\arg\min}\, m_c(\Delta\theta, \Delta f, \Delta t, q; r) \\ &= \left\{ \Delta\theta \,\middle|\, \Delta\theta + \frac{\beta_r^q}{2} + 2\pi\,\Delta f\,(rT_c + \Delta t) = \pi + k \cdot 2\pi,\ k \in \mathbb{Z} \right\} \end{aligned} \tag{28}$$

Note that the CFO Δf, the OFDM symbol index q and the reference symbol index r are fixed in the considered example from Figure 21, i.e., $\Delta f = 1275\,\text{Hz}$, $q = Q - 1$ with $Q = 53$ and $r = 3$. Hence, $\Delta\theta_{m_c}^{\max}$ and $\Delta\theta_{m_c}^{\min}$ can be calculated for arbitrary values of the TO Δt under these constraints. Further, as 21 is a multiple of 7, the parameter combinations of Δt and $\Delta\theta$ that cause extrema of $\widetilde{M}_{\mathbb{R}}^{\pm}(7; q, r)$ also cause extrema of $\widetilde{M}_{\mathbb{R}}^{\pm}(21; q, r)$ and of $\widetilde{M}_{\mathbb{R}}^{\pm}(7; q, r) + \widetilde{M}_{\mathbb{R}}^{\pm}(21; q, r)$. Thus, pairs of values of the TO Δt and the PO $\Delta\theta$ can be obtained through Equation 27 and Equation 28, respectively, such that the real-valued factor $\widetilde{M}_{\mathbb{R}}^{\pm}(7; q, r) + \widetilde{M}_{\mathbb{R}}^{\pm}(21; q, r)$ in Equation 25 has an extremum, e.g.:

$$\{\Delta t = 0\,\text{s},\ \Delta\theta = -0.1403 \cdot 2\pi\} \in \underset{\Delta t \in [0,800]\,\text{ns},\, \Delta\theta \in\,]-\pi,\pi]\,|\,\Delta f = 1275\,\text{Hz},\, q = 52,\, r = 3}{\arg\max} \left\{ \widetilde{M}_{\mathbb{R}}^{\pm}(7; q, r) + \widetilde{M}_{\mathbb{R}}^{\pm}(21; q, r) \right\} \tag{29}$$

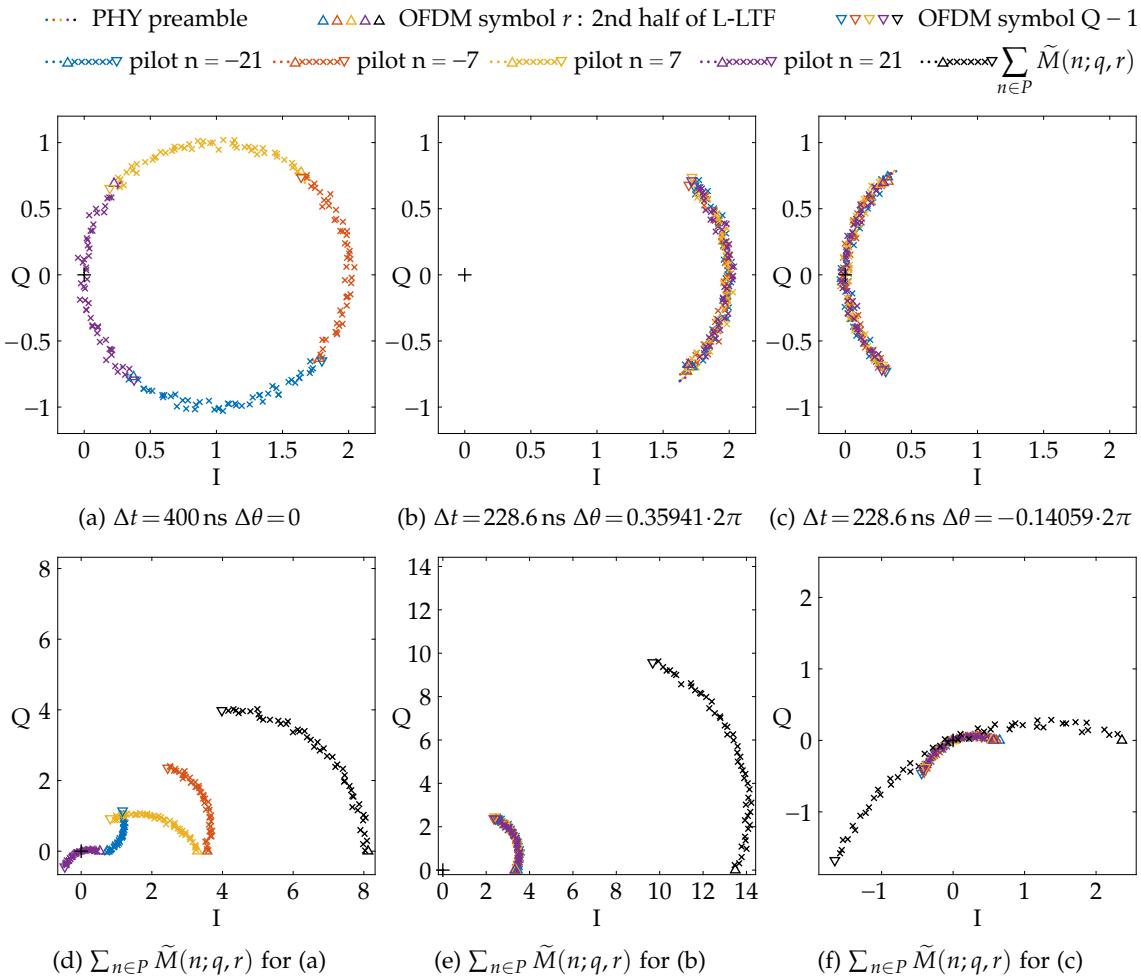
$$\{\Delta t = 228.6\,\text{ns},\ \Delta\theta = 0.35940854 \cdot 2\pi\} \in \underset{\Delta t \in [0,800]\,\text{ns},\, \Delta\theta \in\,]-\pi,\pi]\,|\,\Delta f = 1275\,\text{Hz},\, q = 52,\, r = 3}{\arg\max} \left\{ \widetilde{M}_{\mathbb{R}}^{\pm}(7; q, r) + \widetilde{M}_{\mathbb{R}}^{\pm}(21; q, r) \right\}$$

$$\{\Delta t = 0\,\text{s},\ \Delta\theta = 0.3597 \cdot 2\pi\} \in \underset{\Delta t \in [0,800]\,\text{ns},\, \Delta\theta \in\,]-\pi,\pi]\,|\,\Delta f = 1275\,\text{Hz},\, q = 52,\, r = 3}{\arg\min} \left\{ \widetilde{M}_{\mathbb{R}}^{\pm}(7; q, r) + \widetilde{M}_{\mathbb{R}}^{\pm}(21; q, r) \right\}$$

$$\{\Delta t = 228.6\,\text{ns},\ \Delta\theta = -0.14059147 \cdot 2\pi\} \in \underset{\Delta t \in [0,800]\,\text{ns},\, \Delta\theta \in\,]-\pi,\pi]\,|\,\Delta f = 1275\,\text{Hz},\, q = 52,\, r = 3}{\arg\min} \left\{ \widetilde{M}_{\mathbb{R}}^{\pm}(7; q, r) + \widetilde{M}_{\mathbb{R}}^{\pm}(21; q, r) \right\}$$

Figure 22 illustrates through a Matlab simulation how specific parameter combinations of Δt and $\Delta\theta$ affect the symbol traces on the four pilot subcarriers and the corresponding phase tracking metrics $\widetilde{M}(n; q, r)$ and $\sum_{n \in P} \widetilde{M}(n; q, r)$, respectively. The simulation is executed with the same tools as the simulation presented in Section 3.4.1, but with a CFO of $\Delta f = 1275\,\text{Hz}$ in accordance with the example shown in Figure 21 of this section. Furthermore, the CCT baseband signal is processed through AWGN at 30 dB SNR.

Figure 22a shows the symbol traces on the pilot subcarriers for $\Delta t = 400\,\text{ns}$ and $\Delta\theta = 0$. Note that the TO Δt makes the phases of the subcarriers fan out into various angular directions in the second signal component of $C_\Sigma(n; q)$ due to the second summand of the exponent in Equation 10. Further, the phase of each subcarrier drifts according to the CFO $\Delta f = 1275\,\text{Hz}$, so that the symbol traces run on a circle around the point $(1 + 0j)$.



(a) $\Delta t = 400\,\text{ns}$ $\Delta\theta = 0$ (b) $\Delta t = 228.6\,\text{ns}$ $\Delta\theta = 0.35941 \cdot 2\pi$ (c) $\Delta t = 228.6\,\text{ns}$ $\Delta\theta = -0.14059 \cdot 2\pi$

(d) $\sum_{n\in P} \widetilde{M}(n;q,r)$ for (a) (e) $\sum_{n\in P} \widetilde{M}(n;q,r)$ for (b) (f) $\sum_{n\in P} \widetilde{M}(n;q,r)$ for (c)

Figure 22: Symbol traces of the pilot subcarriers (top) and the corresponding phase tracking metric after Equation 18 (bottom): each figure pair shows an IEEE 802.11 Non-HT CCT with $Q = 53$, $r = 3$, $\Delta f = 1275\,\text{Hz}$, $\beta_r^q\big|_{r=3}^{q=52} \approx \frac{\pi}{2}$, simulated with AWGN at 30 dB SNR.

At position $q = Q - 1$, each subcarrier is shifted around the point $(1 + 0j)$ by $\beta_r^q\big|_{r=3}^{q=52} \approx \frac{\pi}{2}$. Figure 22d shows the complex-valued metric $\widetilde{M}(n;q,r)$ for each pilot subcarrier and the resulting combined phase tracking metric $\sum_{n\in P} \widetilde{M}(n;q,r)$ for $\Delta t = 400\,\text{ns}$ and $\Delta\theta = 0$. Note that $\widetilde{M}(n;q,r)$ drifts within the first quadrant for the pilot subcarriers with the indices $n \in \{-21,-7,7\}$, whereas $\widetilde{M}(n;q,r)$ drifts into the third quadrant for $n = 21$. Figure 21 shows that the parameter combination of Δt and $\Delta\theta$ yields positive real-valued factors $\widetilde{M}^{\pm}_{\mathbb{R}}(7;q,r)$ and $\widetilde{M}^{\pm}_{\mathbb{R}}(21;q,r)$ and that $\widetilde{M}^{\pm}_{\mathbb{R}}(21;q,r) < \widetilde{M}^{\pm}_{\mathbb{R}}(7;q,r)$ holds under the constraints of the example. Figure 22d indeed suggests that $\widetilde{M}(7;q,r) + \widetilde{M}(-7;q,r)$ has a larger amplitude than $\widetilde{M}(21;q,r) + \widetilde{M}(-21;q,r)$. Notably, the last value of the overall phase tracking metric $\sum_{n\in P} \widetilde{M}(n;q,r)$ is $\sim (4.0 + 4.0j)$, which confirms that the CPE estimate of the last OFDM symbol is $\beta_r^q/2\big|_{r=3}^{q=52} \approx \frac{\pi}{4}$, in accordance with Equation 25.

Figure 22b shows the symbol traces on the four pilot subcarriers for $\Delta t = 228.6\,\text{ns}$ and $\Delta\theta = 0.35940854 \cdot 2\pi$, which is a parameter combination for which the real-valued factor $\widetilde{M}^{\pm}_{\mathbb{R}}(7;q,r) + \widetilde{M}^{\pm}_{\mathbb{R}}(21;q,r)$ has a maximum according to Equation 29. Note that the phases of all four pilot subcarriers are aligned with each other, so that the symbol traces overlap. This is because the unwrapped phase difference between two adjacent pilot subcarriers

is 2π according to the second summand of the exponent in Equation 10, i.e., the TO Δt makes the data subcarriers between two adjacent pilot subcarriers fan out into all angular directions in the second signal component of $C_\Sigma(n;q)$, but the pilot subcarriers overlap exactly with each other. Further, the chosen PO $\Delta\theta$ makes the interference between both transmitters most constructive, so that the symbol traces have the largest possible distance from the origin. Figure 22e shows the metric $\widetilde{M}(n;q,r)$ for each pilot subcarrier and the combined phase tracking metric $\sum_{n\in P}\widetilde{M}(n;q,r)$ for the considered parameter combination of Δt and $\Delta\theta$. Since the four pilot subcarriers are aligned with each other, the corresponding instances of $\widetilde{M}(n;q,r)$ look identical, except for noise. The combined phase tracking metric $\sum_{n\in P}\widetilde{M}(n;q,r)$ has in Figure 22e indeed a larger amplitude than in Figure 22d and ends at $\sim(9.7+9.6j)$, which again confirms that the CPE estimate of the last OFDM symbol is $\beta_r^q/2|_{r=3}^{q=52} \approx \frac{\pi}{4}$.

Figure 22c shows another example for the parameter combination $\Delta t = 228.6\,\text{ns}$ and $\Delta\theta = -0.14059147\cdot 2\pi$, for which the real-valued factor $\widetilde{M}_{\mathbb{R}}^{\pm}(7;q,r)+\widetilde{M}_{\mathbb{R}}^{\pm}(21;q,r)$ has a minimum according to Equation 29. The pilot subcarriers are again aligned with each other due to the chosen TO Δt, but the chosen PO $\Delta\theta$ makes the interference between the two transmitters most destructive in this example, so that the symbols have the smallest possible distance from the origin. Notably, the four symbol traces cross the origin at the middle of the frame, i.e., the pilot subcarriers become subject to a deep fade. Figure 22f shows the corresponding traces of $\widetilde{M}(n;q,r)$ and $\sum_{n\in P}\widetilde{M}(n;q,r)$. The metrics $\widetilde{M}(n;q,r)$ of the four pilot subcarriers start in the first quadrant and drift towards the origin, where the deep fade occurs. Then, they drift further into the third quadrant. The combined phase tracking metric $\sum_{n\in P}\widetilde{M}(n;q,r)$ ends at $\sim(-1.7-1.7j)$, confirming that the CPE estimate of the last OFDM symbol is $\beta_r^q/2|_{r=3}^{q=52} \approx \frac{\pi}{4}-\pi$.

The fraction of parameter combinations of Δt and $\Delta\theta$ that produce positive values of $\widetilde{M}_{\mathbb{R}}^{\pm}(7;q,r)+\widetilde{M}_{\mathbb{R}}^{\pm}(21;q,r)$ is 95.4 % in Figure 21c. Note that this fraction is higher for shorter frames in terms of the OFDM symbol count Q and for a lower CFO Δf between the two senders. Thus, an IEEE 802.11 receiver obtains a steady CPE estimate of $\beta_r^q/2$ over all OFDM symbols in most cases. Still, the parameter combinations of Δt and $\Delta\theta$ enclosed by a red line in Figure 21c cause a negative factor $\widetilde{M}_{\mathbb{R}}^{\pm}(7;q,r)+\widetilde{M}_{\mathbb{R}}^{\pm}(21;q,r)$. Since $\widetilde{M}_{\mathbb{R}}^{\pm}(n;q,r)\geq 0$ holds for $q=r$, a deep fade occurs in $\widetilde{M}_{\mathbb{R}}^{\pm}(7;q,r)+\widetilde{M}_{\mathbb{R}}^{\pm}(21;q,r)$ on some OFDM symbol with an index $q\in\{r,\ldots,Q-1\}$ in these cases. If such a deep fade is not handled properly, the phase tracking mechanism inverts the polarity of all data subcarriers when the phase tracking metric is subject to a phase jump by π after the deep fade, probably causing a decoding error as the data subcarriers are not necessarily subject to a deep fade at the same time. Still, a deep fade may occur in the phase tracking metric also in regular transmissions with one sender in scenarios with mobility, due to multi-path propagation effects and due to the Doppler spread [218]. Hence, commodity receiver implementations may handle such an event specifically. First, a receiver may lock its CPE estimate when the magnitude of the phase tracking metric falls below a certain threshold to avoid noise. Second, when the magnitude of the phase tracking metric exceeds the threshold after a deep fade, a receiver can detect a phase jump by comparing the new phase measurement to the previously locked CPE estimate and decide whether to apply the new value. Therefore, commodity devices may possibly deal with a deep fade of the phase tracking metric and retain a phase correction of $\sim\beta_r^q/2$, also for CCT parameter combinations enclosed by a red line in Figure 21c.

3.4.2.2 *Quantification of Data Symbol Errors*

To identify the conditions in a CCT that cause symbol errors on the data subcarriers at a receiver, both the phase tracking mechanism and the channel equalization have to be taken into account, as they alter the phase of received data symbols before modulation demapping or soft-decision convolutional decoding. Equation 25 in Section 3.4.2.1 shows that the estimated CPE of an OFDM symbol with index q is half of the cumulative phase shift β_r^q between two senders of a CCT (see Equation 22), i.e., it grows at half of the rate of the CFO Δf between both senders. While there may be a phase jump by π in the phase tracking metric due to a deep fade in a few cases, it is subsequently assumed that a receiver handles such an event specifically, so that the estimated CPE is $\sim \beta_r^q/2$.

OFDM receivers equalize fading effects by means of channel coefficients that describe the complex-valued channel gains of the corresponding subcarriers (see Section 2.4.5). With the IEEE 802.11 Non-HT PHY, receivers estimate these channel coefficients by means of the L-LTF (see Section 2.4.2). As in Section 3.4.2.1, it is subsequently assumed that the channel estimation takes place at a reference OFDM symbol with index $r = 3$, pointing to the second half of the L-LTF. Since multiple identical signals superimpose in a CCT, the channel coefficients are obtained for a particular state of the senders' interference. For the second signal component of $C_\Sigma(n;q)$, the relative phase of the symbol with the OFDM symbol index r on the subcarrier with the index n is according to Equation 10:

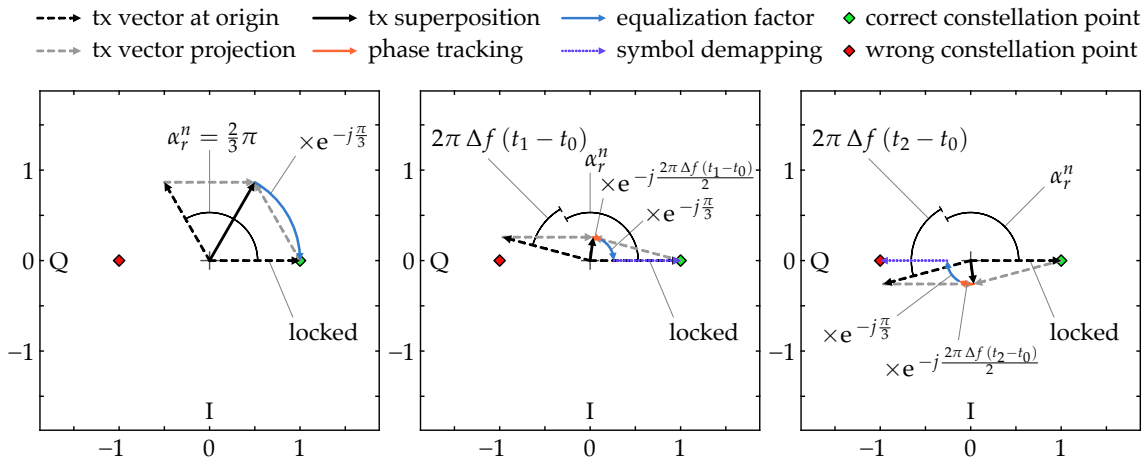
$$\alpha_r^n = \Delta\theta - 2\pi \frac{n}{NT_s} \Delta t + 2\pi \Delta f \left(rT_c + \Delta t\right) \tag{30}$$

Further, the phase of the second signal component drifts at the speed of its CFO Δf with respect to the first signal component of $C_\Sigma(n;q)$, so that its instantaneous phase is:

$$\phi(n;q,r) = \alpha_r^n + \beta_r^q \tag{31}$$

Figure 23 illustrates how the interference of a CCT with two senders affects the received symbols on a particular subcarrier under the influence of the phase tracking mechanism according to Equation 18 and zero-forcing channel equalization [218]. Figure 23a shows the contributions of both signal components at the time $t_0 = rT_c$. According to the system model introduced in Section 3.2, the first signal component is taken as the reference for the second signal component and is marked as locked. The initial phase between both signal components is $\alpha_r^n = \frac{2}{3}\pi$ in this example. Due to the geometry, the superposition has a phase of $\frac{\pi}{3}$ and the same magnitude as the signal components. Since the OFDM symbol with index r is part of the L-LTF, a receiver measures the complex-valued channel gain and uses it subsequently for channel equalization. Figure 23a exemplarily illustrates the process of zero-forcing equalization also for this symbol within the L-LTF.

Figure 23b shows the transmission of another symbol during the same CCT at $t_1 = q_1 T_c$. The phase of the second signal component is shifted by $\beta_r^q\big|_{r=3}^{q=q_1} = 2\pi \Delta f \left(t_1 - t_0\right)$ with regard to its phase at time t_0. With this, the superposition of both signal components is rather destructive. The phase shift is compensated by $\beta_r^q\big|_{r=3}^{q=q_1}/2$ by the phase tracking mechanism. Then, the zero-forcing equalization brings the received symbol to a point on the line between the origin and the correct constellation point. Thus, the received symbol is demapped correctly. Moreover, a deep fade impends on this subcarrier, so that a receiver might not be able to recover some of the subsequent symbols due to noise.



(a) $t_0 = rT_c$: channel estimation (b) $t_1 = q_1 T_c$: impending deep fade (c) $t_2 = q_2 T_c$: abating deep fade

Figure 23: CCT interference events with the IEEE 802.11 Non-HT PHY on a particular subcarrier: all symbols encode the same bit to get a consistent polarity of the constellation points. (a) The channel coefficient is estimated through a reference OFDM symbol in the L-LTF. (b) The superposition of both signals is demapped correctly. (c) After the deep fade, the superposition is subject to a phase jump of π that causes symbol errors.

Figure 23c shows the transmission of another symbol after the deep fade at $t_2 = q_2 T_c$. The interference is still destructive, causing a low magnitude of the superposition. However, the superposition has a phase jump of π after the deep fade. Since the phase tracking metric is obtained from the pilot subcarriers, it is not affected by this deep fade on the data subcarrier. Thus, the phase tracking mechanism shifts the phase of the received symbol by $-\beta_r^q/2\big|_{r=3}^{q=q_2}$. Further, the zero-forcing equalization multiplies the symbol with the factor obtained at time t_0 and brings it to a point that lies on the line between the origin and the wrong constellation point. Thus, the symbol is demapped incorrectly, i.e., a symbol error occurs at a receiver.

While the interference between the two senders gets increasingly constructive for the following symbols, the phase jump in the superimposed signal persists on this subcarrier, so that all symbols after the deep fade are demapped to the wrong constellation point. This effect essentially occurs on all subcarriers that become subject to such a deep fade. As the subcarriers of the second signal component of $C_\Sigma(n;q)$ (see Equation 9) are fanned out into all angular directions when the TO Δt exceeds the value $\Delta t_{\max}$ from Equation 15, there is at least one position in the spectrum of a CCT frame at which adjacent subcarriers successively become subject to a deep fade. Hence, over the duration of a CCT, there is an increasing number of subcarriers on which persistent symbol errors occur.

Figure 24 illustrates the occurrence of symbol errors on individual subcarriers over the duration of a CCT frame in Matlab simulations with different values of the TO Δt. These simulations employ the same tools and settings as the simulations presented in Section 3.4.1 and Section 3.4.2.1, i.e., the CCT frame contains a PSDU with 140 bytes at MCS 0 (see Table 2), resulting in $Q = 53$ OFDM symbols, including the pseudo symbols for the PHY preamble. The CFO is set to $\Delta f = 1275\,\text{Hz}$ so that the cumulative phase shift of the second signal component at the end of the frame is $\beta_r^q\big|_{r=3}^{q=52} \approx \frac{\pi}{2}$. Furthermore, the baseband signal of the CCT frame is processed through AWGN at 30 dB SNR.

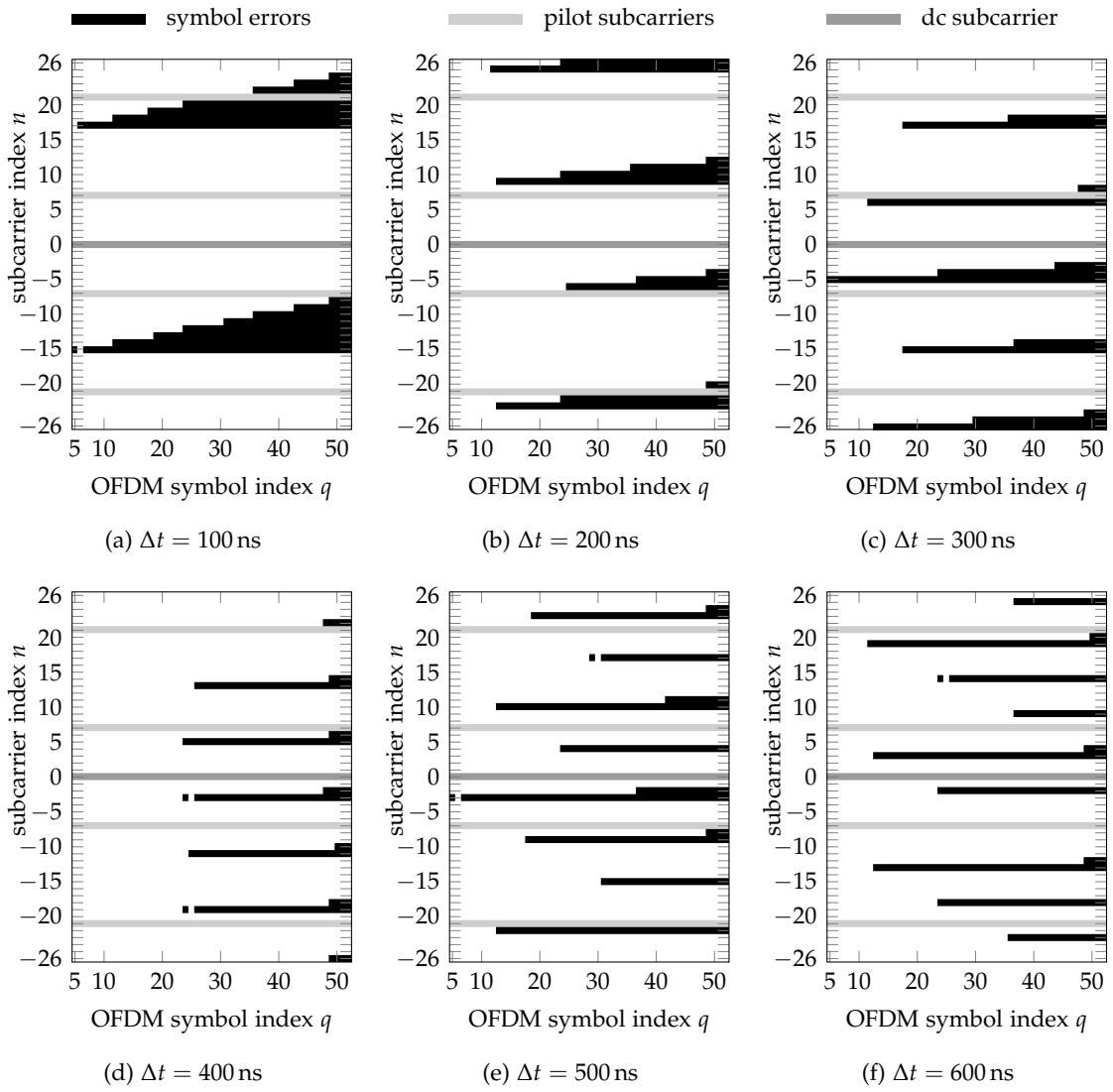


Figure 24: Symbol errors on subcarriers over the OFDM symbol index: each figure shows a single IEEE 802.11 Non-HT CCT frame reception with a 140-byte PSDU, MCS 0 (see Table 2), $Q = 53$, $r = 3$, $\Delta\theta = 0$, $\Delta f = 1275\,\mathrm{Hz}$, $\beta_r^q\big|_{r=3}^{q=52} \approx \frac{\pi}{2}$ and AWGN at 30 dB SNR.

In Figure 24a, the subcarrier at $n = -15$ has a symbol error at $q = 5$, i.e., on the first symbol of the DATA part of the Non-HT PPDU, containing the PSDU (see Section 2.4.2). As the signal quality of the AWGN channel is very good, this symbol error can only be caused by a deep fade due to destructive interference of the CCT. The symbol at $q = 6$ on the same subcarrier is correct, indicating that this subcarrier is dominated by noise due to the low signal power during the deep fade, i.e., a symbol may be received correctly or incorrectly by chance. The subsequent symbols on the same subcarrier, however, are all incorrect, which can be attributed to the phase jump by π after the deep fade. Additionally, the adjacent subcarrier at $n = -14$ starts to have symbol errors at $q = 12$. This is due to the frequency-dependent phase shift of the second signal component of $C_\Sigma(n;q)$ introduced by the second summand in Equation 30. Since $\alpha_r^n\big|_{r=3}^{n=-15} - \alpha_r^n\big|_{r=3}^{n=17} = 2\pi$ holds for $\Delta t = 100\,\mathrm{ns}$, symbol errors start to occur also on the subcarrier at $n = 17$.

With an increasing TO Δt, the phase slope of the subcarriers in the second signal component of $C_\Sigma(n;q)$ becomes steeper, according to the second summand in Equation 30. Thus, the corresponding phase values wrap around more often for an increasing TO Δt. With this, the number of locations in the spectrum in which subcarriers become subject to a deep fade increases. However, the phase difference between adjacent subcarriers increases accordingly. Hence, it also takes longer for adjacent subcarriers to drift into a deep fade at the same CFO Δf. In fact, Figure 24b shows that symbol errors start to occur in four different locations of the spectrum for $\Delta t = 200\,\text{ns}$. Still, the step size until a symbol error occurs on the respective adjacent subcarrier is larger than in Figure 24a. This pattern continues for further increasing values of the TO Δt in Figure 24.

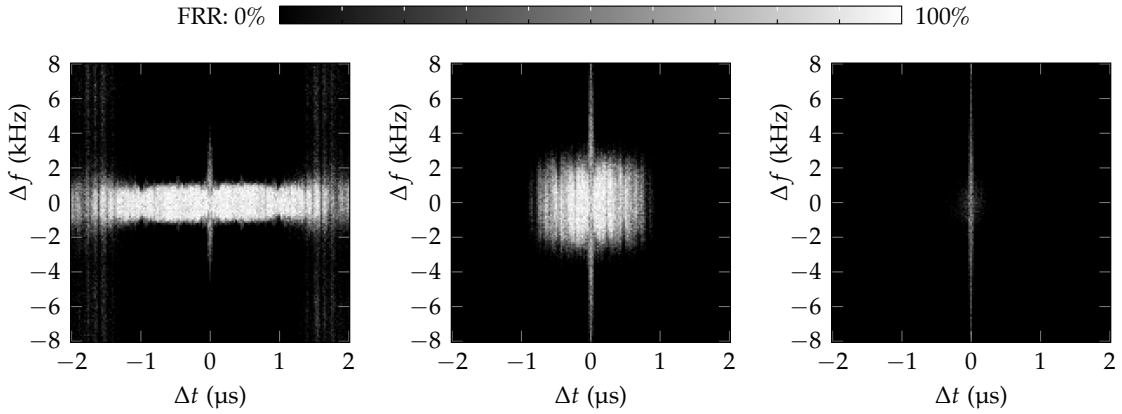
At low values of Δt, the symbol errors are concentrated around one or a few subcarriers. Still, due to deinterleaving, the symbol errors occur in rather random positions for the convolutional decoder, so that the original data may be recovered through FEC. Furthermore, at low values of Δt, the number of deep fades in the spectrum of $C_\Sigma(n;q)$ depends also on the PO $\Delta\theta$, which is randomly distributed in $]-\pi, \pi]$ in practical systems. For $\Delta t = 100\,\text{ns}$, the last OFDM symbol at $q = 52$ has 15 symbol errors in Figure 24a, but it may also be 8 or even less symbol errors for other values of $\Delta\theta$, as confirmed through additional simulation runs. Hence, the decodability of CCT frames may possibly vary across individual frame receptions at low values of Δt.

At higher values of Δt, the number of symbol errors within the last OFDM symbol at $q = 52$ is 12 for each $\Delta t \in \{200, 300, 400, 500, 600\}\,\text{ns}$ in Figure 24. This indicates that the average number of symbol errors per OFDM symbol is independent of the TO Δt. Therefore, the capability of a convolutional decoder to recover the data of a CCT frame can be expected to be quite consistent across different values of $\Delta t < 800\,\text{ns}$, i.e., within the guard interval (GI) of the Non-HT PHY. Also note that the ratio between the number of symbol errors in the last OFDM symbol and the total number of data subcarriers is indeed equal to the ratio between the phase rotation of the second signal component due to its CFO and a full rotation: $\frac{12}{48} = \frac{1}{4} \approx \beta_r^q \big|_{r=3}^{q=52} / 2\pi$.

3.4.3 *CCT Testbed Experiments with IEEE 802.11 Non-HT OFDM*

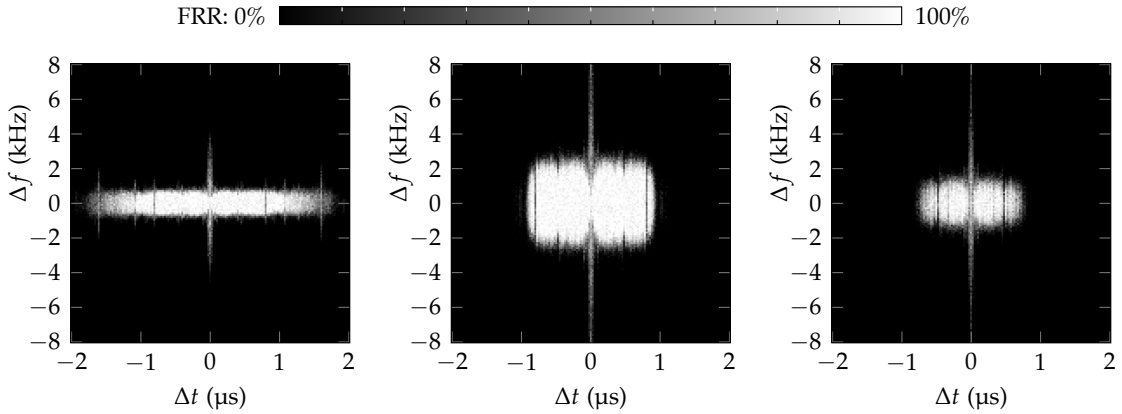
To validate the analysis of a CCT with the IEEE 802.11 Non-HT PHY in practice, testbed experiments with a WARP SDR [246] and commercial IEEE 802.11g devices are conducted. IEEE 802.11 Non-HT data frames are generated with the custom Matlab encoder also used in the simulations shown in Figure 20, Figure 22 and Figure 24. As in Section 3.3.3, the PO $\Delta\theta$, the CFO Δf and the TO Δt between two transmitters are emulated through baseband processing in software of a copy of a generated frame. Both signal components are mixed and transferred to a WARP SDR [246] via WARPLab [250] to transmit the CCT on a single antenna. With this, the power level of both signal components is ensured to be equal at multiple receivers, reconstructing the conditions of the analysis and inhibiting the capture effect through the highest possible amount of interference (see Section 2.1.2).

In accordance with Section 3.3.3, the receivers are notebooks with an Atheros AR928X wireless network adapter and Nexus 5 smartphones with a Broadcom BCM4339 chip. The receivers are placed within the same office as the WARP SDR and are operated in monitor mode to record successful CCT frame receptions. On the Nexus 5 smartphones, the monitor mode is enabled through the firmware patching framework Nexmon [204].



(a) MCS 0: 6 Mbit/s (b) MCS 4: 24 Mbit/s (c) MCS 7: 54 Mbit/s

Figure 25: FRR of Atheros AR928X for CCT with IEEE 802.11 Non-HT OFDM: $\Delta t \times \Delta f$



(a) MCS 0: 6 Mbit/s (b) MCS 4: 24 Mbit/s (c) MCS 7: 54 Mbit/s

Figure 26: FRR of Broadcom BCM4339 for CCT with IEEE 802.11 Non-HT OFDM: $\Delta t \times \Delta f$

Furthermore, to mitigate collisions with other devices, the experiments are conducted on channel 14 [104, Section 15.4.4.3], the use of which requires a license in Germany. In addition, each CCT is prepended with an RTS frame to block other devices.

The CCT experiments are conducted with combined parameter sweeps of the TO Δt and the CFO Δf within specified parameter ranges and with constant step sizes, whereas the PO $\Delta\theta$ is randomly drawn from the interval $]-\pi, \pi]$ to emulate the effect of uncoupled carrier frequency oscillators. As in Section 3.3.3, 100 frames are transmitted for each parameter combination. The CCT parameter values are written into the MSDU so that they can be extracted from correctly received frames. In this way, the FRR can be calculated for each parameter combination. As in the simulations presented in Section 3.4.1 and Section 3.4.2, the MSDU has 112 bytes, so the PSDU has 140 bytes (see Section 2.4.7).

Figure 25 shows the FRR of a notebook with an Atheros AR928X wireless network adapter for MCS 0, 4 and 7 (see Table 2), whereas Figure 26 shows the FRR of a Nexus 5 smartphone with a Broadcom BCM4339 chip exposed to the same CCT frames. Brighter regions again indicate a better CCT frame reception performance.

It can be observed in all plots of Figure 25 and Figure 26 that the successful frame receptions with the highest absolute CFO Δf are found at $\Delta t = 0\,\mathrm{s}$. This finding matches with the analytical result from Equation 13 that entails that the tolerable CFO Δf has a maximum at $\Delta t = 0\,\mathrm{s}$ when symbol errors should be avoided. While a receiver may correct a certain amount of symbol errors through FEC, as discussed in Section 3.4.2, the tolerable CFO for $\Delta t \neq 0\,\mathrm{s}$ is much lower than the CFO $\Delta f_{\max}\big|_{\Delta t=0\,\mathrm{s}}$ in Equation 16. This is because the CFO $\Delta f_{\max}\big|_{\Delta t=0\,\mathrm{s}}$ allows for almost a full rotation of the phase of the second sender, as illustrated in Figure 20. This condition, however, would cause a phase jump of π on almost all subcarriers by the last OFDM symbol of a CCT frame when the TO exceeds $\Delta t_{\max}$ from Equation 15, according to the analysis in Section 3.4.2.2. Therefore, a CFO $\Delta f_{\max}\big|_{\Delta t=0\,\mathrm{s}}$ in combination with a TO $|\Delta t| > \Delta t_{\max}$ would lead to symbol errors on nearly all subcarriers, causing frame loss. Also note that these characteristics are in contrast to the IEEE 802.11 DSSS PHY, which is most susceptible to a CFO Δf at $\Delta t = 0\,\mathrm{s}$, as shown in Figure 17 and Figure 18.

The maximum tolerable CFO $\Delta f_{\max}\big|_{\Delta t=0\,\mathrm{s}}$ is well predicted by means of Equation 16. With a PSDU of 140 bytes at MCS 0, a frame consists of $Q = 53$ OFDM symbols, including the four pseudo OFDM symbols for the PHY preamble, as introduced in Section 3.4.1. With this, Equation 16 yields $\Delta f_{\max}\big|_{\Delta t=0\,\mathrm{s}} \approx 4716\,\mathrm{Hz}$. In Figure 25a, the highest observed CFO at $\Delta t = 0\,\mathrm{s}$ is $\Delta f = 4400\,\mathrm{Hz}$ with an FRR of 5 %. Further, in Figure 26a, the highest observed CFO at $\Delta t = 0\,\mathrm{s}$ is $\Delta f = 4300\,\mathrm{Hz}$ with an FRR of 2 %. Similarly, the FRR is 5 % at $\Delta t = 0\,\mathrm{s}$ and $\Delta f = -4500\,\mathrm{Hz}$ in both Figure 25a and Figure 26a. Note that these data points are barely visible on the black background due to the low FRR values. Also note that a CCT with a CFO near $\Delta f_{\max}\big|_{\Delta t=0\,\mathrm{s}}$ can only be received when the PO $\Delta\theta$ is close to $-\pi + \epsilon$, as illustrated in Figure 20. Since the PO $\Delta\theta$ is randomly drawn from the interval $]-\pi, \pi]$ in this experiment, most of the 100 CCT frames of each parameter combination do not meet this criterion. Further, the CCT frames are subject to noise and undergo physical channel effects. Hence, it is likely that only frame receptions with a slightly smaller CFO than the theoretical maximum $\Delta f_{\max}\big|_{\Delta t=0\,\mathrm{s}}$ are observed.

For TO values exceeding $\Delta t_{\max}$ from Equation 15, i.e., $|\Delta t| > \Delta t_{\max}$, the maximum observed CFO $\Delta f_{\max}\big|_{|\Delta t|>\Delta t_{\max}}$ with successful frame receptions is quite consistent for wide ranges of the TO Δt, which confirms the analysis in Section 3.4.2.2. In particular, $\Delta f_{\max}\big|_{|\Delta t|>\Delta t_{\max}} \simeq 1200\,\mathrm{Hz}$ holds for most values of Δt in Figure 25a. Note that this CFO is close to the value of 1275 Hz employed in the examples in Figure 22 and Figure 24, yielding a phase shift of $\beta_r^q\big|_{r=3}^{q=52} \approx \frac{\pi}{2}$ of the second signal component at the frame end. The ratio of the CFO $\Delta f_{\max}\big|_{|\Delta t|>\Delta t_{\max}}$ to the CFO that would lead to a full phase rotation of the second signal component, as determined through Equation 22, indicates that the Atheros AR928X adapter can correct a symbol error rate (SER) reaching up to $\sim 24\,\%$ by the last OFDM symbol in a CCT with coding rate 1/2, in accordance with Section 3.4.2.2. Further, in Figure 26a, the maximum observed CFO is $\Delta f_{\max}\big|_{|\Delta t|>\Delta t_{\max}} \simeq 900\,\mathrm{Hz}$ for most values of Δt. Thus, the Broadcom BCM4339 chip can correct a SER of up to $\sim 18\,\%$ by the last OFDM symbol in a CCT with coding rate 1/2. The slight variations of the maximum observed CFO $\Delta f_{\max}\big|_{|\Delta t|>\Delta t_{\max}}$ over different TOs Δt might be caused by fading effects of the physical channel. Also, the TO range of correct frame receptions exceeds the GI duration of 800 ns in Figure 25a and in Figure 26a, indicating that the receivers are apparently robust to a certain degree of ISI at MCS 0. In fact, the sum of the TO and the channel delay spread (CDS) must not exceed the GI duration to avoid ISI.

For higher MCS indices, the frame length in terms of the number of OFDM symbols Q is smaller for the same number of bytes in the PSDU. Thus, the CFO Δf in a CCT can generally be higher than for MCS 0 to reach the same SER at the last OFDM symbol. Hence, CCT frames can be decoded at higher values of the CFO Δf than for MCS 0, given that the coding rate of the selected MCS is also 1/2 and that the SNR is sufficiently high for the selected MCS. Figure 25b and Figure 26b show the FRR for the combined parameter sweep of the TO Δt and the CFO Δf at MCS 4, which is 16-QAM with coding rate 1/2 according to Table 2. At MCS 4, the number of OFDM symbols is $Q = 17$ for the 140-byte PSDU. The maximum observed CFO for $|\Delta t| > \Delta t_{\max}$ in Figure 25b is $\Delta f_{\max}\big|_{|\Delta t|>\Delta t_{\max}} \simeq 3400\,\text{Hz}$. According to Section 3.4.2.2, this value indicates that the Atheros AR928X adapter can correct a SER of up to $\sim 18\,\%$ by the last OFDM symbol. This reduction of the CCT decoding capability, in comparison to MCS 0, can be attributed to the additional impact of noise at the higher modulation scheme, which may cause symbol errors besides the phase jumps by π introduced through the CCT. Further, the maximum observed CFO is $\Delta f_{\max}\big|_{|\Delta t|>\Delta t_{\max}} \simeq 3200\,\text{Hz}$ in Figure 26b, indicating that the Broadcom BCM4339 chip can correct a SER of up to $\sim 17\,\%$ by the last OFDM symbol. Besides, the area of correct frame receptions in Figure 26b is brighter than in Figure 25b, indicating that the Broadcom BCM4339 chip has better channel conditions than the Atheros AR928X adapter in this particular experiment.

At MCS 7, the 140-byte PSDU corresponds to a PPDU length of $Q = 11$ OFDM symbols, including the pseudo OFDM symbols for the PHY preamble. This shorter frame duration again implies that an even higher CFO can be accepted to reach a certain phase shift of the second signal component by the end of the CCT frame than with a lower MCS. However, MCS 7 requires a high SNR due to the dense IQ constellations of 64-QAM. Additionally, MCS 7 employs coding rate 3/4, reducing the capability of the convolutional decoder to correct symbol errors, as compared to coding rate 1/2. These two last factors may reduce the tolerable CFO in comparison to the previous examples or inhibit CCT frame receptions completely. In fact, Figure 25c shows that the Atheros AR928X adapter can receive CCT frames only for $\Delta t \simeq 0$, whereas it misses frames with $|\Delta t| > \Delta t_{\max}$, with a few exceptions. Thus, the Atheros AR928X adapter cannot tolerate symbol errors introduced by the CCT at MCS 7 in this experiment. On the contrary, Figure 26c shows that the Broadcom BCM4339 chip can also tolerate a TO $|\Delta t| > \Delta t_{\max}$ with an observable CFO of up to $\Delta f_{\max}\big|_{|\Delta t|>\Delta t_{\max}} \simeq 1600\,\text{Hz}$, which indicates excellent channel conditions.

3.5 CFO Characteristics of Devices

Since the IEEE 802.11 standard requires a frequency accuracy better than $\pm 20\,\text{ppm}$ for a channel bandwidth of 20 MHz [104, Section 17.3.9.5], commercial devices should typically operate within $\sim \pm 50\,\text{kHz}$ around the specified center frequency in the 2.4 GHz band. However, this frequency accuracy is not sufficient to facilitate a CCT with a moderate PSDU size of 140 bytes, as shown by the experiment results in Figure 25 and in Figure 26.

This section provides several CFO measurements of different WARP v3 SDRs [246] and Asus RT-AC86U wireless routers with respect to a particular WARP v3 SDR to assess their *actual* carrier frequency characteristics. To this end, multiple frame transmissions from each device under test (DUT) are captured with the WARP SDR. The captured frames are fed to a Matlab workspace through WARPLab [250] for offline evaluation.

The IQ samples of the received frames are processed with an IEEE 802.11 decoder software in Matlab. In Section 3.5.1, this step is performed by the IEEE 802.11 Non-HT decoder also used for the simulations presented in Figure 20, Figure 22 and Figure 24 and that is also part of the IEEE 802.11 Non-HT implementation used for the testbed experiments presented in Section 3.4.3. In Section 3.5.2, the IQ samples are processed by a decoding script that is based on the Matlab wireless LAN (WLAN) toolbox [150]. To get accurate CFO measurements, both IEEE 802.11 decoders employ an enhanced CFO estimation technique that operates in two steps, as described further in Section 3.6. In the first step, the IEEE 802.11 decoder estimates the CFO by means of the PHY preamble and corrects it through baseband processing of the IQ samples (see Section 2.4.4). In the second step, the IEEE 802.11 decoder refines the CFO estimate by means of the CPE (see Section 3.4.2.1), which significantly improves the accuracy of the CFO estimate. This CFO estimation technique is also integrated into the WARP prototype through an implementation on its field-programmable gate array (FPGA), as presented in Chapter 4.

In the following, CFO measurements are presented for WARP v3 SDRs in Section 3.5.1 and for Asus RT-AC86U wireless routers in Section 3.5.2. It is important to note that the IEEE 802.11 decoder does *not* estimate a CFO between the transmitters of a CCT in this section but the CFO between a single transmitter and the receiving WARP SDR. Furthermore, since carrier frequency oscillators are subject to jitter (see Section 2.2.2), each device may exhibit different CFO readings at different points in time. Still, each CFO reading may be considered to be valid for a certain amount of time during which the frequencies of the two respective oscillators do not change significantly. In accordance to the duration of frames transmitted in tests, this time span is found to be at least 2 ms for both device types, as indicated by a quite consistent CPE step size between successive OFDM symbols. With this, also the CPE-based CFO estimation can be optimally used.

3.5.1 *WARP v3 SDRs*

To measure the frequency characteristics of a WARP SDR with WARPLab [250] as a DUT, a Matlab script lets the DUT send 10 frames in succession with a pause of 10 s in between. The frame has an MSDU of 1508 bytes at MCS 0 (see Table 2), amounting to a PSDU of 1536 bytes (see Section 2.4.7) and a duration of 2072 µs. This frame length corresponds to a 1500-byte packet, i.e., the maximum transmission unit (MTU) size of Ethernet [102].

Table 3 lists the CFO characteristics of several WARP v3 SDRs. Each row contains the individual readings of the CFO between the respective DUT and the receiving WARP SDR. The last two columns hold the mean CFO and the standard deviation (std) of each DUT. Since most of the values in Table 3 are positive, the reference WARP SDR is at the lower end of the frequency range of the listed devices. Further, the largest absolute average CFO between any two devices in the list is $\Delta f = 2771.9\,\text{Hz} + 83.5\,\text{Hz} = 2855.4\,\text{Hz}$. Moreover, the standard deviation of the CFO readings varies between 64 Hz and 187.5 Hz, indicating a high frequency stability of all listed WARP SDRs, which is due to the temperature-compensated crystal oscillator (TCXO) of the WARP v3 (see Section 2.2.3).

Device ID	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	mean	std
W3-a-00537	reference device / receiver										0.0	0.0
W3-a-00140	25	-76	-127	-58	52	-262	-264	38	-222	56	-83.5	129.3
W3-a-00134	10	72	203	279	53	336	71	100	195	105	142.4	106.1
W3-a-00259	1759	1748	1504	1696	1507	1612	1544	1688	1374	1568	1599.9	123.6
W3-a-00121	2402	2295	1977	2001	2227	2014	1984	1885	2070	2108	2096.3	162.6
W3-a-00116	1167	913	808	867	846	825	818	788	1057	732	882.0	132.6
W3-a-00130	1308	1176	1259	1192	1403	1233	1156	1115	1253	1200	1229.4	82.6
W3-a-00120	1285	1049	747	978	1034	1319	1136	1084	961	773	1036.7	187.5
W3-a-00156	588	567	498	710	396	574	395	395	369	402	489.3	115.6
W3-a-00551	106	142	146	91	68	176	81	53	182	-30	101.5	64.0
W3-a-00516	2654	2832	2825	2674	2806	2842	2666	2757	2882	2783	2771.9	81.6
W3-a-00131	1749	1774	1572	1506	1322	1397	1512	1472	1279	1441	1502.1	162.5
W3-a-00579	1528	1380	1404	1445	1205	1270	1449	1280	1478	1465	1390.2	105.4
W3-a-00536	1111	895	1292	1224	1005	1225	1234	1213	1069	1201	1146.9	124.3

Table 3: CFO characteristics (Hz) of WARP v3 SDRs within a time interval of 100 s

Device ID	RTx IFS Q 5 % (ns)	RTx IFS median (ns)	RTx IFS Q 95 % (ns)	CFO Q 5 % (Hz)	CFO mean (Hz)	CFO Q 95 % (Hz)	CFO std (Hz)
W3-a-00537	reference device / receiver			0	0	0	0.0
RT-AC86U #3	18350	18550	18650	18842	19477	20066	360.4
RT-AC86U #5	18350	18500	18600	-50906	-50284	-49712	364.5
RT-AC86U #6	18450	18600	18700	-66032	-65464	-64925	337.5
RT-AC86U #7	18400	18550	18600	-72509	-71928	-71415	334.6
RT-AC86U #8	18450	18600	18700	-57294	-56832	-56440	265.8
RT-AC86U #9	18350	18550	18650	-49624	-49069	-48554	316.1
RT-AC86U #10	18350	18500	18600	-37145	-36632	-36205	288.3
RT-AC86U #11	18500	18650	18750	-43222	-42780	-42348	264.5
RT-AC86U #12	18400	18600	18700	-787	-351	60	256.3
RT-AC86U #13	18450	18600	18700	-28179	-27684	-27276	268.1
RT-AC86U #14	18350	18500	18600	-56752	-56205	-55668	325.9

Table 4: RTx IFS & CFO characteristics of Asus RT-AC86U devices for 1000 readings within 30 s

3.5.2 *Asus RT-AC86U Wireless Routers*

To measure the timing and frequency characteristics of Asus RT-AC86U wireless routers, the device firmware is modified such that a device retransmits a trigger frame with a specified MAC address after a SIFS of 16 µs (see Section 2.4.8), like in Section 6.5. The trigger frames are generated in software by means of the Matlab WLAN toolbox [150]. For each DUT, 1000 IEEE 802.11 HT trigger frames with a 32-byte MSDU are transmitted at MCS 0 within 30 s. The response frames from the DUT are captured by the WARP SDR and are saved for offline evaluation. The IQ samples are processed through a decoding script that is based on the Matlab WLAN toolbox [150].

Table 4 lists the characteristics of 11 different Asus RT-AC86U wireless routers in terms of their retransmission (RTx) IFS and their CFO, respectively. The RTx IFS is the time gap between the last sample of the trigger frame sent by the WARP SDR and the first sample of the received response frame from the DUT. Table 4 contains columns for the 5 % and the 95 % quantiles as well as the median of the RTx IFS. The RTx IFS of the RT-AC86U devices is apparently a bit larger than a SIFS but quite consistent across different devices, which would allow for accurate time synchronization in a CCT with RT-AC86U routers.

The CFO of most of the RT-AC86U devices is negative with respect to the WARP SDR. Note that the largest absolute CFO is found at the device RT-AC86U #7 with −72.5 kHz. Hence, the Asus RT-AC86U routers are not suitable for CCT without further measures such as frequency synchronization. Still, the standard deviation of the CFO readings ranges between 256.3 Hz and 364.5 Hz, which indicates quite stable carrier frequencies.

3.6 Enhanced CFO Estimation

IEEE 802.11 receivers estimate the CFO of a received frame by means of the PHY preamble and accordingly correct the CFO through baseband processing in order to mitigate ICI when applying an FFT on the received OFDM symbols, as described in Section 2.4.4. In this process, a receiver generates a single CFO estimate for each frame reception, representing the instantaneous CFO between the transmitter and itself. The CFO estimate based on the PHY preamble, however, may be inaccurate due to noise, so that a residual CFO may remain in the received signal also after the CFO correction. The residual CFO, in turn, can be addressed by a receiver through phase tracking and correction by means of the pilot subcarriers, as described in Section 3.4.2.1.

This section presents an enhanced CFO estimation technique that refines the CFO estimate from the PHY preamble through the CPE estimate from the pilot subcarriers. With this, the additional information of the estimated CPE is used to mitigate the impact of noise in the CFO estimate. This CFO estimation technique is particularly intended for frequency synchronization in CCT-based communication systems (see Section 1.2).

The CFO estimation starts with the PHY preamble in order to obtain a coarse CFO estimate based on the L-STF and a fine CFO estimate based on the L-LTF (see Section 2.4.2). For the L-STF-based CFO estimation, the receiver computes a weighted average of the phase shifts between the short training sequence (STS) symbols contained in the L-STF. In this process, the receiver may select only a subset U of the inner STS symbols (e.g., $U = \{2, 3, ..., 8\}$, omitting the indices 0, 1 and 9) to avoid ISI, which may occur in the outer STS symbols when the symbol timing estimate of the receiver is poor (see Section 2.4.3).

With a set of subcarriers $P_{\text{STS}} = \{-6, ..., -1, 1, ..., 6\}$ and by denoting the symbol on the n-th subcarrier of the u-th STS symbol as $C_{\text{STS}}(n; u)$, the phase shift $\Delta\phi_{\text{STS}}$ between two successive STS symbols is estimated as:

$$\Delta\hat{\phi}_{\text{STS}} = \arg\left(\sum_{u \in U} \sum_{n \in P_{\text{STS}}} C_{\text{STS}}(n; u) \cdot C^{*}_{\text{STS}}(n; u-1)\right) \tag{32}$$

With the STS symbol duration $T_{\text{STS}} = 800\,\text{ns}$, the CFO is estimated as:

$$\Delta\hat{f}_{\text{L-STF}} = \frac{\Delta\hat{\phi}_{\text{STS}}}{2\pi\, T_{\text{STS}}} \tag{33}$$

After this, the receiver corrects the CFO through baseband processing of the received IQ samples, which is typically carried out in hardware by real-time implementations. Further, the receiver may also refine its symbol timing estimate. However, there may be a residual CFO in the IQ samples due to a CFO estimation error. Thus, the receiver typically computes a refined CFO estimate by means of the L-LTF (see Section 2.4.2). With a set of subcarriers $P_{\text{LTS}} = \{-26, ..., -1, 1, ..., 26\}$ and by denoting the symbol on the n-th subcarrier of the u-th long training sequence (LTS) symbol as $C_{\text{LTS}}(n; u)$, the phase shift $\Delta\phi_{\text{LTS}}$ between the two LTS symbols is estimated as:

$$\Delta\hat{\phi}_{\text{LTS}} = \arg\left(\sum_{n \in P_{\text{LTS}}} C_{\text{LTS}}(n; 2) \cdot C^{*}_{\text{LTS}}(n; 1)\right) \tag{34}$$

With the LTS symbol duration $T_{\text{LTS}} = 3.2\,\mu\text{s}$, the residual CFO is estimated as:

$$\Delta\hat{f}_{\text{L-LTF}} = \frac{\Delta\hat{\phi}_{\text{LTS}}}{2\pi\, T_{\text{LTS}}} \tag{35}$$

The receiver either corrects the residual CFO of the already processed IQ samples or it combines the L-STF-based and the L-LTF-based CFO estimates to correct the CFO of the original IQ samples. Note that the CFO estimation and correction steps described above are common practice in receiver implementations [150, 263]. However, a residual CFO may remain also after the L-LTF-based CFO correction, especially in low SNR regimes. Hence, the enhanced CFO estimation technique refines the CFO estimate another time *after* applying an FFT on the OFDM symbols. To this end, the receiver estimates the CPE by means of the pilot subcarriers, as described in Section 3.4.2.1. By using Equation 18, the unwrapped CPE estimate of an OFDM symbol with index q can be written as:

$$\hat{\psi}_{\text{u}}(q; r) = \sum_{x=r+1}^{q} \arg\left(\mathrm{e}^{j\left(\hat{\psi}(x;r) - \hat{\psi}(x-1;r)\right)}\right) \tag{36}$$

Note that the unwrapped CPE has an offset of a multiple of 2π from the wrapped CPE:

$$\exists\, k \in \mathbb{Z}: \;\; \hat{\psi}(q; r) + k \cdot 2\pi = \hat{\psi}_{\text{u}}(q; r) \quad \forall\, q \in \{r+1, ..., Q-1\} \tag{37}$$

With a regular OFDM symbol duration of $T_c = 4\,\mu\text{s}$, the residual CFO is estimated by means of the unwrapped CPE estimate of the last OFDM symbol with index $Q-1$:

$$\Delta\hat{f}_{\text{CPE}} = \frac{\hat{\psi}_{\text{u}}(Q-1; r)}{2\pi\,(Q-(r+1))\,T_c} \tag{38}$$

This residual CFO estimate is particularly robust to noise, which can be shown by separating the true CPE $\psi(q;r)$ from the estimation error $\epsilon(q)$ on the q-th OFDM symbol:

$$\hat{\psi}(q;r) = \psi(q;r) + \epsilon(q) \tag{39}$$

By taking into account that the CFO is assumed to be invariant over the frame duration, the CPE-based estimate of the residual CFO from Equation 38 can be written as:

$$\begin{aligned} \Delta\hat{f}_{\text{CPE}} &= \frac{\psi_{\text{u}}(Q-1;r) - \epsilon(r) + \epsilon(Q-1)}{2\pi\,(Q-(r+1))\,T_c} \\ &= \Delta f_{\text{CPE}} + \frac{\epsilon(Q-1) - \epsilon(r)}{2\pi\,(Q-(r+1))\,T_c} \end{aligned} \tag{40}$$

This representation shows that noise affects the CPE-based estimate of the residual CFO only through the OFDM symbols with the indices r and $Q-1$. With this, the impact of noise on the CPE-based CFO estimation is reduced with an increasing frame length.

The CPE-based estimate of the residual CFO is not intended to be applied on the received IQ samples. Therefore, it does not mitigate the ICI of a received frame, unless a receiver restarts decoding after having processed the pilot subcarriers of the frame. Since each residual CFO estimate is obtained after having shifted the frequency of the IQ samples according to the previous CFO estimate, respectively, the enhanced CFO estimate of the original frame reception is obtained by adding the individual estimates:

$$\Delta\hat{f} = \Delta\hat{f}_{\text{L-STF}} + \Delta\hat{f}_{\text{L-LTF}} + \Delta\hat{f}_{\text{CPE}} \tag{41}$$

The enhanced CFO estimation is validated in a testbed experiment with a WARP SDR. The WARP SDR sends an IEEE 802.11 Non-HT frame on one RF interface while shifting the frequency artificially through baseband processing. In addition, the WARP SDR captures its own transmission with the other RF interface. As both RF interfaces are driven by a shared clock (see Section 2.2.3), the true CFO of the received frame is equal to the known artificial CFO. With a PSDU of 1536 bytes at MCS 0 (see Table 2), the frame has $Q = 518$ OFDM symbols and a duration of 2072 µs, including the four pseudo OFDM symbols for the PHY preamble, as introduced in Section 3.4.1.

The CFO estimation by means of the L-STF and the L-LTF is found to deviate from the true CFO by typically hundreds to a few thousands of hertz, depending on the SNR. In contrast, the enhanced CFO estimation according to Equation 41 is typically accurate to $\sim \pm 10$ Hz around the true CFO. With this, the enhanced CFO estimation technique provides substantial improvements over the preamble-based CFO estimation, making it suitable for frequency synchronization of distributed network nodes to facilitate CCT.

3.7 Conclusion

The preceding analysis reveals that the IEEE 802.11 DSSS PHY [104, Section 15.3] and the OFDM-based IEEE 802.11 Non-HT PHY [104, Section 17.3] pose different requirements on the synchronization in time and frequency between concurrent transmitters to facilitate the reception of a CCT with state-of-the-art receivers.

With the IEEE 802.11 DSSS PHY, the decodability of a CCT is governed by the effects arising in the cases with and without time synchronization of the transmitters, respectively.

When the transmitters of a CCT are synchronized in time, i.e., when $\Delta t \simeq 0\,\text{s}$, they also have to be synchronized in frequency to facilitate correct decoding of the superimposed signal by a receiver. Otherwise, when the TO Δt between two transmitters of a CCT exceeds the chip duration $T_p = \frac{1}{11}\,\mu\text{s} = 90.\overline{90}\,\text{ns}$, the capture effect can be exploited with and without frequency synchronization, unless the TO Δt is a multiple of the symbol duration $T_c = 1\,\mu\text{s}$. Furthermore, Figure 17c and Figure 18c show that the FRR is quite consistent in testbed experiments in both cases $\Delta t \simeq 0\,\text{s}$ and $|\Delta t| > T_p$, i.e., the centers of the maps have the same shades as the respective regions left and right.

However, these CCT characteristics of the IEEE 802.11 DSSS PHY differ slightly from the characteristics of the IEEE 802.15.4 O-QPSK DSSS PHY [107, Section 12] that is employed by Glossy [73]. In a CCT with the IEEE 802.15.4 O-QPSK DSSS PHY, the best frame detection and reception performance is achieved when the TO between the concurrent transmitters stays below the chip duration $T_p = 0.5\,\mu\text{s}$ [73, Figures 2 and 11]. Otherwise, when the TO in a CCT with the IEEE 802.15.4 O-QPSK DSSS PHY exceeds the chip duration $T_p = 0.5\,\mu\text{s}$, frames may still be detected and received due to the capture effect, but with a reduced rate [73, Figures 2 and 11]. This difference between the IEEE 802.11 DSSS PHY and the IEEE 802.15.4 O-QPSK DSSS PHY might be primarily due to their different spreading sequences. In particular, the 11-chip Barker code employed by the IEEE 802.11 DSSS PHY has minimum off-peak auto-correlation coefficients [93], while the 32-chip sequences employed by the IEEE 802.15.4 O-QPSK DSSS PHY are "related to each other through cyclic shifts and/or conjugation" [107, Section 12.2.5, Table 12-1]. Therefore, to facilitate CCTs with a TO Δt exceeding the PHY-specific chip duration T_p, the IEEE 802.11 DSSS PHY is better suited than the IEEE 802.15.4 O-QPSK DSSS PHY.

In order to evade the reduced FRR performance in case of $|\Delta t| > T_p$, Glossy employs techniques to achieve time synchronization better than the chip duration $T_p = 0.5\,\mu\text{s}$. Its authors accordingly define interference to be constructive "if a receiver detects the superposition of the baseband signals generated by multiple transmitters" [73]. By analogy, they define interference to be destructive "if it prevents a receiver from correctly detecting the superimposed baseband signals" [73]. However, these definitions are rather general and do not take any specific conditions of the interfering signals into account. In fact, follow-up research with Glossy shows that the reception performance in a CCT can be enhanced further through frequency synchronization of the transmitters, especially with long frames [236, 257, 258], as discussed in Section 2.1.5.

The system model introduced in Section 3.2 takes a more sophisticated approach to characterize interference and introduces three parameters that describe certain relations between the superimposing signals in a CCT: the PO $\Delta\theta$, the CFO Δf and the TO Δt. The analyses in Section 3.3 and in Section 3.4 consider the interference of two signals to be constructive if their instantaneous phases are aligned to each other, which causes an increase of the resulting signal amplitude. Accordingly, the interference of two signals is considered to be destructive if their instantaneous phases are inverted with respect to each other, causing a deep fade of the superposition. In this sense, the occurrence of events with constructive and destructive interference in a CCT depends on all three factors $\Delta\theta$, Δf and Δt as well as the elapsed time since the start of the CCT.

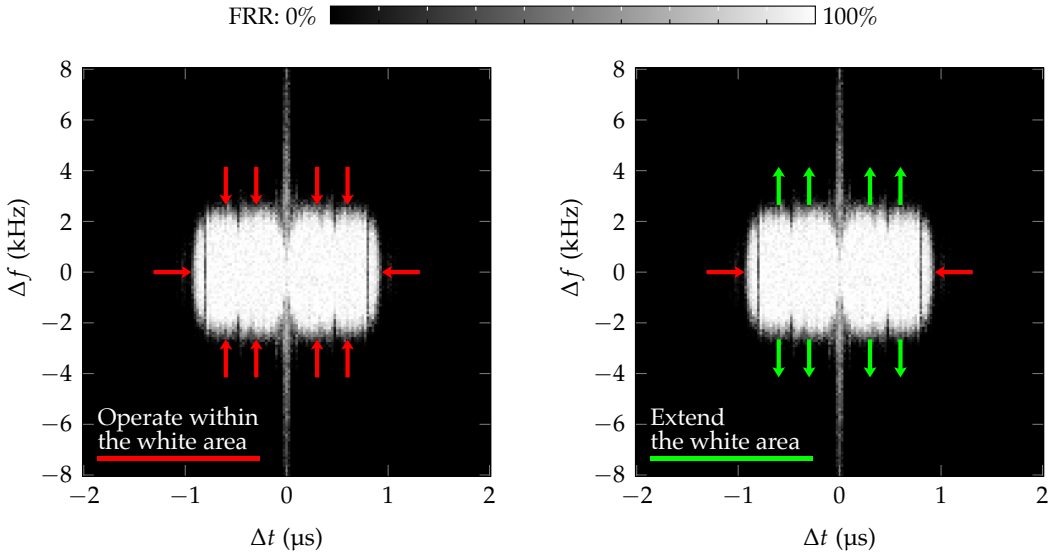
In case of a CCT with time synchronization, i.e., $\Delta t = 0\,\text{s}$, Section 3.3.1 shows that the IEEE 802.11 DSSS PHY necessitates also frequency synchronization of the transmitters. More precisely, Equation 5 provides the maximum tolerable CFO $\Delta f_{\max}$ of a CCT.

However, a CCT with $\Delta f_{\max}$ can only be received with a specific value of the PO $\Delta\theta$. Since the PO $\Delta\theta$ is randomly distributed in $]-\pi, \pi]$ in practice, a CCT frame reception with $\Delta f \leq \Delta f_{\max}$ is only successful with a certain probability as given by Equation 6. Section 3.4.1 shows that the IEEE 802.11 Non-HT PHY has similar characteristics in a CCT with $\Delta t = 0\,\text{s}$ as the IEEE 802.11 DSSS PHY. Accordingly, Equation 16 provides the maximum tolerable CFO $\Delta f_{\max}$ and Equation 17 provides the corresponding frame reception probability. Figure 20 illustrates CCTs with $\Delta t = 0\,\text{s}$ and two different values of the PO $\Delta\theta$ for the IEEE 802.11 Non-HT PHY. In case of a successful CCT frame reception, there may be events of both constructive and almost completely destructive interference.

In case of a CCT with imperfect time synchronization, i.e., $|\Delta t| > \Delta t_{\max}$ according to Equation 15, Section 3.4.2 shows that the IEEE 802.11 Non-HT PHY is subject to symbol errors occurring systematically on specific OFDM subcarriers. These symbol errors can be corrected to a certain degree through FEC by a receiver, depending on the coding rate of the employed MCS (see Table 2), the convolutional decoding algorithm employed by the receiver as well as the impact of noise in accordance with the modulation scheme of the employed MCS. The total amount of symbol errors within the last OFDM symbol of a CCT frame is quite consistent for a range of values of the TO Δt, so that a receiver is typically able to decode a CCT frame up to a certain CFO Δf for various values of the TO Δt within the duration of the GI. Only for $|\Delta t| < \sim 150\,\text{ns}$, the amount of symbol errors may vary across different CCT frame receptions since it additionally depends on the PO $\Delta\theta$ in this case, which is random in practice, as discussed in Section 3.4.2.2.

To enable CCT-based network flooding and extensive broadcasting with the IEEE 802.11 Non-HT PHY, the corresponding synchronization requirements must be met exactly. On the one hand, the IEEE 802.11 Non-HT PHY employs a GI of 800 ns that is filled with a cyclic prefix (CP) to accommodate the channel delay spread (CDS), which in turn occurs in multi-path propagation environments (see Section 2.4.6). When the CDS is shorter than the GI, there is a certain clearance to accommodate the TO of a CCT. Figure 25b and Figure 26b show that the tolerable TO in a CCT is in fact equal to the GI duration of 800 ns in the office environment of the testbed experiment. Thus, with a TO of up to 800 ns, ISI may be largely avoided in a CCT with the IEEE 802.11 Non-HT PHY. On the other hand, the frequency synchronization requirement of a CCT depends on factors such as the coding rate of the used MCS, the convolutional decoding algorithm employed by the receiver, the frame duration and the SNR. With a 140-byte PSDU, for example, Figure 25 and Figure 26 show that the CFO Δf between the transmitters of a CCT should not exceed 1 to 2 kHz in the 2.4 GHz band, depending on the MCS.

Section 3.5 presents the carrier frequency characteristics of various WARP SDRs and Asus RT-AC86U wireless routers. Table 3 shows that most of the measured WARP SDRs meet the aforementioned CFO requirement. An unsyntonized CCT with WARP SDRs may thus be decodable by state-of-the-art receivers, especially for frames with a PSDU of less than 140 bytes. Furthermore, Table 4 shows that the Asus RT-AC86U wireless routers have much larger CFOs with respect to each other than the WARP SDRs, which might inhibit the reception of unsyntonized CCTs from these devices. Still, Table 4 also shows that the Asus RT-AC86U wireless routers have very consistent timing characteristics, i.e., their median RTx IFS values do not differ by more than 150 ns. Even if a TO of $\Delta t = 150\,\text{ns}$ accumulates at each hop in CCT-based network flooding (see Section 1.2), a flooding frame could propagate five hops before the TO reaches the GI duration of 800 ns.



(a) Synchronization in time and frequency

(b) Relaxation of frequency synchronization

Figure 27: CCT modes of operation with (a) coherent and (b) incoherent interference

This work explores two different approaches to facilitate CCTs with distributed devices, as illustrated in Figure 27. The first approach is shown in Figure 27a and comprises synchronization in time and frequency, i.e., the TO Δt and the CFO Δf of the transmitters should both be kept as small as possible. On the one hand, accurate time synchronization mitigates ISI and can be achieved through hardware timers, as shown for Asus RT-AC86U wireless routers in Table 4. On the other hand, accurate frequency synchronization allows generating coherent interference (CI), i.e., the phase relation of concurrent transmitters is kept invariant throughout a CCT. Notably, CI may comprise both constructive and destructive interference on different OFDM subcarriers at the same time. The relevant feature of CI is that it enables a receiver to process the superposition of a CCT like a signal received from a single sender after undergoing multi-path propagation effects. Distributed nodes may leverage the PHY preamble to synchronize in frequency according to their respective CFO estimates during a frame reception, as described in Section 2.4.4. Section 3.6 shows that a CPE-based CFO estimation can enhance the accuracy further. Chapter 4 and Chapter 5 present prototypes that realize the approach of Figure 27a and that are based on WARP v3 SDRs and Asus RT-AC86U wireless routers, respectively.

The second approach is shown in Figure 27b and comprises only time synchronization, whereas the requirement on the frequency synchronization in a CCT should be relaxed. One goal is to support scenarios in which distributed nodes cannot be synchronized in frequency through a common frame reception, so that the transmitters operate with their native carrier frequencies. Note that the extensive broadcasting scenario presented in Section 1.2.2 may fall into this category. Another goal is to enhance the reliability of CCT-based low-latency communications by making a receiver more robust to frequency synchronization errors in a CCT. Chapter 6 presents time-variant zero-forcing (TVZF), an equalization technique that deals with incoherent interference at the receiver's end. To enhance the tolerance towards the CFOs in a CCT, TVZF essentially analyzes and reconstructs the time-variant interference of a received CCT frame to equalize it.

4 SOFTWARE-DEFINED RADIO PROTOTYPE – WARP

This chapter presents a prototype for concurrent cooperative transmission (CCT) with coherent interference (CI) through synchronization in time and frequency in real-time. The prototype is based on the Wireless Open Access Research Platform (WARP) v3 software-defined radio (SDR) [246], which has a Xilinx Virtex-6 field-programmable gate array (FPGA), allowing physical hardware components, such as radio frequency (RF) interfaces, to be connected to logical hardware components, such as signal processing algorithms running in the FPGA fabric. Software components running on soft processors in the FPGA can flexibly control the hardware components and perform computations. The prototype is based on the IEEE 802.11 reference design for WARP v3 [247], which is a real-time implementation of the IEEE 802.11g/n physical layers (PHYs) and medium access control (MAC), allowing for interaction with commercial IEEE 802.11 devices. The IEEE 802.11 reference design for WARP v3 is modified such that the prototype can perform a retransmission (RTx) of a received flooding frame with synchronization in time and frequency with respect to the received frame itself, which in turn may arrive as a transmission from a single device or as a CCT with CI from a group of devices. With this, multiple WARP prototype nodes can rapidly disseminate data in an area through CCT-based network flooding, as introduced in Section 1.2.1.

The WARP prototype enables the initiator of a flooding process to tune the latency, the reliability and the area coverage through a low-level MAC protocol so that different application demands can be met. Since the WARP prototype processes the low-level MAC protocol in real-time, the behavior of a network with multiple nodes is configured for each flooding process through the flooding frame itself. The flooding initiator defines the modulation and coding scheme (MCS) and the payload, which in combination govern the frame transmission duration and hence the latency per hop of a flooding process. Further, the reliability of a flooding process can be tuned through the low-level MAC protocol by defining the number of RTx repetitions at each node, as illustrated in Figure 4. In addition, the WARP prototype can detect whether its next-hop neighbors generate an RTx of a forwarded frame. If a node does not detect an expected RTx from its next-hop neighbors, it can automatically retry to reach them by repeating its own RTx, according to a setting in the low-level MAC protocol. Further, the low-level MAC protocol comprises a sequence number by which each node can detect duplicate frame receptions so that each flooding process can be terminated. While the WARP prototype deviates from the medium access policies of the IEEE 802.11 standard, it retains PHY compatibility and employs the IEEE 802.11 MAC header, enabling commodity devices to receive data frames disseminated through CCT-based network flooding. In addition, the low-level MAC protocol also allows commercial devices to inject data frames for flooding.

Figure 28 shows an architectural overview of the WARP prototype, which comprises certain modifications of the IEEE 802.11 reference design [247]. First, the PHY receive (Rx) core is extended with an enhanced carrier frequency offset (CFO) estimator [9] that works with the common phase error (CPE) on the pilot subcarriers, as described in Section 3.6.

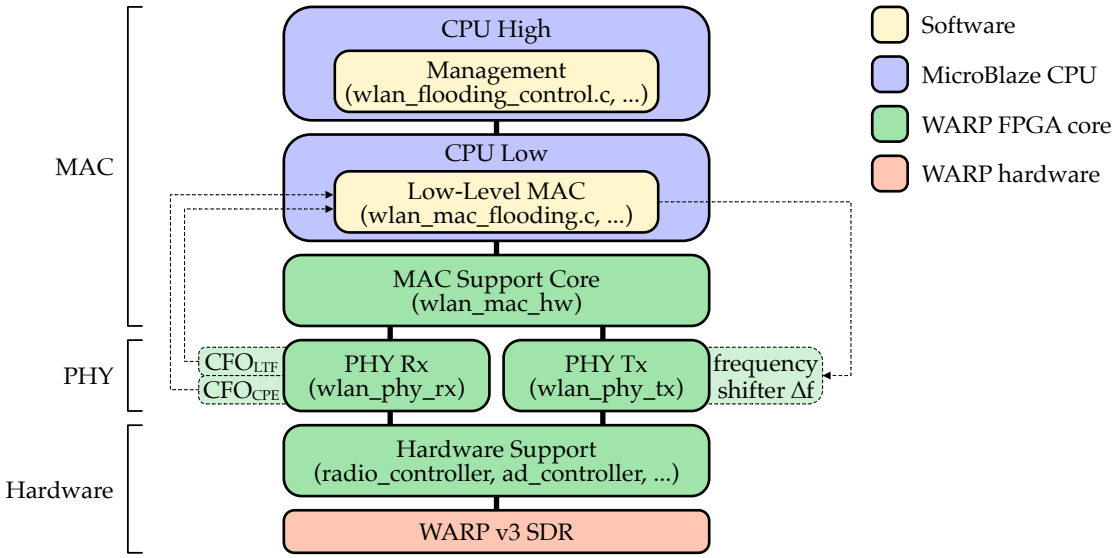


Figure 28: Architecture of the WARP prototype for CCT with CI: the IEEE 802.11 reference design for WARP v3 [247] is extended with an enhanced CFO estimator in the PHY Rx core [9], a frequency shifter in the PHY Tx core [9], a low-level MAC and management software.

This extension refines the CFO estimation through the legacy long training field (L-LTF), which is provided by the IEEE 802.11 reference design, as described in Section 2.4.4. Both the L-LTF-based CFO estimate and the CPE-based estimate of the residual CFO are exposed to the central processing units (CPUs) through memory-mapped registers. Second, the PHY transmit (Tx) core is extended with a frequency shifter that allows the frequency of an RTx to be synchronized to the frequency of a previous frame reception [9]. Third, the low-level MAC for CCT-based network flooding runs on CPU Low and replaces the IEEE 802.11 distributed coordination function (DCF) [104, Section 10.3] provided by the IEEE 802.11 reference design. It executes the low-level MAC protocol upon a frame reception, employs the MAC support core to achieve precise time synchronization and accesses memory-mapped registers of the PHY Rx and the PHY Tx cores to perform frequency synchronization. Fourth, the software running on CPU High performs tasks that are less time critical than the low-level MAC. In particular, it allows flooding-related settings to be managed through a user interface by means of a universal asynchronous receiver-transmitter (UART), push buttons and two 7-segment displays of the WARP v3. While the software is based on the IEEE 802.11 independent basic service set (IBSS) implementation of the reference design, IBSS functions such as probe requests and probe responses are disabled to avoid side effects. The software running on CPU High can be flexibly extended, e.g., to process received frames further, to connect the wireless flooding facilities with an Ethernet bridge or to periodically schedule messages for flooding.

This chapter is organized as follows. Section 4.1 presents the hardware design of the WARP prototype, which comprises the enhanced CFO estimator in the PHY Rx core and the frequency shifter in the PHY Tx core. Section 4.2 presents the software design of the WARP prototype, which comprises the low-level MAC protocol, three MAC header formats that facilitate different modes of operation, the user interface with a UART, push buttons, 7-segment displays and light-emitting diodes (LEDs), a mechanism that prevents the CFO from drifting during a flooding process, a logging facility capable of

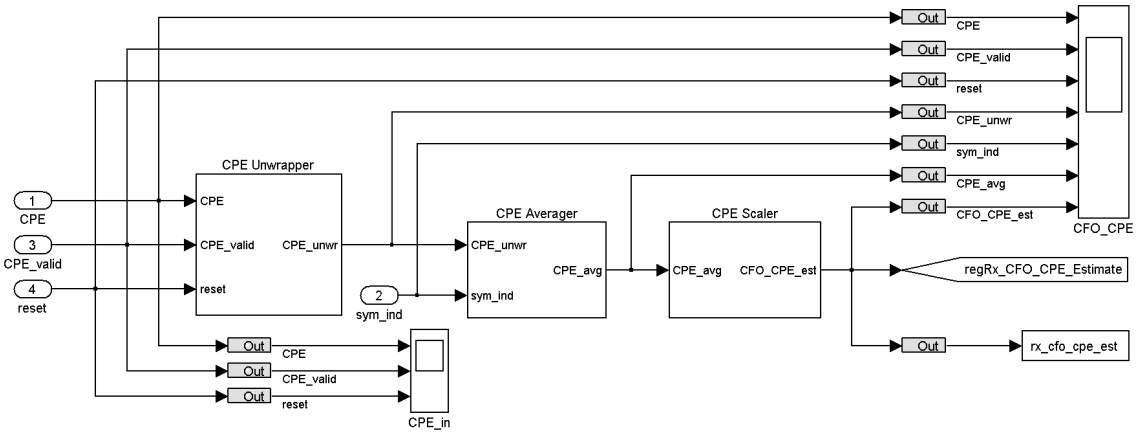
calculating statistical evaluation metrics of a flooding process and two debug outputs. Section 4.3 describes the processing of a received frame in real-time and highlights the critical processing paths for short and long MAC service data units (MSDUs), respectively. Section 4.4 presents functional validations of the WARP prototype regarding its ability to deal with sporadic interference caused by foreign devices, the calibration of its time synchronization mechanism, the containment of the CFO during a flooding process and the accuracy of its frequency synchronization. Section 4.5 presents the evaluation of the WARP prototype in testbed experiments in which various performance metrics are acquired in runs of 10 000 successive CCTs with up to nine concurrent transmitters. Section 4.6 provides concluding remarks.

4.1 Hardware Design

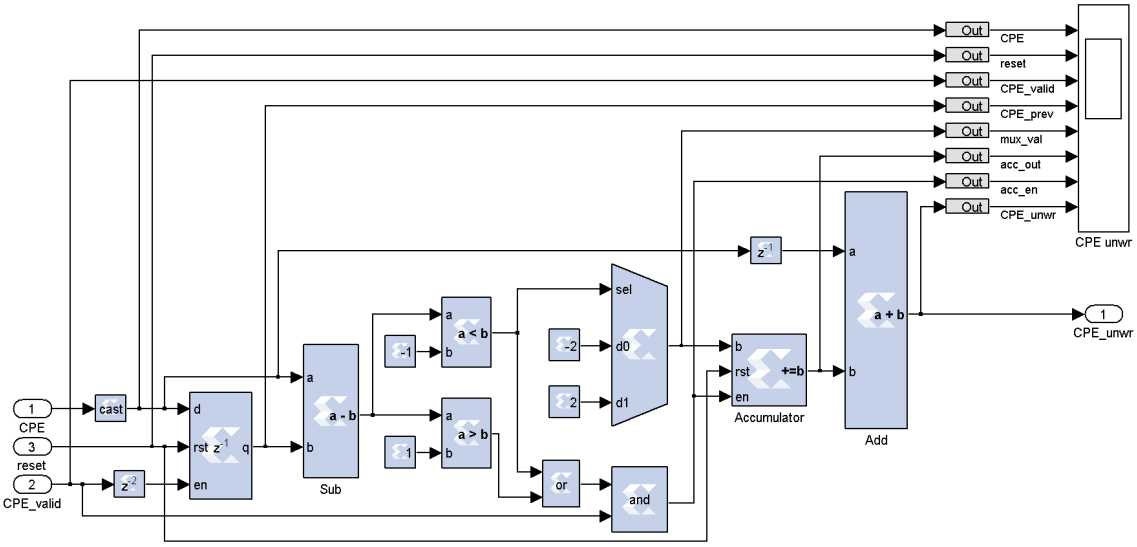
The hardware design of the WARP prototype comprises an enhanced CFO estimator in the PHY Rx core and a frequency shifter in the PHY Tx core, which in combination allow multiple uncoupled nodes to synchronize their carrier frequencies during a CCT by means of an immediately preceding frame reception [9]. The enhanced CFO estimator in the PHY Rx core performs computations on an incoming stream of data while a frame is received so that the CFO estimate becomes available immediately after a frame reception. Similarly, the frequency shifter in the PHY Tx core processes the outgoing in-phase quadrature (IQ) samples in real-time while a frame is transmitted. The FPGA cores of the WARP v3 are designed with Xilinx System Generator blocks in Matlab Simulink, allowing for simulations in Simulink on the one hand and for FPGA synthesis on the other hand. Both hardware components of the WARP prototype are validated in isolation and in combination with each other in various simulations and testbed experiments [9].

4.1.1 *PHY Rx Core: Enhanced CFO Estimator*

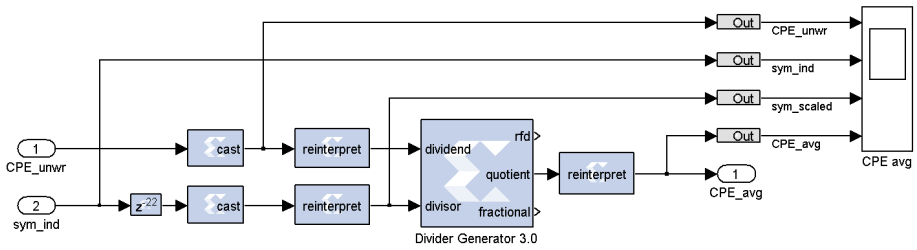
The PHY Rx core of the IEEE 802.11 reference design for WARP v3 [247] estimates the CFO during a frame reception by means of the L-LTF and corrects it through baseband processing in order to mitigate intercarrier interference (ICI), as described in Section 2.4.4. Still, a CFO estimate from the PHY preamble may be subject to an error ranging up to hundreds or a few thousands of hertz, depending on the signal-to-noise ratio (SNR). To cope also with the residual CFO that remains after the L-LTF-based CFO correction, the PHY Rx core of the IEEE 802.11 reference design corrects the phase of the received signal according to the CPE obtained from the pilot subcarriers, after a fast Fourier transform (FFT) of the orthogonal frequency-division multiplexing (OFDM) symbols. However, the WARP prototype of this work requires an accurate phase alignment not only with respect to a transmitter during a frame reception but also with respect to other transmitters during an RTx to generate a CCT with CI. Hence, the PHY Rx core of the IEEE 802.11 reference design [247] is extended with an enhanced CFO estimator that obtains the residual CFO by means of the CPE [9], as described in detail in Section 3.6. With this, the L-LTF-based CFO estimate and the CPE-based estimate of the residual CFO can be combined to obtain a more accurate CFO estimate that is more robust to noise. Both estimates are exposed to the MicroBlaze CPUs through memory-mapped registers so that they can be used for frequency synchronization shortly after a frame reception.



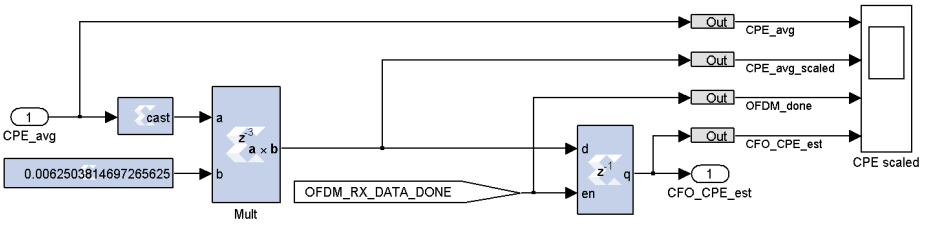
(a) Overview: the CPE from the PHY Rx core is processed through three custom blocks.



(b) The CPE unwrapper accumulates the phase shift over all OFDM symbols.



(c) The CPE averager computes the average phase shift per OFDM symbol.



(d) The CPE scaler computes and buffers the average phase shift per IQ sample.

Figure 29: CPE-based estimator of the residual CFO within the PHY Rx core [9]

Figure 29 shows the Simulink model of the CPE-based estimator of the residual CFO [9], which is integrated into the module for pilot subcarrier processing in the PHY Rx core. Figure 29a shows the three top-level blocks of the estimator that take as inputs the CPE, a signal that indicates if the CPE is valid, a reset signal and the OFDM symbol index. The blocks in the top right corners of the subfigures are virtual scopes for waveform simulations in which the PHY Rx core takes an IEEE 802.11 frame as input.

Figure 29b shows the interior of the first block, which computes the unwrapped CPE. Note that angular quantities are provided in an interval of $[-\pi, \pi[$ and are normalized by $\frac{1}{\pi}$ in the Simulink model. The CPE unwrapper detects overruns of the CPE across the boundary at $\pm\pi$ in any direction and accordingly accumulates full phase rotations by 2π or -2π in a register. In particular, the CPE unwrapper accumulates 2π when the difference between the CPE values of two successive OFDM symbols is smaller than $-\pi$, whereas it accumulates -2π when the difference exceeds π.

Figure 29c shows the interior of the second block, which computes the average phase shift between successive OFDM symbols by dividing the unwrapped CPE by the current OFDM symbol index. Note that this procedure mitigates the impact of noise with an increasing number of OFDM symbols, as described in Section 3.6.

Figure 29d shows the interior of the third block, which scales the average phase shift per OFDM symbol down to the average phase shift per IQ sample. Since the frequency shifter in the PHY Tx core uses this representation, the residual CFO is not computed in terms of hertz. Furthermore, this block buffers the result in a register when the frame reception is done so that the estimate can be read by a CPU shortly afterwards.

The performance of the enhanced CFO estimator, which combines the L-LTF-based CFO estimate with the CPE-based estimate of the residual CFO, is validated in various simulations and testbed experiments [9]. In the simulations, the PHY Rx core processes an IEEE 802.11 frame that has an artificial CFO, which is introduced by shifting the frequency of the baseband signal through multiplication with a complex exponential. Simulations conducted with an MSDU of 100 bytes at MCS 0 (see Table 2) and with an SNR of 20 dB show that the L-LTF-based CFO estimate has an error of up to $\sim \pm 1.5$ kHz, whereas the enhanced CFO estimate has an error of up to $\sim \pm 80$ Hz [9]. The accuracy of the enhanced CFO estimate is very similar for CFOs of -5 kHz, 0 Hz and 5 kHz [9].

Testbed experiments are conducted with a WARP SDR running WARPLab [250] and a WARP SDR with the modified PHY Rx core as device under test (DUT), while both SDRs are driven by a shared clock through wired synchronization [9]. The WARP SDR running WARPLab transmits IEEE 802.11 frames that are generated with an artificial CFO in Matlab, while the DUT prints the estimated CFO of each received frame via UART. Since both SDRs are driven by the same clock, the true CFO at the DUT is exactly the artificial CFO, so that the estimation error is the difference between the estimated CFO and the artificial CFO. The WARP SDRs are placed in line-of-sight (LOS) at a distance of 30 cm from each other. With an MSDU size of 100 bytes at MCS 0 and with artificial CFOs in the range of $[-80, 80]$ kHz, the error of the L-LTF-based CFO estimate is consistently in the range of $\sim [-1.2, 1.6]$ kHz, whereas the error of the enhanced CFO estimate is consistently in the range of $\sim [-60, 45]$ Hz [9]. This result confirms that the implementation of the enhanced CFO estimator is effective, in accordance with Section 3.6.

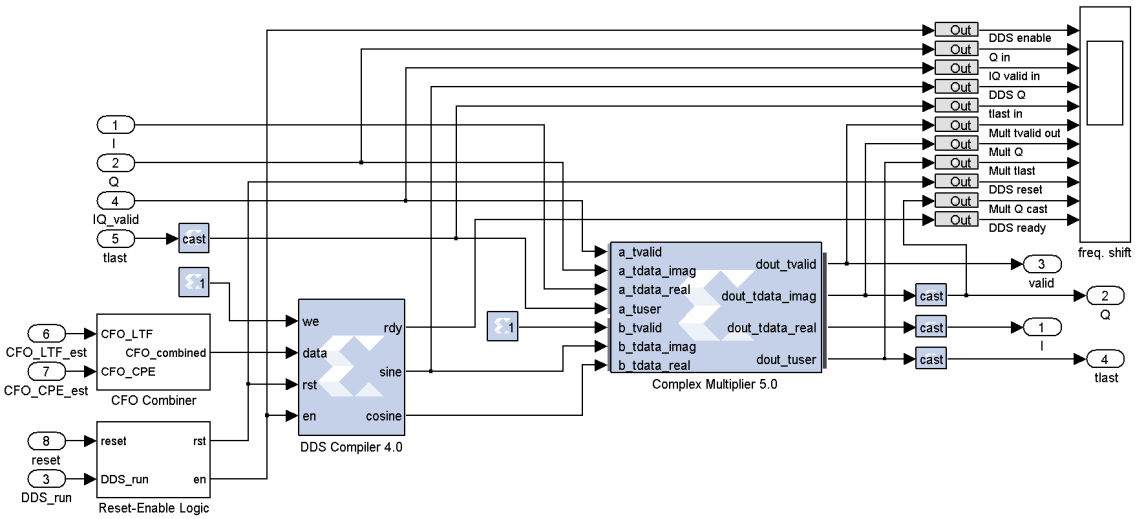


Figure 30: Frequency shifter within the PHY Tx core [9]

4.1.2 *PHY Tx Core: Frequency Shifter*

Figure 30 shows the Simulink model of the frequency shifter [9], which is integrated on the signal path of the generated IQ samples to the digital-to-analog converter (DAC) outputs of the PHY Tx core. The frequency of the baseband signal is shifted by multiplying the IQ samples with a complex exponential generated through direct digital synthesis (DDS). The frequency of the DDS signal is the sum of the L-LTF-based CFO estimate and the CPE-based estimate of the residual CFO, which are provided through memory-mapped registers by a CPU. The design of the frequency shifter is based on the phase correction in the PHY Rx core [247] and incorporates certain adjustments of the input signals and the DDS compiler to facilitate the integration into the PHY Tx core [9]. According to the precision of the fixed-point computations, the quantization step size of the frequency shifter is 20 Hz. This is considered to be sufficiently accurate since the practical accuracy of the enhanced CFO estimation technique presented in Section 3.6 is comparable.

In addition, to ensure that different WARP SDRs generate the same waveform for the same MSDU in a CCT, the PHY Tx core is also configured to reset the linear-feedback shift register (LFSR) of the scrambler to a fixed value for each frame generation [9].

The frequency shifter of the WARP prototype is validated in various simulations and testbed experiments [9]. In simulations, the PHY Tx core generates and exports IEEE 802.11 frames with an artificial CFO that are analyzed in Matlab with the enhanced CFO estimation technique described in Section 3.6. The error of the frequency shifter is the difference between the estimated CFO and the artificial CFO. The simulations confirm that the frequency shifter is accurate to $\sim \pm 10$ Hz for artificial CFOs in the range of $[-80, 80]$ kHz and for MSDU sizes from 100 bytes to 800 bytes at MCS 0 [9].

In testbed experiments, the combined accuracy of the enhanced CFO estimator in the PHY Rx core and the frequency shifter in the PHY Tx core is assessed by letting a DUT retransmit a trigger frame with frequency synchronization [9]. With an MSDU size of 100 bytes at MCS 0, the frequency is typically accurate to $\sim [-54, -10]$ Hz [9]. With an MSDU size of 800 bytes at MCS 0, it is typically accurate to $\sim [0, 15]$ Hz [9]. Section 4.4.4 presents a more comprehensive characterization of the frequency synchronization.

4.2 Software Design

The WARP prototype comprises various functions running as software components on the two MicroBlaze processors. Section 4.2.1 presents the protocol fields of the low-level MAC protocol that allow a receiver to decide within a few microseconds whether to retransmit a received frame. Section 4.2.2 describes the frame format of the low-level MAC protocol as well as alternative frame formats that enable commercial devices to inject data frames for flooding and that allow for experiments with many successive CCTs, respectively. Section 4.2.3 describes the user interface of the WARP prototype that facilitates the configuration of a node through push buttons and a UART interface and that visualizes events through two 7-segment displays and LEDs. Section 4.2.4 presents a feature that contains the CFO within a standard-conformant range during a flooding process with a large number of retransmissions. Section 4.2.5 presents a logging facility that facilitates the computation of evaluation metrics after a flooding process. Section 4.2.6 describes two debug outputs that expose the internal state of the WARP prototype during a flooding process and that are used to check the adherence to certain timing constraints.

4.2.1 *Low-Level MAC Protocol*

The WARP prototype implements a low-level MAC protocol that empowers a flooding initiator to tune the latency, the reliability and the area coverage of a flooding process. The WARP prototype processes the low-level MAC protocol in real-time upon a frame reception and decides within a few microseconds whether to retransmit a received frame. In addition, a received frame must have a correct frame check sequence (FCS) in order to qualify for an RTx. With this, the behavior of the overall system during a flooding process can be defined by means of the flooding frame itself, i.e., without additional signaling. While the area coverage and the reliability of a flooding process can be tuned explicitly through protocol fields, the latency is defined implicitly as follows.

The latency per hop is the sum of the frame transmission duration and the subsequent idle period before the next RTx, the latter of which is a short interframe space (SIFS) with a duration of 16 µs for the Non-HT PHY (see Section 2.4.8). Therefore, the flooding initiator can tune the latency per hop through the MSDU size on the one hand and through the MCS (see Table 2) on the other hand. However, a higher MCS reduces the communication range since it requires a higher SNR, so the propagation speed of a flooding process is not necessarily increased with a higher MCS. Thus, the flooding initiator should select an MCS that is suitable for the node density of the network.

4.2.1.1 *Sequence Number*

The protocol field `sequence_number` allows a node to identify already processed frames. Sequence numbers are valid within a certain range ahead of the current sequence number. When a node receives a frame with a new sequence number that is valid, it updates its internal state according to the protocol fields of the received frame and possibly retransmits the frame, according to the conditions of the protocol. When a node receives a frame with the current sequence number, it possibly retransmits the frame another time, according to the conditions of the protocol. When a node receives a frame with an invalid sequence number, it discards the frame and keeps its internal state.

4.2.1.2 *Global Retransmission Counter*

The protocol field `rtx_global_counter` denotes the number of remaining Tx slots of a flooding frame in the network. The initiator of a flooding process uses this protocol field to define up to how many hops a frame shall be flooded through the network. When a node receives a frame with a valid sequence number, it retransmits the frame only if the `rtx_global_counter` value is greater than zero. The `rtx_global_counter` value is decremented by one in the RTx of a frame. With this, each RTx of a frame has the same `rtx_global_counter` value at a particular point in time, also in distant areas. On the one hand, this ensures that all transmitted waveforms are identical so that the CCTs can be decoded by the respective receivers. On the other hand, this also allows for implicit time synchronization between all nodes in the network when the initial value of `rtx_global_counter` is included in the MSDU, similar to Glossy [73].

4.2.1.3 *Local Repetition Counter*

The protocol field `rtx_rep_per_node` defines the maximum number of additional RTx repetitions of a flooding frame by each node after the first RTx of the frame, respectively. The `rtx_rep_per_node` value is kept unchanged in a frame during a flooding process. When a node receives a frame for the first time, i.e., when the sequence number is new, it initializes a local repetition counter with the `rtx_rep_per_node` value of the frame. After generating the first RTx of a received frame, a node may receive the frame again, e.g., as a CCT from its next-hop neighbors when they retransmit the frame. When a node receives a frame with the current sequence number, it retransmits the frame another time if the `rtx_global_counter` value in the frame and its local repetition counter are both greater than zero. In this process, the node decrements the `rtx_global_counter` value and its local repetition counter each by one. Thus, the first RTx repetition of a node typically occurs concurrently with the first RTx of its two-hop neighbors and has an `rtx_global_counter` value that is two less than the value transmitted during its first RTx of the respective frame. With this, the waveforms transmitted by all nodes are identical at each point in time, allowing for CCTs with CI. Still, different nodes may have different internal states with respect to their local repetition counters.

4.2.1.4 *Automatic Retry*

The protocol flag `auto_retry` indicates whether a node is allowed to automatically retry to reach its next-hop neighbors if it does not receive an RTx immediately after its own RTx or, in case of the initiator, the initial Tx of a flooding frame. When a node retries to reach its next-hop neighbors, it behaves similarly as if it had received an RTx from them. In particular, it sends a so-called retryRTx if its local repetition counter is greater than zero and if the `rtx_global_counter` value would have been greater than zero in the RTx frame that it had expected but that it did not receive from its next-hop neighbors. The node sends the retryRTx in the subsequent Tx slot, during which its two-hop neighbors would send an RTx if they received an RTx by the node's one-hop neighbors. Still, there is a chance that the next-hop neighbors of the node did send an RTx and that the node did not receive it correctly. Therefore, the `rtx_global_counter` value of the retryRTx is reduced by two with respect to the previous Tx or RTx of the node so that the retryRTx may contribute to a CCT with the two-hop neighbors of the node.

4.2.1.5 *Discussion*

The protocol field `rtx_rep_per_node` allows the initiator of a flooding process to tune the reliability through redundant CCTs of the same message, as introduced in Section 1.2.1 and as illustrated in Figure 4. Since the additional RTx repetitions may occur as CCTs in conjunction with other hops while the flooding frame propagates through the network, the additional costs in terms of channel utilization are limited.

Consider a network in which no other transmissions are allowed during a flooding process and in which the maximum number of hops of any node from the initiator is $n_{\text{hops}}^{\text{max}}$. If the initial `rtx_global_counter` value is smaller than $n_{\text{hops}}^{\text{max}}$, any number of additional RTx repetitions will not incur any additional costs in terms of channel utilization since the flooding process will terminate when the `rtx_global_counter` value reaches zero. In contrast, if the `rtx_global_counter` field is initialized with a larger value than $n_{\text{hops}}^{\text{max}}$, additional RTx repetitions may prolong the flooding process. More precisely, the number of Tx slots used during the propagation of a flooding process through the network is:

$$n_{\text{Tx}}^{\text{propagation}} = 1 + \min(\texttt{rtx\_global\_counter},\ n_{\text{hops}}^{\text{max}}) \tag{42}$$

Conversely, the number of Tx slots that are used for additional RTx repetitions at the end of a flooding process, i.e., after all nodes may have processed the frame once, is:

$$n_{\text{Tx}}^{\text{repetitions}} = \min(\max(0,\ \texttt{rtx\_global\_counter} - n_{\text{hops}}^{\text{max}}),\ 2 \cdot \texttt{rtx\_rep\_per\_node}) \tag{43}$$

For instance, if a network has a radius of 10 hops around the initiator and if the initiator sets the `rtx_global_counter` field to 10, the flooding process will comprise 11 Tx slots. Note that the RTx during the last Tx slot does not reach any new relay nodes, but it may still reach passive consumers of the message at the network border. If the initiator sets the `rtx_rep_per_node` field to 1, all nodes except for the last 2 hops will send 2 RTx frames, while the last 2 hops will send only 1 RTx frame. With this, the reliability can be increased in the area covered by the first 8 hops, while the total number of used Tx slots stays 11. If the initiator sets the `rtx_global_counter` field to 12 and the `rtx_rep_per_node` field to 1, $n_{\text{Tx}}^{\text{repetitions}}$ will be 2, i.e., there will be 2 Tx slots that are used by the last 2 hops for additional RTx repetitions. With this, the reliability can also be increased for the network area covered by the last 2 hops, at the expense of 18.2 % more channel utilization.

This example illustrates that the low-level MAC protocol empowers the initiator of a flooding process to tune the area coverage and the reliability in combination with the efficiency of the channel utilization. Also note that additional RTx repetitions incur additional costs in terms of energy expenditure of the nodes. Furthermore, the initiator of a flooding process can tune the latency per hop by partitioning or by aggregating the data inserted into the MSDU of each flooding frame and by adjusting the MCS appropriately. Overall, the initiator can tune certain performance tradeoffs to specific requirements by using the low-level MAC protocol and the ability of the WARP prototype to handle frames of different sizes. With this, the low-level MAC protocol facilitates the design of applications of the decentralized URLLC (dURLLC) paradigm introduced in Chapter 1. Moreover, the low-level MAC protocol is sufficiently lightweight to be processed in real-time by the WARP prototype, as further discussed in Section 4.3.

4.2.2 *MAC Header Formats*

The low-level MAC protocol of the WARP prototype is integrated into the IEEE 802.11 MAC header in order to allow also commercial devices to receive flooding frames from the WARP SDRs and to inject flooding frames into the network. The low-level MAC protocol is exposed through three different formats that enable different operation modes. A general overview of the IEEE 802.11 MAC header is provided in Section 2.4.7.

4.2.2.1 *Flooding Feed Format*

The flooding feed format is the main format for network flooding of the WARP prototype. This format is used to disseminate a message in a spatial area spanning a specified number of hops around the initiator. A flooding feed frame is a broadcast data frame, i.e., the Address 1 field of the IEEE 802.11 MAC header is set to `FF:FF:FF:FF:FF:FF`. Furthermore, the Address 2 and the Address 3 fields are both set to `B1:6F:10:0D:FE:ED`, allowing commercial devices to consume flooding frames through a MAC address filter. The low-level MAC protocol fields are encoded into the 16 bits of the Duration/ID field of the IEEE 802.11 MAC header since this field is normally set to zero and thus unused for non-QoS broadcast data frames [104, Section 9.3.2.1.5]. While the flooding feed format deviates from the IEEE 802.11 standard in this respect, commercial devices can receive such frames without any problems in tests. Figure 31a illustrates how the fields of the low-level MAC protocol are mapped to the 16 bits of the Duration/ID field.

4.2.2.2 *Frame Injection Format*

When an IEEE 802.11 MAC protocol data unit (MPDU) is passed to `pcap_inject()`, the driver of the attached network interface may overwrite certain MAC header fields before sending the frame [144]. Tests with two different Atheros AR928X wireless network adapters show that the Ath9k driver overwrites the Duration/ID field of a non-QoS broadcast data frame with zero. Therefore, commercial devices may not be able to use the flooding feed format to pass flooding frames to the WARP SDRs, unless their respective device driver or chip firmware is modified to prevent such alterations.

In order to overcome this limitation, the WARP prototype offers a format that enables commodity devices to inject data frames for flooding by encoding the low-level MAC protocol fields into the Address 2 field of the IEEE 802.11 MAC header. The Duration/ID field is not used by this format and may be set to zero. The Address 1 field denotes a broadcast data frame, i.e., it is set to `FF:FF:FF:FF:FF:FF`. The Address 2 field may range from `F4:57:F1:00:D0:00` to `F4:57:F1:00:DF:FF`, i.e., the first 36 bits are used as an identifier, whereas the last 12 bits encode certain fields of the low-level MAC protocol, as illustrated in Figure 31b. The Address 3 field is set to `B1:6F:10:0D:FE:ED`.

When a WARP SDR receives an injected frame, it increments its local sequence number and parses the fields of the low-level MAC protocol that are mapped into the Address 2 field of the IEEE 802.11 MAC header. If the `rtx_global_counter` value is greater than zero, it retransmits the frame. The RTx frame is encoded in the flooding feed format and has the new `sequence_number` value and a decremented `rtx_global_counter` value. When multiple WARP SDRs receive an injected frame for flooding, they synchronously update their local sequence numbers and send the RTx frame as a CCT with CI.

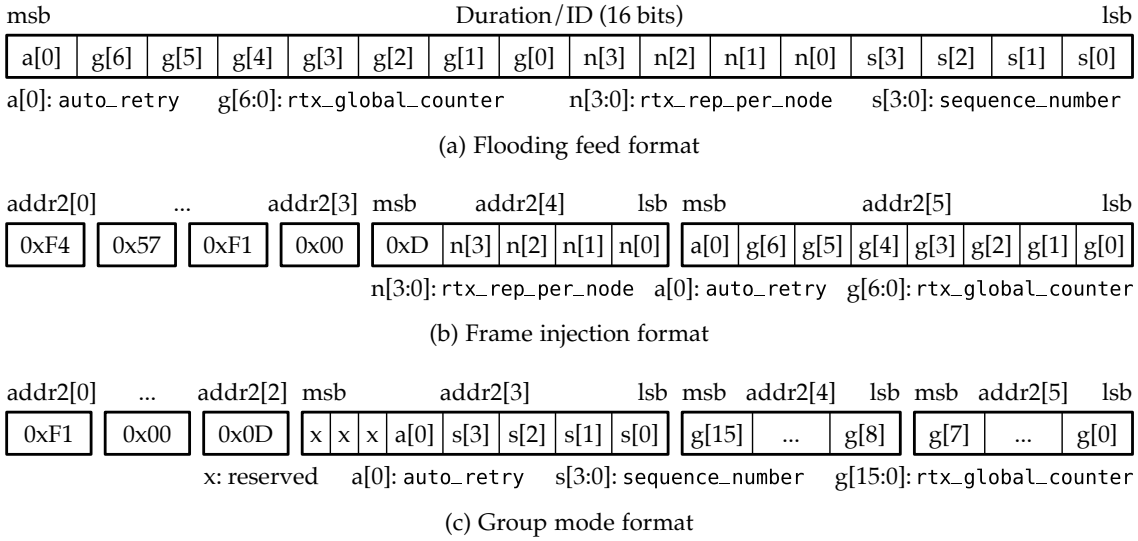


Figure 31: MAC header formats of the low-level MAC protocol

4.2.2.3 *Group Mode Format*

The WARP prototype offers a third MAC header format that changes the semantics of the low-level MAC protocol slightly in order to facilitate experiments in which a limited number of nodes can generate a large number of successive CCTs with CI. To this end, the nodes are partitioned into groups that repeatedly pass on a frame in a cyclical manner. In the simplest case, two groups of nodes pass on a frame in a ping-pong fashion.

Each node has a local configuration variable `group_count` that denotes the total number of groups in the network. All network nodes must set this variable to the same value. Further, each node has a local configuration variable `group_id` that defines its group membership and that may take values in the range $[0, \texttt{group\_count} - 1]$. When a node receives a frame in the group mode format, it checks if the `rtx_global_counter` value is greater than zero and if `(rtx_global_counter % group_count == group_id)` is true. If both conditions are met, the node retransmits the frame, potentially as a CCT with other nodes of its group. The `rtx_global_counter` value is decremented in the RTx so that another group is triggered to retransmit the frame further during the next turn. This procedure may continue until the `rtx_global_counter` value reaches zero.

The protocol field `rtx_rep_per_node` is not used in the group mode format since each node must retransmit the same frame up to $\lceil \texttt{rtx\_rep\_per\_node}/\texttt{group\_count} \rceil$ times. However, the protocol flag `auto_retry` can be set to allow each node to send one retryRTx if it does not receive an expected RTx from its next-hop neighbors after its own RTx.

In the group mode format, the Duration/ID field of the IEEE 802.11 MAC header is not used and may be set to zero. The Address 1 field is set to `FF:FF:FF:FF:FF:FF`. The Address 2 field may range from `F1,00,0D,00,00,00` to `F1,00,0D,FF,FF,FF`, i.e., the first 3 octets are used as an identifier, whereas the last 3 octets encode certain fields of the low-level MAC protocol. Figure 31c illustrates how the fields of the low-level MAC protocol are mapped to the Address 2 field of the IEEE 802.11 MAC header. Note that the protocol field `rtx_global_counter` has 16 bits, allowing for up to 65535 successive CCTs. The Address 3 field is set to `B1:6F:10:0D:FE:ED`. When a received frame in the group mode format triggers an RTx, the RTx is also encoded in the group mode format.

4.2.3 *User Interface*

The WARP prototype has a user interface that allows a user to manage configuration settings and internal variables of the low-level MAC protocol. It is exposed through both the UART interface and the three push buttons in the top left corner of the WARP v3 SDR. The user interface allows a user to cycle through a list of menu items either by pressing the *up* and *down* buttons on the WARP SDR or by pressing the *s* and *f* keys in a terminal connected to the UART. The menu item selection is shown on the two 7-segment displays and is printed via UART along with the corresponding value or setting. A menu item can be entered by pressing the *middle* button on the WARP SDR or by pressing the *d* key in a terminal connected to the UART. Upon entering a menu item, the corresponding value or setting is shown on the two 7-segment displays and may possibly be modified by pressing the *up* and *down* buttons or the *s* and *f* keys, respectively. The user interface returns to the menu item selection by pressing the *middle* button or the *d* key.

The user interface is implemented in CPU High and maintains all settings, except for the `sequence_number` and the `rtx_global_counter` values of the low-level MAC protocol, which are maintained and updated in real-time by CPU Low. CPU High communicates with CPU Low through a mailbox mechanism of the IEEE 802.11 reference design [247] in order to apply PHY settings in CPU Low, to retrieve values from CPU Low and to grant control over the 7-segment displays to CPU Low, depending on the respective menu item. The menu items are represented on the two 7-segment displays as follows upon selection, while they allow a user to perform the described actions upon entering, respectively:

- SE: displays and modifies the current `sequence_number` value in CPU Low.
- rH: displays the higher 8 bits of the last seen `rtx_global_counter` value as two hexadecimal digits. These bits are only used by frames in the group mode format.
- rL: displays the lower 8 bits of the last seen `rtx_global_counter` value as two hexadecimal digits. The lower 7 bits are used by all three MAC header formats.
- PO: displays and modifies the transmission power in the range of $[-9, 21]$ dBm.
- CH: displays and modifies the channel number, shown as two hexadecimal digits. The WARP v3 supports channels in the 2.4 GHz band and in the 5 GHz band.
- FS: displays and selects the PHY sampling rate from {10, 20, 40} MS/s. At 40 MS/s, the WARP prototype operates the Non-HT and HT-MF PPDUs for 20 MHz channel bandwidth at double clock speed.
- CO: displays and modifies the CFO containment factor in permille, represented as two hexadecimal digits. This feature is described in Section 4.2.4.
- An: displays and selects the employed antenna port. When A is selected, antenna port A is active. When b is selected, antenna port B is active. When 2 is selected, the WARP prototype listens through both antenna ports on the wireless channel and selects the antenna port with the better signal quality upon a frame reception. If an RTx is triggered, it is sent via the same antenna port selected during Rx.
- GC: displays and modifies the `group_count` value, which defines the total number of groups in the group mode, as described in Section 4.2.2.3.

- G 1: displays and modifies the `group_id` value, which defines the group membership of the respective node, as described in Section 4.2.2.3.
- db: enables or disables a debug mode that prints certain system state information into the MSDU of an RTx frame while processing the low-level MAC protocol. This feature is described in more detail in Section 4.2.6.
- AC: enables or disables application control over the hex displays by the IBSS implementation of the IEEE 802.11 reference design [247].
- LS: allows a user to load and store multiple configuration settings from and to the electrically erasable programmable read-only memory (EEPROM) of the WARP v3. The configuration settings of the following menu items are included when loading or storing: P0, CH, FS, CO, An, GC, G 1. When the LS menu item is entered, the decimal points of both 7-segment displays light up and a submenu can be accessed by pressing the *up* button, the *down* button, the *s* key or the *f* key. The user interface returns to the main menu either after completing a selection cycle in the submenu or after executing one of the submenu items. A selected submenu item is executed by pressing the *middle* button or the *d* key. The submenu has the following items:
  - Ld: loads the settings from the EEPROM and applies them.
  - Sb: stores the current settings to the EEPROM.
  - CL: clears the settings in the EEPROM by overwriting them with zeros.

  Upon execution of each of these submenu items, the columns of LEDs next to the two 7-segment displays blink green on success and red otherwise.

  Since the EEPROM is a non-volatile storage, the stored configuration settings persist also when the device is powered off. When starting up, the WARP prototype checks if configuration settings are stored in the EEPROM by means of a magic string that resides next to the settings. If the magic string is found, the WARP prototype automatically loads the configuration settings from the EEPROM and applies them, whereas it starts with default settings otherwise.

In addition, the WARP prototype indicates successful and erroneous flooding frame receptions by means of the two columns of green and red LEDs next to the 7-segment displays, respectively. In each column, one LED is illuminated at any point in time. The green LED is incremented upon reception of a flooding frame with a good FCS, whereas the red LED is incremented upon reception of a flooding frame with a bad FCS. In contrast to the original IEEE 802.11 reference design [247], the WARP prototype of this work indicates only receptions of flooding frames, i.e., other frame types are ignored.

4.2.4 *CFO Containment*

During a flooding process, each node that generates an RTx synchronizes in frequency to its respective reception of the flooding frame to ensure that CCTs are generated with CI. Hence, the frequency synchronization is performed several times by different nodes in succession, according to the network size and the initial `rtx_global_counter` value. However, each execution of the frequency synchronization may introduce a small error,

making the frequency of an RTx differ slightly from the frequency of the previous RTx. In particular, while the enhanced CFO estimation technique presented in Section 3.6 mitigates the impact of noise, it is still affected by noise according to Equation 40. Further, the enhanced CFO estimator in the PHY Rx core (see Section 4.1.1) and the frequency shifter in the PHY Tx core (see Section 4.1.2) both work with fixed-point computations, so the frequency of an RTx may be shifted marginally due to quantizations [9].

Since the carrier frequency of a flooding process may change slightly towards both higher and lower frequencies with each RTx, it may fluctuate around the carrier frequency of the initiator over time. However, experimental tests show that the frequency shift can also be biased towards any of both directions in some cases when the positions of the nodes and the frame length are not changed. Such a bias might be introduced by the quantization of fixed-point numbers under recurring experimental conditions. Therefore, the CFO of a flooding process with respect to the specified center frequency may potentially drift into a particular direction. If this effect makes the CFO grow too large, it can eventually lead to frame loss and thus to the abortion of a flooding process. Note that the IEEE 802.11 standard requires the center frequency of a 20 MHz channel to be met with an accuracy better than ±20 parts per million (ppm) [104, Section 17.3.9.5], i.e., the CFO of a flooding process should stay within ∼±50 kHz in the 2.4 GHz band.

To prevent the CFO of a flooding process from drifting widely, the WARP prototype has a CFO containment mechanism that enables distributed nodes to steer the CFO towards ∼0 Hz without explicit coordination. To this end, each node reduces its CFO estimate $\Delta\hat{f}$ by a small CFO containment factor v before generating an RTx:

$$\Delta\hat{f}_{\mathrm{v}} = \Delta\hat{f} \cdot (1 - v), \quad 0 \leq v \ll 1 \tag{44}$$

Since various WARP v3 SDRs have CFOs better than 3 kHz with respect to each other, as presented in Section 3.5.1, they usually have similar CFO estimates of a received frame. Thus, if the CFO of a flooding frame is relatively small, all nodes obtain small CFO estimates, so that the frequency change through the CFO containment is negligible. In contrast, if a flooding frame has a relatively large CFO, each node reduces its CFO estimate by a certain amount that is similar to the amounts applied by the other nodes. Therefore, the nodes cooperatively pull the CFO of a flooding process towards ∼0 Hz either when the CFO tends to grow towards positive or negative values or when a frame with a relatively large CFO is injected for flooding by a commercial device. The CFO containment mechanism is validated in testbed experiments presented in Section 4.4.3.

4.2.5 *Statistics Log*

To observe the temporal performance of the WARP prototype with a large number of successive CCTs by means of a limited number of nodes, the testbed experiments in Section 4.5 are conducted in group mode (see Section 4.2.2.3). In each of these experiments, a WARP SDR with WARPLab [250] triggers a flooding process and records the received signal of the CCTs, allowing various performance metrics to be computed for each CCT. However, the signal of a received CCT does not reveal how many and which transmitters participated in the generation of the CCT since all transmitters sent the same signal while being synchronized in time and frequency.

In order to gain more insights into the performance of each individual node during a flooding process in group mode, the WARP prototype has a logging facility that runs in real-time on CPU Low, allowing for subsequent computation of statistical metrics. The logging facility keeps track of the frames received and retransmitted during a flooding process by means of a bitmap in which each bit indicates whether a flooding frame with a particular `rtx_global_counter` value was processed by the individual node. With this lightweight data structure, a node can write the log with minimal overhead during the real-time processing of a received flooding frame as follows.

When a flooding frame with a new sequence number is received, the logging facility initializes two variables with the received `rtx_global_counter` value, one of which is kept unchanged during the flooding process, whereas the other one is updated with each received and retransmitted frame. Upon the reception of a frame, the logging facility sets the bit corresponding to the `rtx_global_counter` value of the received frame to one as well as the bit corresponding to the `rtx_global_counter` value of its RTx if an RTx is generated according to the low-level MAC protocol. In addition, it clears any bits representing `rtx_global_counter` values that are larger than the value of the currently received frame and that are smaller than the value of the previously marked frame. With this, the resetting of bits is carried out incrementally through several function calls, avoiding to reset the entire bitmap during the reception of the first frame, which would potentially violate a timing constraint. In addition to recording processing events of Rx and RTx frames, the logging facility also counts the number of generated retryRTx frames. After a flooding process, the logging facility provides the following statistical metrics:

- The **frame reception rate (FRR)** is the fraction of Rx frames received by the node over the total number of Rx frames of its group. An Rx frame of a group is a flooding frame with an `rtx_global_counter` value that cannot be contained in an RTx frame generated by a node of the group.
- The **frame transmission rate (FTR)** is the fraction of RTx frames generated by the node, without retryRTx frames, over the total number of RTx frames of its group. An RTx frame of a group is a flooding frame with an `rtx_global_counter` value that may be contained in an RTx frame generated by a node of the group.
- The **frame transmission rate of group (FTR-G)** is the fraction of observed RTx frames retransmitted by any nodes within the group of the node, including the node itself, over the total number of RTx frames of the group.

Note that a bit corresponding to an RTx frame may be set through two different events. First, a node sets the bit when a received frame triggers the node to generate an RTx. Second, a node sets the bit also when it overhears an RTx of other nodes in its group after it missed the reception of the frame that would have triggered it to generate an RTx. The logging facility is able to distinguish between these two cases by means of the bit corresponding to the next higher `rtx_global_counter` value. If this bit is set, the logging facility treats the RTx as generated by the node itself, i.e., the RTx is considered for the calculation of both the FTR and the FTR-G. Otherwise, the logging facility treats the RTx as generated by other nodes within the same group, i.e., the RTx is considered for the calculation of the FTR-G only. Also note that the FTR-G reflects the observations of a particular node and may differ between the nodes of a group. The metrics can be retrieved by pressing the *g* key in a terminal connected to the UART.

4.2.6 *Debug Outputs*

The WARP prototype can expose runtime information on two debug outputs to support the validation of the correct interaction between software and hardware.

Besides logical correctness, another important aspect of the WARP prototype is the adherence to timing constraints. The WARP prototype employs the MAC support core of the IEEE 802.11 reference design [247] (see Figure 28) to generate an RTx of a received flooding frame with accurate timing in relation to certain Rx and Tx events. To this end, the generation of an RTx is configured within the `frame_receive()` function upon the reception of a flooding frame and is eventually triggered through a timer running in the MAC support core. Thus, the execution of `frame_receive()` must reach certain code sections in a limited amount of time, as described in Section 4.3. These timing constraints can be checked by means of the control and status register of the MAC support core, which can be read by software through `wlan_mac_get_tx_ctrl_status()` [247].

After reading the control and status register, its contents can be printed to a terminal via UART in order to inspect the internal state of the MAC support core at the point in time when the register was read. However, printing debug messages via UART at runtime inevitably delays the software execution process, so only one timing constraint can be checked at a time, while the code execution must be considered as retarded afterwards. Thus, the timing constraints are checked one by one during the development process, while each trigger frame is transmitted by a WARP SDR with WARPLab [250].

In addition to meeting the timing constraints during the generation of an RTx in response to a single trigger frame, the timing constraints must also be met during the generation of successive RTx frames in real-time, as required by the group mode or when the `rtx_rep_per_node` value is greater than zero. In particular, the execution of `frame_receive()` must not be delayed by any other tasks so that it starts timely when the PHY Rx core detects a frame reception. Further, the execution of `frame_receive()` must finish in time so that the next frame can be received. For instance, if the logging facility (see Section 4.2.5) did not reset its bitmap incrementally but all at once upon the reception of the first frame of a flooding process, it could potentially still run when the next frame arrives. In this case, a node would not be able to generate the next RTx.

To facilitate the inspection of the internal state of a node during real-time processing, the WARP prototype has a debug mode that can be enabled through the user interface, as described in Section 4.2.3. By enabling the debug mode, the WARP prototype writes readings of its internal state at certain points in time into specific locations of the MSDU. In contrast to printing via UART, this method delays the code execution just marginally since it requires only a few additional memory accesses. To avoid interference between nodes with different internal states, the debug mode should only be enabled when each hop comprises a single node, e.g., with a single node per group in group mode. A monitor device can capture the successive RTx frames of a real-time flooding process, decode them, extract the values written into the MSDUs by the respective nodes and process them further to perform certain checks and to augment evaluation metrics.

Section 4.4 presents experiments conducted in debug mode. In these experiments, a WARP SDR with WARPLab [250] triggers ping-pong transmissions between two nodes in group mode and records the successive RTx frames. Each frame generated as a retryRTx is marked by the debug mode and can thus be highlighted in the plots.

4.3 Real-Time MAC Processing

The WARP prototype processes a received frame in real-time and decides according to the low-level MAC protocol whether to generate an RTx immediately after the reception. In this process, the software and hardware components of the WARP prototype run in parallel and must be geared to each other. On one side, the PHY Rx core starts to decode a frame directly upon its detection on a listening RF interface and writes the contents of the OFDM symbols byte by byte to a dedicated Rx buffer. On the other side, CPU Low starts to process the available portions of the MPDU while the reception is still ongoing. In particular, CPU Low parses the IEEE 802.11 MAC header to determine if the frame is a valid flooding frame and if an RTx should be generated. While the WARP prototype prepares an RTx in software, it employs the MAC support core of the IEEE 802.11 reference design [247] to trigger the execution of an RTx by means of a hardware timer, allowing it to achieve accurate time synchronization with respect to a received frame. Thus, CPU Low must also reach certain code sections in a limited amount of time, e.g., before the running hardware timers expire.

This section describes the real-time processing of a received flooding frame with the low-level MAC protocol and highlights the underlying timing constraints. Section 4.3.1 describes the detection and processing of an incoming frame by the PHY Rx core. Section 4.3.2 describes how CPU Low processes the MPDU provided by the PHY Rx core and how it generates an RTx. Section 4.3.3 describes how CPU Low generates a retryRTx if the `auto_retry` flag is set in the flooding frame. Section 4.3.4 describes how CPU Low deals with errors during the frame reception process, e.g., due to a bad FCS. While some timing-related requirements apply equally for all frames, the severity of certain timing constraints depends on the duration or the data rate of the flooding frame. Thus, to ascertain that the WARP prototype works for all MSDU lengths and for all MCS indices, the timing constraints must be checked under the conditions of two edge cases. Section 4.3.5 discusses the edge case with the lowest possible latency per hop, which is 44.2 µs and which is achievable with the Non-HT PHY at MCS 5 or higher (see Table 2). Section 4.3.6 discusses the edge case with the highest PHY data rate that is provided by the WARP prototype, which is 65 Mbit/s with the HT PHY at MCS 7 [104, Section 19.5].

Figure 32 illustrates the alignment of the software functions that are executed during the generation of an RTx and of a retryRTx with the waveform of the respective frames. The waveform is generated by means of a WARP SDR with WARPLab [250] that records ping-pong transmissions between two nodes close to each other, as shown in Figure 33. The considered node first receives a flooding frame from the other node, which in turn triggers the considered node to generate an RTx after a SIFS. Since the considered node does not overhear an RTx from the other node after its own RTx and since the `auto_retry` flag is set in the frame, the considered node additionally generates a retryRTx. The frame is a Non-HT PPDU at MCS 5 that encapsulates an MSDU of 1 byte, which results in a frame duration of 28 µs. Note that this example also reflects the edge case with the lowest possible latency per hop of 44.2 µs that is further discussed in Section 4.3.5.

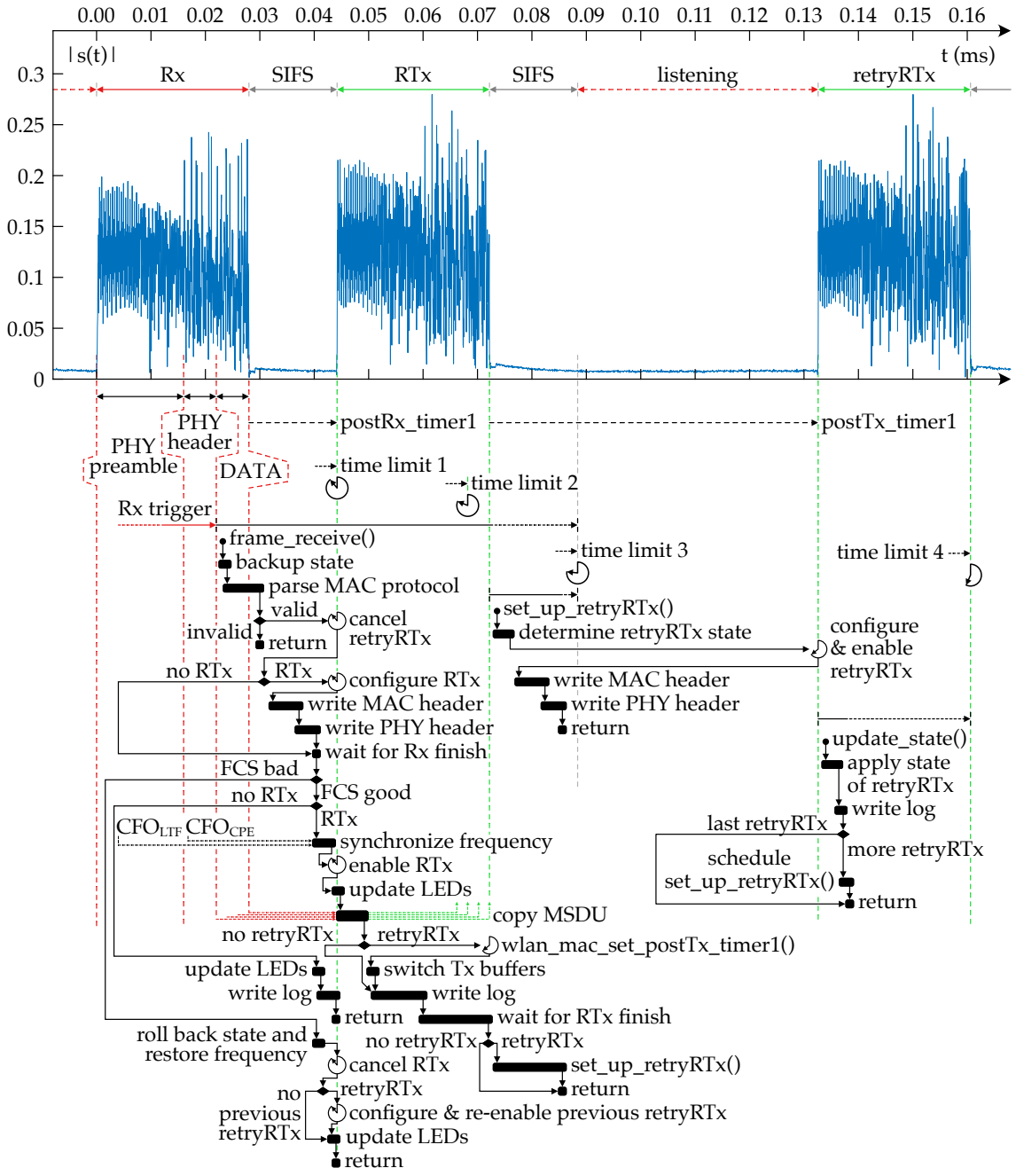


Figure 32: Adherence to timing constraints by the software functions: The waveform exemplarily shows that the considered node receives a flooding frame, generates an RTx, does not receive an RTx while listening and generates a retryRTx. Time limit 1 refers to enabling the RTx or re-enabling a previous retryRTx. Time limit 2 refers to copying the MSDU. Time limit 3 refers to returning from `set_up_retryRTx()` and from `frame_receive()`. Time limit 4 refers to returning from `update_state()`. The illustrated latency per hop of 44.2 µs can be achieved with a Non-HT PPDU at MCS 5 or higher (see Table 2).

4.3.1 *Rx Processing by the PHY Rx Core*

When the WARP prototype is idle, it continuously listens on one or both of its RF interfaces for incoming frames, according to the antenna selection of the user interface, as described in Section 4.2.3. Upon the arrival of a frame, the PHY Rx core of the IEEE 802.11 reference design [247] detects the PHY preamble, as described in Section 2.4.3, and initiates a frame reception process. In this process, the PHY Rx core estimates the symbol timing of the incoming frame, as described in Section 2.4.3, and performs a CFO estimation and correction with respect to the incoming frame by means of the L-LTF, as described in Section 2.4.4. In addition, the PHY Rx core uses the L-LTF to estimate the channel state information (CSI) of the incoming frame, as described in Section 2.4.5. Further, the PHY Rx core detects the PHY format and, in case of a Non-HT PPDU, retrieves the MCS and the length of the frame from the legacy SIGNAL field (L-SIG), allowing it to decode the remainder of the PPDU (see Section 2.4.2). At this point in time, the enhanced CFO estimator in the PHY Rx core starts to run, as described in Section 4.1.1. After initializing the descrambler by means of the SERVICE field (see Section 2.4.2) and after checking the validity of the PHY header, the PHY Rx core decodes and writes the PHY service data unit (PSDU) byte by byte to a dedicated Rx buffer. Furthermore, it sets a flag in a memory-mapped status register that indicates that an Rx process is ongoing, besides other flags about its processing state. Additionally, it exposes the number of PSDU bytes written to the Rx buffer through another memory-mapped register [247].

4.3.2 *Rx Processing by CPU Low and Generation of RTx*

Up to this point, CPU Low continuously polls the aforementioned status register of the PHY Rx core to check if an Rx process has started. As soon as the PHY Rx core indicates that a frame is received and that the PHY header is valid, CPU Low collects several pieces of information from the PHY and executes the `frame_receive()` function [247], which carries out the low-level MAC protocol of the WARP prototype, as illustrated in Figure 32. Since the `frame_receive()` function must meet certain timing constraints, its start must not be significantly delayed by other tasks running on CPU Low, which is particularly considered in the design of the functions related to the generation of a retryRTx.

The flowchart of the `frame_receive()` function in Figure 32 shows that CPU Low first creates a backup of its state variables, which allows CPU Low to restore this state if the frame turns out to be invalid during the reception process, e.g., due to a bad FCS. Then, it parses the low-level MAC protocol, which may be realized in one of three different formats mapped into the IEEE 802.11 MAC header, as described in Section 4.2.2. To this end, CPU Low has to wait until the PHY Rx core has written the Address 2 field of the IEEE 802.11 MAC header, which is the last field that is parsed or checked for each of the three MAC header formats. Note that the MAC header formats are designed to proceed quickly at this point, i.e., the waiting time would be longer if some of the fields of the low-level MAC protocol were mapped into IEEE 802.11 MAC header fields that arrive later, such as the Sequence Control field (see Section 2.4.7).

If the received frame appears to be a valid flooding frame according to its MAC header, the WARP prototype cancels a potentially pending retryRTx, since a retryRTx should only be executed when an expected RTx from the one-hop neighbors is not received.

If an RTx should be generated according to the low-level MAC protocol, CPU Low configures the state machine B of the MAC support core to generate a transmission from a specified Tx buffer with certain PHY parameters as soon as `postRx_timer1` expires, but it does not enable the state machine yet. Then, CPU Low writes the MAC header of the Tx buffer, e.g., with a decremented `rtx_global_counter` value, and the PHY header. Figure 32 additionally shows that `postRx_timer1` is configured to run automatically for the duration of a SIFS after the end of the Rx frame, like for the generation of an acknowledgment (ACK) frame in the original IEEE 802.11 reference design [247].

At this point, CPU Low waits until the Rx process of the PHY Rx core finishes so that it can check by means of the PHY Rx core if the FCS of the incoming frame is correct. In case of a very short frame, the Rx process of the PHY Rx core may already be done, so that CPU Low can proceed immediately, as illustrated in the example in Figure 32. In case of a longer frame, however, CPU Low has to wait an according amount of time. Then, CPU Low retrieves the result of the FCS check from the PHY Rx core.

If the FCS is correct and if an RTx should be generated, CPU Low synchronizes the frequency of the PHY Tx core to that of the Rx frame. Note that the enhanced CFO estimation in the PHY Rx core, which is presented in Section 4.1.1, has just finished. CPU Low reads the L-LTF-based CFO estimate and the CPE-based estimate of the residual CFO from the registers in the PHY Rx core. Then, it applies the CFO containment procedure described in Section 4.2.4 and writes the resulting values to the corresponding registers of the frequency shifter in the PHY Tx core, as presented in Section 4.1.2.

Next, CPU Low enables the state machine B of the MAC support core to trigger the generation of the configured RTx when `postRx_timer1` expires. Since this code section must be reached before `postRx_timer1` expires, it marks the first time limit during the execution of `frame_receive()`, as shown in Figure 32. In case of the shortest possible frame duration, `postRx_timer1` expires briefly afterwards, so that the PHY Tx core instantly starts to transmit the PHY preamble of the RTx. In the meanwhile, CPU Low updates the LEDs to indicate the successful frame reception. In addition, it possibly also updates the contents of the 7-segment displays, as described in Section 4.2.3.

Since the Tx buffer for the RTx only holds the PHY and MAC headers at this point, CPU Low further proceeds with copying the MSDU from the Rx buffer to the Tx buffer. While this copy operation terminates quickly for short frames, as illustrated in Figure 32, it takes much longer for frames with a larger MSDU size. With this, the MSDU bytes are written to the Tx buffer right before they are read by the PHY Tx core for transmission. Consequently, the copy operation must proceed at least as fast as the PHY Tx core reads data from the Tx buffer i.e., the copying must finish before the transmission of the RTx, which is marked as the second time limit in Figure 32. Note that the PHY Tx core and the PHY Rx core use disjoint buffers in fixed memory locations, so the PHY Tx core cannot simply transmit the contents of an Rx buffer.

If the `auto_retry` flag is set in the frame, CPU Low further configures a retryRTx, as described in the subsequent Section 4.3.3. Otherwise, CPU Low only writes the log, as described in Section 4.2.5. In any case, the `frame_receive()` function must return within a SIFS duration after the generated RTx so that the node is ready to receive an RTx from its one-hop neighbors. This condition is marked as the third time limit in Figure 32.

4.3.3 *Generation of retryRTx*

If a retryRTx should be generated according to the low-level MAC protocol, CPU Low configures `postTx_timer1` of the MAC support core after copying the MSDU. As a result, `postTx_timer1` runs for the duration of a SIFS plus the duration of the current flooding frame plus the duration of another SIFS after the end of the ongoing RTx. With this, `postTx_timer1` expires exactly when the two-hop neighbors of the considered node should start their reception-triggered RTx. The timer calibration is validated in a testbed experiment presented in Section 4.4.2.

Furthermore, CPU Low switches its Tx buffers, i.e., it makes the buffer of the ongoing RTx the secondary buffer that is subsequently used to generate a retryRTx and vice versa. With this, the preparation of the next reception-triggered RTx does not modify the buffer of the retryRTx so that the canceled retryRTx can be re-enabled if the preparation of the next reception-triggered RTx is aborted, as further described in Section 4.3.4.

Next, CPU Low writes the log, as described in Section 4.2.5, and waits for the ongoing RTx to finish. Then, it executes `set_up_retryRTx()` as a function call of `frame_receive()`. In this function, CPU Low determines the `rtx_global_counter` value of the retryRTx as the latest `rtx_global_counter` value reduced by two, as described in Section 4.2.1.4. Further, CPU Low configures state machine B of the MAC support core to generate a transmission from the secondary Tx buffer with certain PHY parameters as soon as `postTx_timer1` expires and enables the state machine. Then, it writes the MAC header of the secondary Tx buffer, according to the determined protocol fields of the pending retryRTx, and the PHY header. After executing `set_up_retryRTx()`, CPU Low must still have enough time in `frame_receive()` to return within a SIFS duration after the RTx. As stated in Section 4.3.2, this condition is marked as the third time limit in Figure 32. It ensures that the node can subsequently receive an RTx from its one-hop neighbors.

During the next Tx slot, which starts directly a SIFS duration after the generated RTx, the considered node typically receives an RTx from its one-hop neighbors. In this case, CPU Low proceeds exactly as for the frame that it received just before the generation of the RTx, i.e., it cancels the configured retryRTx and processes the received frame. However, sometimes a node might not receive an RTx from its one-hop neighbors, as described in Section 4.2.1.4. In this case, the configured retryRTx is transmitted and may contribute to a CCT with CI with the two-hop neighbors of the node.

If the transmission of the retryRTx is triggered, CPU Low executes `update_state()`. With this, it reduces its local `rtx_global_counter` value by two and decrements its local repetition counter, as described in Section 4.2.1.4 and in Section 4.2.1.3, respectively. If both counters permit, CPU Low schedules another execution of `set_up_retryRTx()` right after the ongoing retryRTx. Thus, CPU Low must return from `update_state()` before the ongoing retryRTx finishes, which is marked as the fourth time limit in Figure 32. If another execution of `set_up_retryRTx()` is scheduled, it is executed in the context of the run loop of CPU Low. Hence, the checks before executing `set_up_retryRTx()` must be performed efficiently in order to avoid delaying the execution of other functions, as mentioned in the beginning of Section 4.3.2.

4.3.4 *Error Handling*

CPU Low modifies its local state variables immediately upon the reception of a frame, i.e., before the FCS is received, which allows it to reach the code section enabling the RTx before `postRx_timer1` expires, which marks the first time limit shown in Figure 32. However, a received frame may turn out to be corrupted when the FCS arrives, which would leave the node in a bad state if not properly handled. Consequently, CPU Low creates a temporary backup of its local state variables upon executing `frame_receive()`. If the FCS turns out to be bad, CPU Low rolls back the state variables and restores the previous frequency synchronization by means of this backup. Further, if a retryRTx was configured and canceled beforehand, CPU Low re-enables it. Since CPU Low employs two different Tx buffers, the Tx buffer of the retryRTx was not modified while CPU Low wrote the MAC header for the unexecuted RTx. The retryRTx must be re-enabled before `postTx_timer1` expires, which as well corresponds to the first time limit in Figure 32. However, in this case, CPU Low does not necessarily have to return from `frame_receive()` before `postTx_timer1` expires since a retryRTx follows and since the `update_state()` function still has some clearance with respect to the fourth time limit shown in Figure 32.

4.3.5 *Critical Path for the Lowest Latency per Hop*

To meet the first time limit shown in Figure 32, CPU Low must enable state machine B of the MAC support core before the relevant timer expires. If the FCS of a received flooding frame is good, CPU Low must enable the RTx before `postRx_timer1` expires. If the FCS of a received flooding frame is bad and if a retryRTx is configured, CPU Low must re-enable the temporarily canceled retryRTx before `postTx_timer1` expires.

Figure 32 shows the processing of a frame with the lowest possible latency per hop, which is 44.2 µs and which can be achieved with a Non-HT PPDU at MCS 5 or higher, resulting in a PPDU duration of 28 µs (see Section 2.4.2 and Table 2). This edge case is most critical for reaching the first time limit. In particular, the code section in which CPU Low waits for the PHY Rx core to finish is executed after the end of the received frame in this case since the DATA field of the PPDU is shorter than the execution time of `frame_receive()` up to this code section. Therefore, the critical software execution paths for the first time limit range from the beginning of the `frame_receive()` function to the points at which state machine B is enabled for an RTx or re-enabled for a retryRTx, respectively. The resulting timing constraints of the RTx and the retryRTx are checked by means of the debug outputs described in Section 4.2.6, respectively.

4.3.6 *Critical Path for the Highest Throughput*

To meet the second time limit shown in Figure 32, CPU Low must copy the MSDU bytes from the Rx buffer to the Tx buffer before the PHY Tx core reads them for transmission. This copy procedure starts shortly before or after the PHY Tx core begins to transmit the PHY preamble of the RTx, depending on the frame duration, and must finish before the PHY Tx core reads the bytes for the last OFDM symbol. This software execution path is checked with short and long frames at the highest possible PHY data rate of the WARP prototype, which is 65 Mbit/s with the HT PHY at MCS 7 [104, Section 19.5].

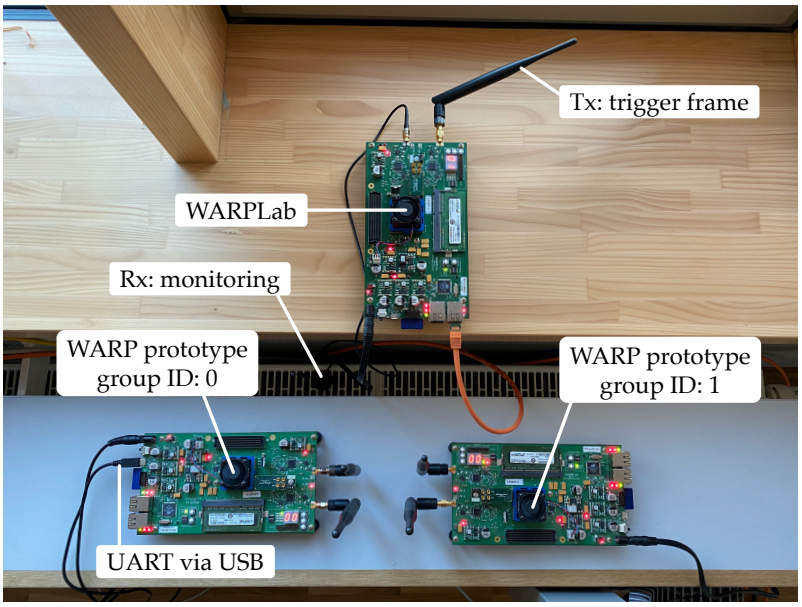


Figure 33: Testbed setup for ping-pong transmissions between two WARP prototype nodes

4.4 Functional Validation

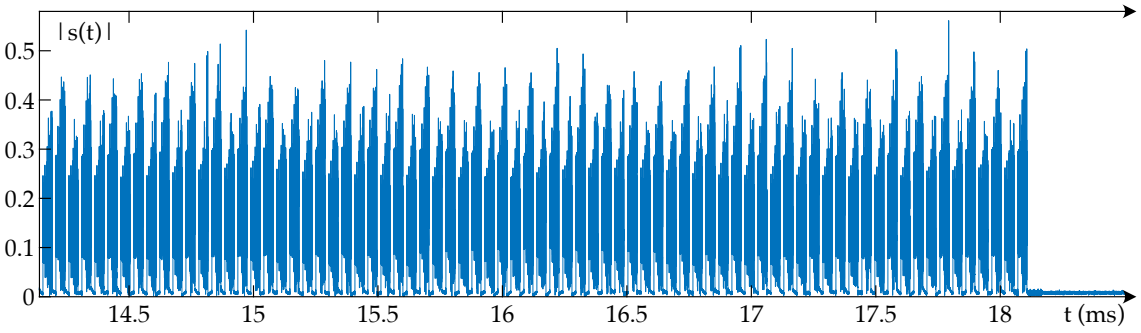
To ensure that the WARP prototype functions properly, testbed experiments are conducted with ping-pong transmissions between two nodes. Figure 33 shows the testbed setup. A third WARP SDR that runs WARPLab [250] is connected to a Matlab workspace and triggers flooding processes in group mode between the two WARP prototype nodes. In addition, the WARP SDR with WARPLab monitors each triggered flooding process and transfers the received IQ samples to the Matlab workspace, where they are processed. The WARP prototype nodes are operated in debug mode, as described in Section 4.2.6, which allows the identification of frames that are generated as a retryRTx.

The experiments are conducted with an initial `rtx_global_counter` value of 10 000. While flooding over 10 000 hops might not be meaningful in practice, this value is applied here to study the temporal performance of the WARP prototype.

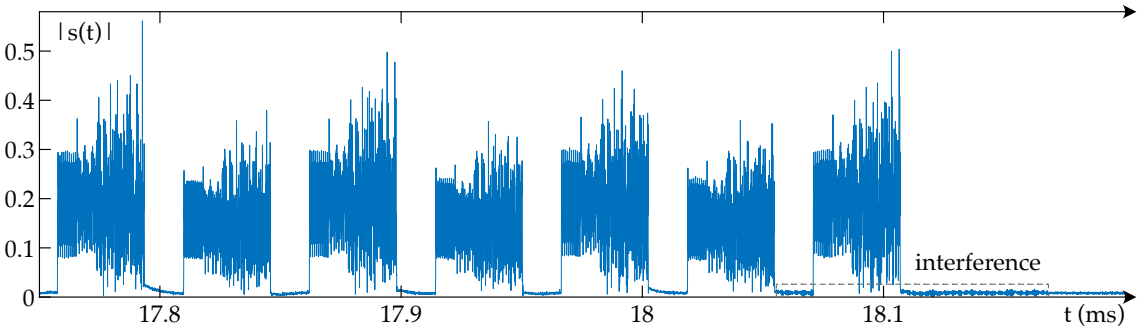
4.4.1 *Overcoming Sporadic Interference*

The OFDM-based IEEE 802.11 PHYs are susceptible to interference (see Section 2.1.2), so sporadic interference from other devices may distort a flooding process and may even lead to an interruption if all nodes are affected, e.g., in a sparse network.

Figure 34 shows the magnitude of IQ samples during a flooding process in group mode. The experiment is conducted with a Non-HT PPDU at MCS 3 (see Table 2), encapsulating an MSDU of 1 byte. Furthermore, the `auto_retry` flag is unset (see Section 4.2.1.4). Figure 34a shows that the RTx train consistently persists for 18 ms and suddenly stops. Figure 34b shows the segment around the last frames and reveals that interference sets in during the idle period between the second last RTx and the last RTx. Even though the magnitude of the interference is much lower than that of the frame transmissions, it suffices to distort the last frame reception and to interrupt the flooding process.



(a) The RTx train is unexpectedly interrupted after the 346th RTx.



(b) A detailed view of the waveform reveals that the last RTx is affected by interference.

Figure 34: Ping-pong transmissions between two nodes with the `auto_retry` flag unset (disabled)

Given that the experiments presented in this chapter are conducted on channel 14, which requires a license to operate IEEE 802.11 networks in Germany, the interference in Figure 34 is most probably caused by an external device that also uses the 2.4 GHz Industrial, Scientific and Medical (ISM) band. In particular, Bluetooth devices are found to cause detrimental interference as shown in Figure 34 and are therefore disabled during experiments whenever possible. Still, foreign devices may cause interference.

Figure 35 shows the magnitude of the waveform of another experiment in which the `auto_retry` flag is set. This experiment is conducted with a Non-HT PPDU at MCS 0 and with an MSDU size of 1 byte. The RTx train of this experiment is uninterrupted, i.e., it contains all 10 000 frames. Figure 35 shows a segment of the RTx train in which the frame at 694.75 ms is a retryRTx according to its debug information. The previous RTx from the other node is subject to interference, which the retryRTx helps to overcome.

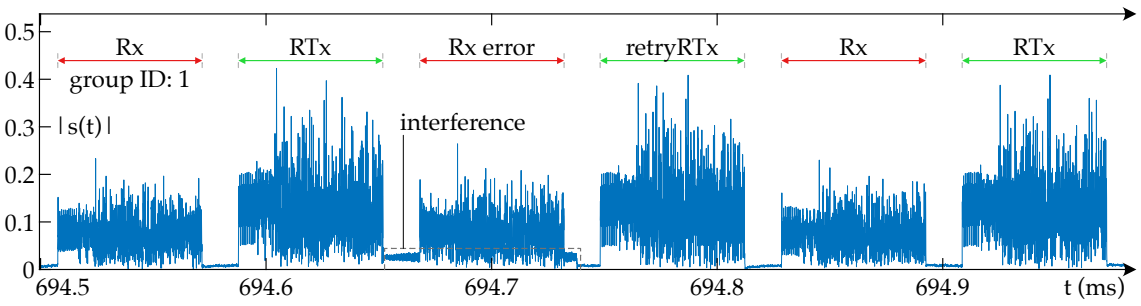


Figure 35: Ping-pong transmissions between two nodes with the `auto_retry` flag set (enabled)

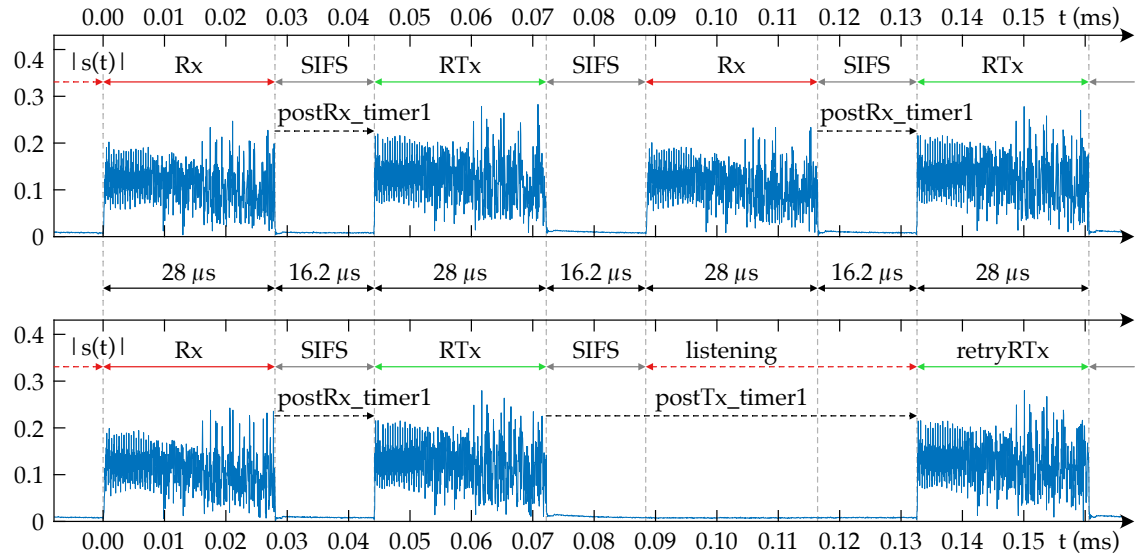


Figure 36: The retryRTx timing is calibrated with sample accuracy with respect to the RTx timing.

4.4.2 *Timing Calibration*

The WARP prototype triggers the generation of a configured RTx through `postRx_timer1`, as described in Section 4.3.2. The timer is configured to run for the duration of a SIFS after a frame reception with the calibration of the original IEEE 802.11 reference design [247]. In addition, the WARP prototype can generate a retryRTx by means of `postTx_timer1`, as described in Section 4.3.3. As a retryRTx should be synchronous in time with an RTx by the two-hop neighbors of the considered node, `postTx_timer1` runs for a SIFS duration plus the PPDU duration of the flooding frame plus another SIFS duration after an RTx.

Figure 36 illustrates that the calibration of `postTx_timer1` achieves sample-accurate timing in the testbed shown in Figure 33, i.e., a retryRTx is generated with the same timing as a second reception-triggered RTx. Still, at larger distances between nodes, the timing of a reception-triggered RTx changes slightly due to the additional propagation delay, whereas the timing of a retryRTx is fixed. Further, the measured SIFS duration is 16.2 µs, which slightly exceeds the 16 µs defined by the IEEE 802.11 standard (see Section 2.4.8). This uncritical deviation might be caused by a small propagation delay and by the frequency shifter inserted into the signal path of the PHY Tx core (see Section 4.1.2).

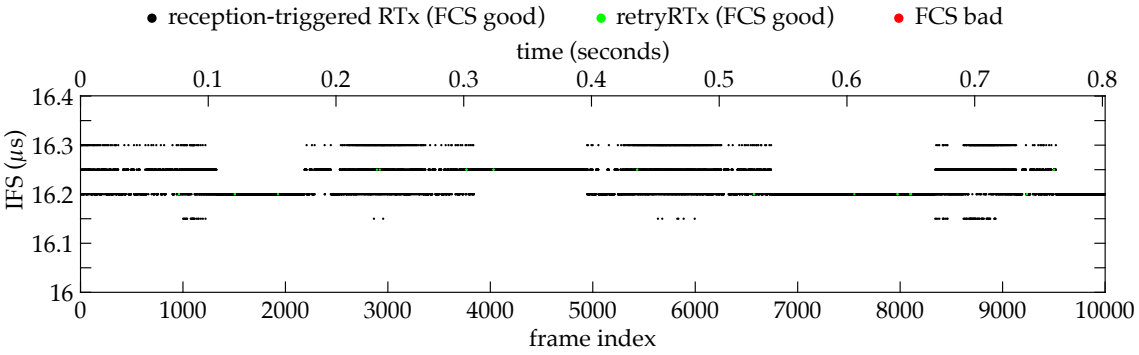


Figure 37: The measured IFS is consistent for 10 000 successive ping-pong transmissions.

Figure 37 illustrates the measured interframe space (IFS) between 10 000 successive ping-pong transmissions during a testbed experiment with the setup shown in Figure 33. The experiment is conducted with a Non-HT PPDU at MCS 0 that encapsulates an MSDU of 1 byte and that has the `auto_retry` flag set. With this, the experiment lasts ∼0.8 s. The IQ samples received by the WARP SDR with WARPLab are processed by a decoder that is based on the Matlab WLAN toolbox [150]. The decoder measures the IFS between two frames as the time difference between their detected start of frame (SOF) positions minus the PPDU duration. Further, the decoder identifies retryRTx frames by means of the embedded debug information, as described in Section 4.2.6, allowing these frames to be highlighted in the plot. Figure 37 shows that the IFS measurements of RTx frames consistently range between 16.15 µs and 16.3 µs, while the IFS measurements of retryRTx frames are within this range, which confirms that `postTx_timer1` is properly calibrated. The IFS resolution of 50 ns corresponds to the employed sampling rate of 20 MS/s.

Note that each occurrence of a retryRTx indicates a preceding reception error of a WARP prototype node, possibly caused by sporadic interference from a foreign device. While there are 25 retryRTx frames from the WARP prototype nodes, the Matlab decoder recovers 9 997 out of 10 000 frames with a correct FCS, indicating a superior performance of the Viterbi algorithm with soft-decision decoding [158, Section 12.3].

4.4.3 *CFO Containment*

The CFO containment mechanism presented in Section 4.2.4 is validated in various testbed experiments to identify a suitable CFO containment factor. On the one hand, the CFO containment factor should be large enough to contain the CFO when it tends to drift into one particular direction. On the other hand, the CFO containment factor should still be small to alter the estimated CFO just marginally.

Figure 38 shows the progression of the CFO of the previous experiment, the measured IFS of which is shown in Figure 37. The CFO containment is turned off in this experiment. The CFO is measured in Matlab by means of the enhanced CFO estimation technique presented in Section 3.6 with respect to the monitoring WARP SDR running WARPLab. Figure 38 shows that the CFO oscillates in both directions between −1 kHz and 17 kHz. In other runs, the CFO also takes more negative values and stays within ±50 kHz.

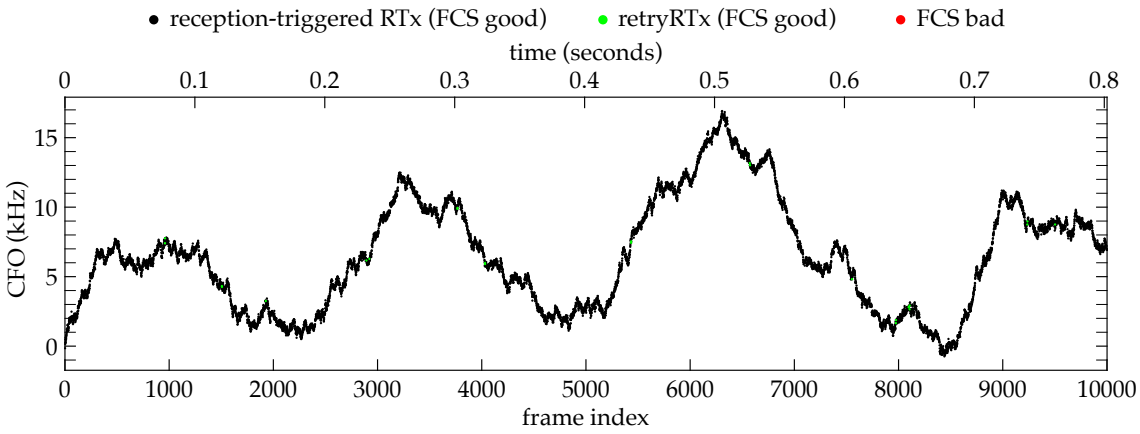
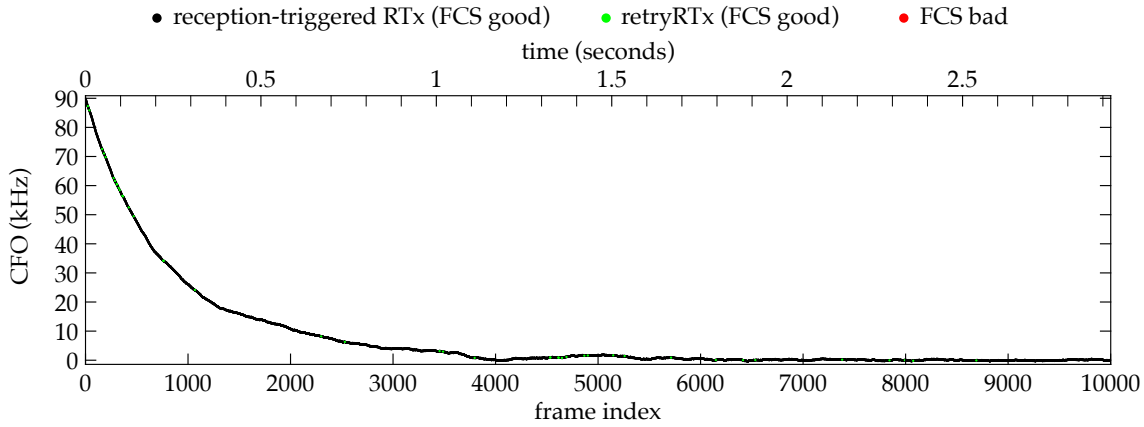
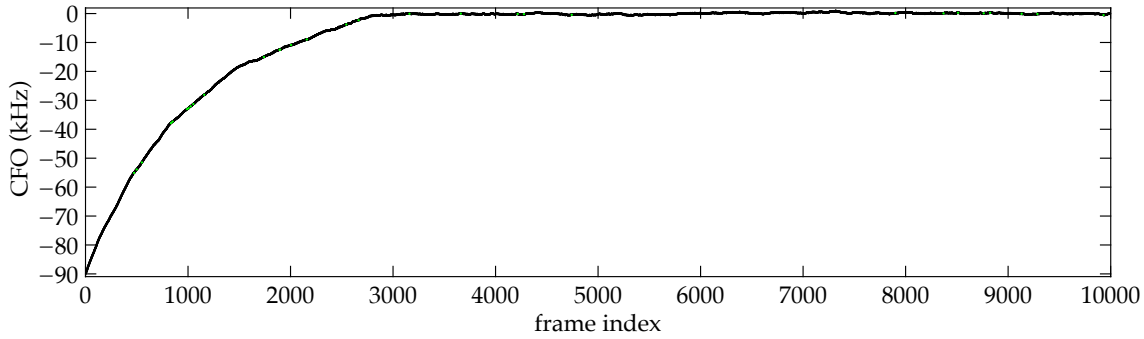


Figure 38: CFO progression of ping-pong transmissions without CFO containment



(a) Artificial CFO of trigger frame: 90 kHz



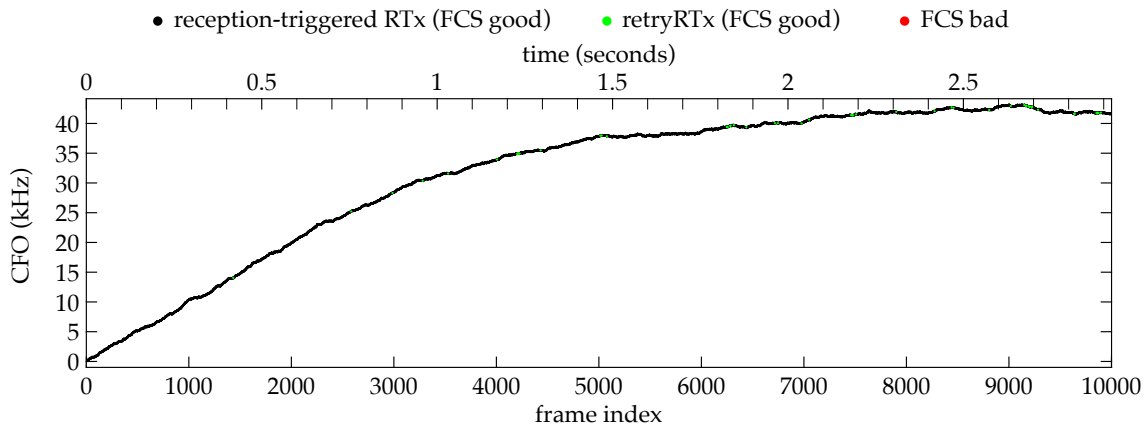
(b) Artificial CFO of trigger frame: −90 kHz

Figure 39: A CFO containment factor of $v = 0.001$ may suffice to pull the CFO towards ~ 0 Hz.

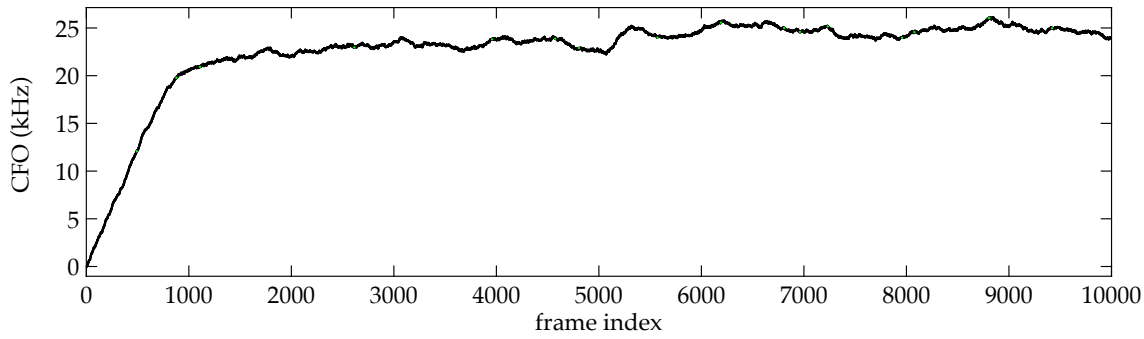
Figure 38 is obtained with a Non-HT PPDU at MCS 0 that contains an MSDU of 1 byte, resulting in a PPDU duration of 64 µs. With this, the PPDU contains 12 OFDM symbols that can be utilized by the CPE-based estimator of the residual CFO in the PHY Rx core. Note that the enhanced CFO estimator becomes more accurate with a higher number of OFDM symbols according to Equation 40. Thus, in ping-pong experiments with a longer PPDU, there are less short-term variations of the CFO. However, the CFO may still oscillate in ranges comparable to the ones shown in Figure 38. Further, with slightly different node positions, the CFO may also drift into one particular direction.

The CFO containment mechanism is assessed in another experiment in which the trigger frame has an artificial CFO to emulate a frame injection from a detuned device. This experiment employs a Non-HT PPDU at MCS 0 that contains an MSDU of 161 bytes, resulting in a PPDU duration of 276 µs. With this, the enhanced CFO estimator can leverage 65 OFDM symbols. The nodes are positioned as shown in Figure 33, such that the CFO oscillates randomly in test runs, i.e., it does not drift into one particular direction. The CFO containment factor is set to $v = 0.001$, taking effect as expressed in Equation 44. Figure 39 shows that the CFO is pulled towards ~ 0 Hz in two different runs in which the artificial CFO of the trigger frame is set to 90 kHz and to -90 kHz, respectively.

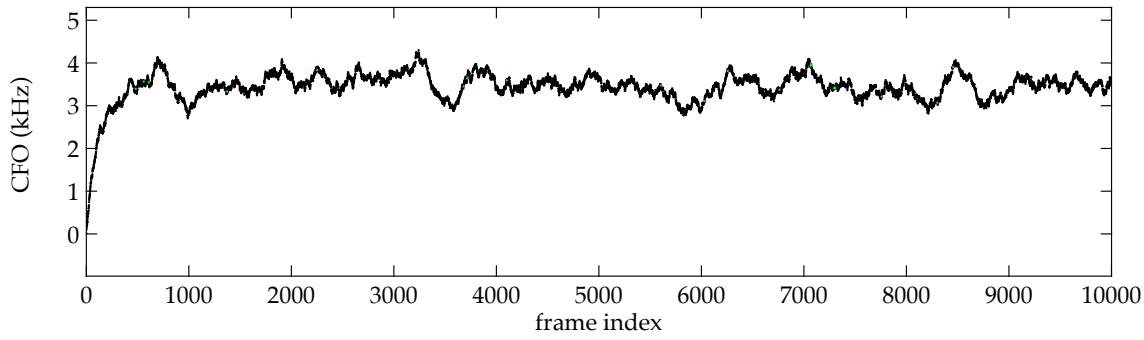
In another experiment, the nodes are slightly moved such that the CFO drifts towards higher frequencies. In some test runs, the CFO exceeds 100 kHz and causes frame loss. Figure 40 shows the effects of different CFO containment factors. The setting $v = 0.01$ provides good performance and is thus selected for the experiments in Section 4.5.



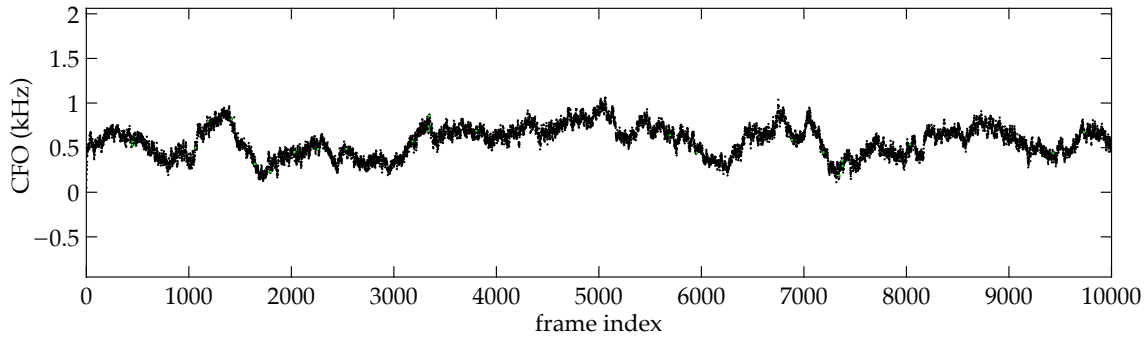
(a) CFO containment factor: $v = 0$ (off)



(b) CFO containment factor: $v = 0.001$



(c) CFO containment factor: $v = 0.01$



(d) CFO containment factor: $v = 0.1$

Figure 40: Comparison of CFO containment factors when the CFO drifts towards higher values: the CFO containment factor allows balancing the tradeoff between the frequency synchronization accuracy and a small distance of the CFO progression from ~ 0 Hz. A CFO containment factor of $v = 0.01$ provides a suitable compromise.

4.4.4 *Frequency Synchronization*

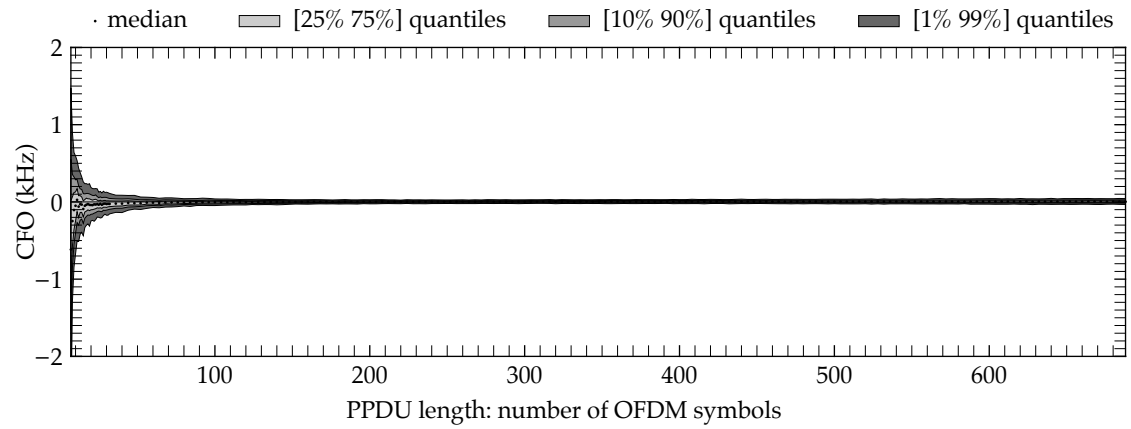
The frequency synchronization mechanism of the WARP prototype, which consists of the enhanced CFO estimator in the PHY Rx core (see Section 4.1.1) and the frequency shifter in the PHY Tx core (see Section 4.1.2), is assessed in another testbed experiment. The goal is to determine the characteristics of these two components in combination and to identify parameter regions that are most suitable to achieve CI during a CCT. In particular, the accuracy of the frequency synchronization is assessed as a function of the PPDU length in terms of the number of OFDM symbols since the accuracy of the enhanced CFO estimator in the PHY Rx core depends on it, as expressed by Equation 40. Further, the accuracy of the frequency synchronization is also assessed in different CFO regimes with respect to the specified center frequency.

Contrary to the testbed setup shown in Figure 33, the experiment is conducted with a WARP SDR running WARPLab [250] and a single WARP prototype node as a DUT. A Matlab script generates trigger frames by means of the Matlab WLAN toolbox [150] and uses the WARP SDR with WARPLab to transmit them and to record the corresponding RTx frames from the WARP prototype node. In this process, the PPDU length is swept from 7 to 32 OFDM symbols in steps of 1 OFDM symbol and then from 36 to 688 OFDM symbols in steps of 4 OFDM symbols, including the PHY preamble with an equivalent length of 4 OFDM symbols (see Section 2.4.2). To this end, MCS 5 is employed for a PPDU length of up to 15 OFDM symbols and MCS 0 for any larger PPDUs, while each number of OFDM symbols is translated to the corresponding maximum MSDU size. Note that the enhanced CFO estimator is not affected by the MCS change as the pilot subcarriers invariably employ binary phase-shift keying (BPSK) [104, Section 17.3.5.10]. For each PPDU length, 1 000 trigger frames are transmitted in close succession.

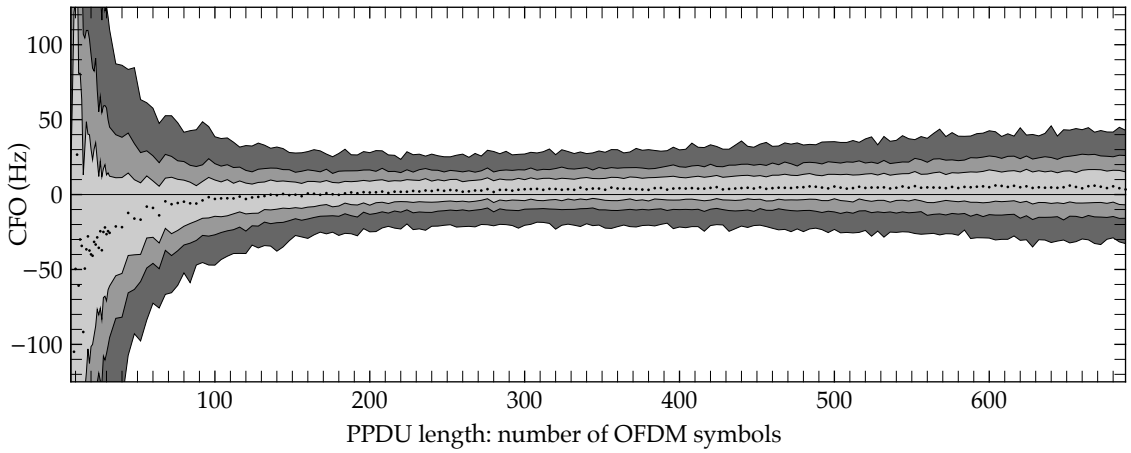
The RTx frames received from the WARP prototype node, in turn, are processed with the assistance of the Matlab WLAN toolbox [150] and by means of the enhanced CFO estimation technique presented in Section 3.6. Since the WARP prototype node synchronizes in frequency with respect to each trigger frame before generating an RTx, the CFO of an RTx with respect to its trigger frame should ideally be close to 0 Hz. Further, since each RTx is generated a SIFS duration after the respective trigger frame, the oscillator frequencies of both nodes should be stable during each measurement, i.e., they may change just slightly due to jitter, if at all (see Section 2.2.2).

Figure 41 shows the characteristics of the CFO as a function of the PPDU length in terms of the number of OFDM symbols. Figure 41a reveals that the CFO exceeds ± 1 kHz in a few cases with very short PPDU lengths, whereas it declines sharply for increasing PPDU lengths, confirming that the enhanced CFO estimator in the PHY Rx core is effective. Figure 41b provides a more detailed view of low absolute CFO values and shows that the CFO is close to 0 Hz for a wide range of PPDU lengths, indicating a high accuracy.

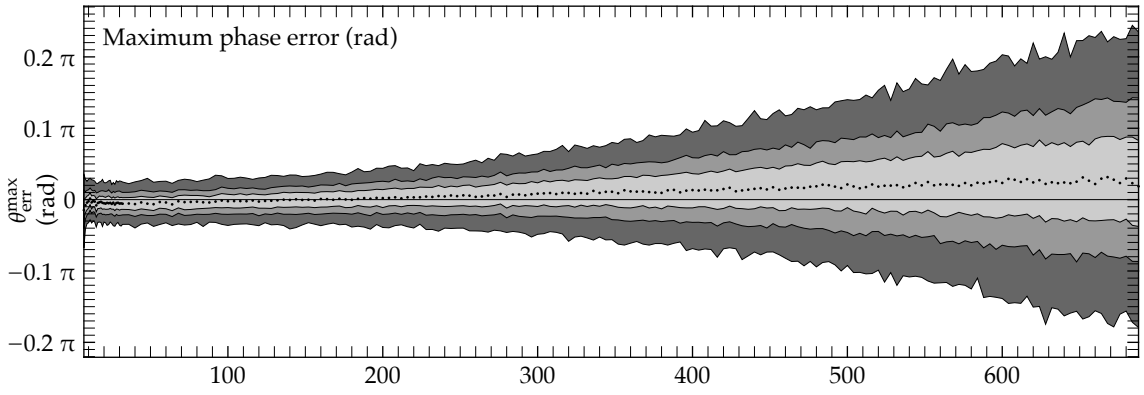
While deviations of the CFO from 0 Hz can generally be attributed to noise and interference, the enhanced CFO estimation in Matlab causes a small additional error. Further, Figure 41b shows that the CFO has a negative bias for short frames and a positive bias for long frames, which might be caused by quantizations of the signal processing algorithms running on the FPGA in a static environment. Moreover, the absolute CFO has a minimum for PPDU lengths of around 220 OFDM symbols and slightly increases for larger PPDU lengths, which might be caused by phase jitter of the two nodes.



(a) CFO of an RTx with respect to its trigger frame: full scale



(b) CFO of an RTx with respect to its trigger frame: detailed view



(c) CFO-induced phase error of an RTx at the end of the PPDU

Figure 41: Characteristics of the frequency synchronization mechanism of the WARP prototype

Figure 41c shows the maximum phase error caused by the CFO of an RTx, which is calculated as the measured CFO multiplied by the PPDU duration after the PHY preamble. This result reveals that CI is best achieved with PPDUs of up to $\sim$250 OFDM symbols. Note that a higher absolute CFO and a shorter PPDU duration compensate one another. In CFO regimes of $\pm$60 kHz, the characteristics are almost identical to those in Figure 41.

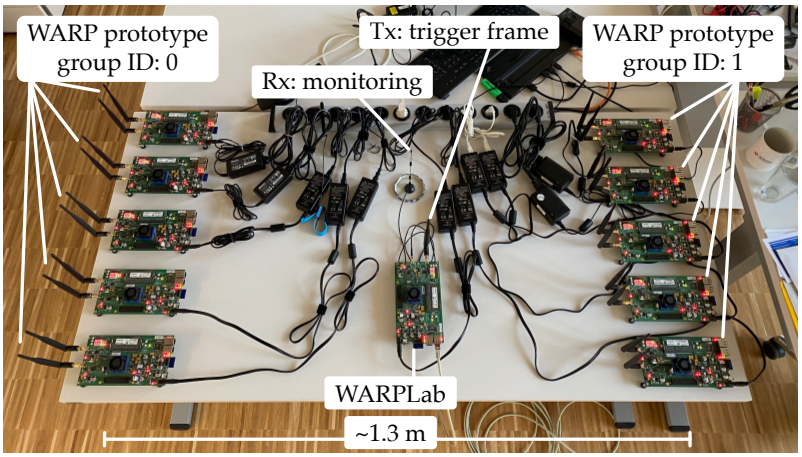


Figure 42: Testbed setup for ping-pong transmissions between two groups of five nodes

4.5 Evaluation

To assess the feasibility of CCT-based network flooding with IEEE 802.11 in practice, testbed experiments are conducted in which multiple WARP prototype nodes generate successive CCTs with CI by means of the group mode (see Section 4.2.2.3). A key question of these experiments is whether the synchronization in time and frequency keeps up over time since each CCT triggers the generation of the respective next CCT and since each node may introduce a small synchronization error when contributing to a CCT, so that a synchronization error may possibly propagate and grow over multiple hops. In order to achieve statistical significance and to observe the temporal progression of certain performance metrics, the trigger frame of each flooding process is initialized with an `rtx_global_counter` value of 10 000 (see Section 4.2.1.2). While the `auto_retry` flag is set to overcome interference by foreign devices (see Section 4.2.1.4 and Section 4.4.1), this feature does not mitigate synchronization errors as each node generates a retryRTx with the same time and frequency reference as the previous RTx. The CFO containment factor is set to $v = 0.01$ on all WARP prototype nodes (see Section 4.2.4 and Section 4.4.3). Further, the debug mode is disabled on all WARP prototype nodes (see Section 4.2.6) to make sure that all nodes generate the same waveform when contributing to a CCT. Each flooding process in group mode is triggered and monitored by a WARP SDR with WARPLab [250], which facilitates further analysis of the received CCT signals in Matlab. The performance of each individual WARP prototype node is evaluated by means of the logging facility (see Section 4.2.5). In addition, commercial devices with Atheros AR928X wireless network adapters are set up in monitor mode in the same environment to assess their ability to receive the CCTs by means of Wireshark and `libpcap` [241, 242].

4.5.1 *2 Groups: 1 vs. 5 Transmitters*

Figure 42 shows the testbed setup of an experiment with ping-pong transmissions between two groups of five nodes each. This experiment is designed to compare certain performance metrics between the cases when only one node per group generates an RTx at a time and when up to five nodes per group generate each RTx as a CCT with CI.

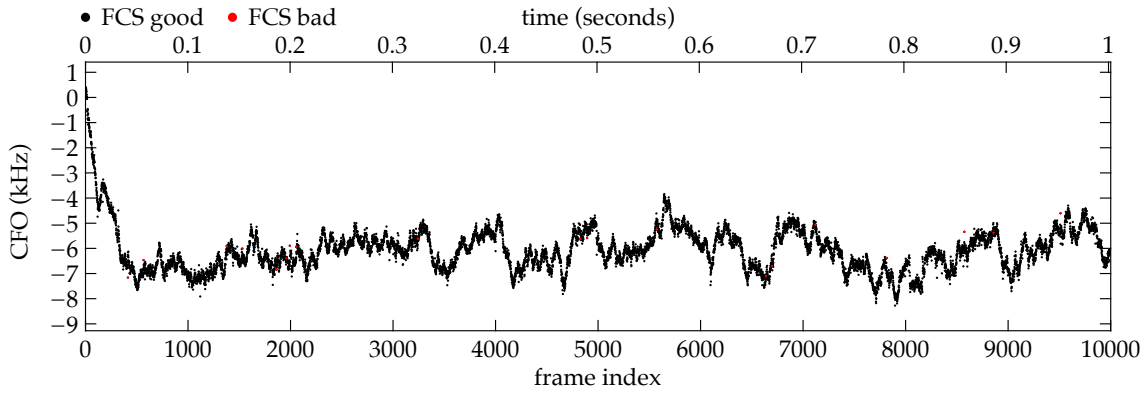
The WARP SDR with WARPLab [250] is used to trigger ping-pong transmissions with 10 000 successive CCTs and to record the received CCT signals for analysis in Matlab. The flooding frame is a Non-HT PPDU that encapsulates an MSDU with 161 bytes, while the MCS is swept from 0 to 7 in steps of 1 (see Table 2) in successive experiment runs. Four Acer notebooks and four Asus Eee PC netbooks with Atheros AR928X wireless network adapters are set up in monitor mode. A device with the ID *Acer-03* is placed on the upper table in Figure 42 at a distance of ~ 1.5 m from the WARP SDR with WARPLab, whereas the other seven commercial devices are placed at a distance of $\sim 2 - 3$ m from the WARP prototype nodes in partial LOS, all next to each other.

The experiment is first conducted with one WARP prototype node in each group, i.e., only one particular node of each group is turned on while the other nodes are turned off. Table 12 in Appendix A lists the FRRs of both WARP prototype nodes, showing that each node receives at least 99.6 % of the frames from the other node in all experiment runs. When a node misses a frame reception, it generates a retryRTx during its next Tx slot. With this, each experiment run finishes after 10 000 successive frame transmissions.

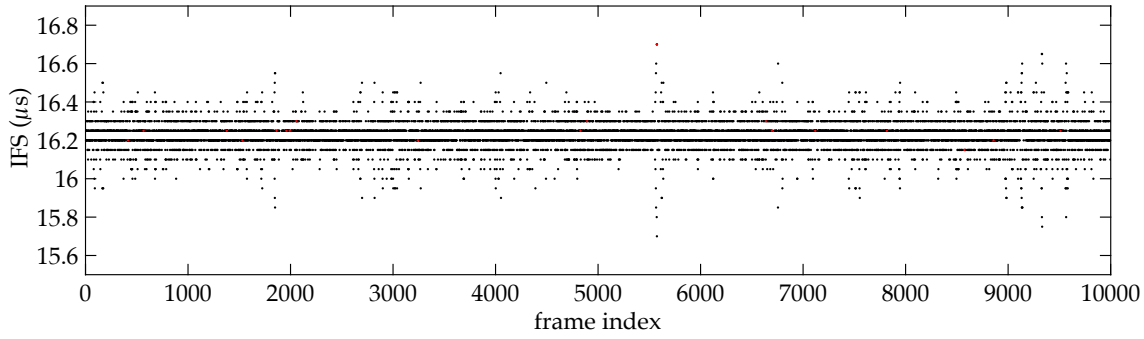
Table 13 in Appendix A lists the FRRs of the Matlab decoder that is based on the Matlab WLAN toolbox [150] and that processes the IQ samples from the WARP SDR with WARPLab as well as the FRRs of the eight commercial devices in monitor mode. These FRRs are calculated separately for each group ID and for a reception from any of both groups. The latter metric emulates a flooding scenario in which a device is located in between two groups of nodes that successively forward a flooding frame, so that the device has two chances to receive the same flooding frame. The Matlab decoder has an excellent reception performance, which can be attributed to both a good signal quality in proximity to the nodes and to its use of the Viterbi algorithm with soft-decision decoding [158, Section 12.3]. From the commercial devices, Acer-03 often has a good reception performance, which can be attributed to its proximity to the nodes. The reception performance of the other commercial devices varies, especially with an increasing MCS, despite their seemingly similar conditions with respect to their locations. Notably, EeePC-02 often has a high FRR for group ID 0 but a low FRR for group ID 1. Such variations may occur in practical settings due to fading effects.

The experiment is conducted another time with all WARP prototype nodes turned on, so that each RTx of the flooding frame is generated as a CCT with CI by up to five nodes. Table 14 in Appendix A lists the FRRs of the individual WARP prototype nodes, showing that the FRRs are good but reduced in comparison to the FRRs in Table 12 with only one active node in each group. Due to the redundancy of nodes in each group and with the help of retryRTx frames, each experiment run finishes after 10 000 successive CCTs.

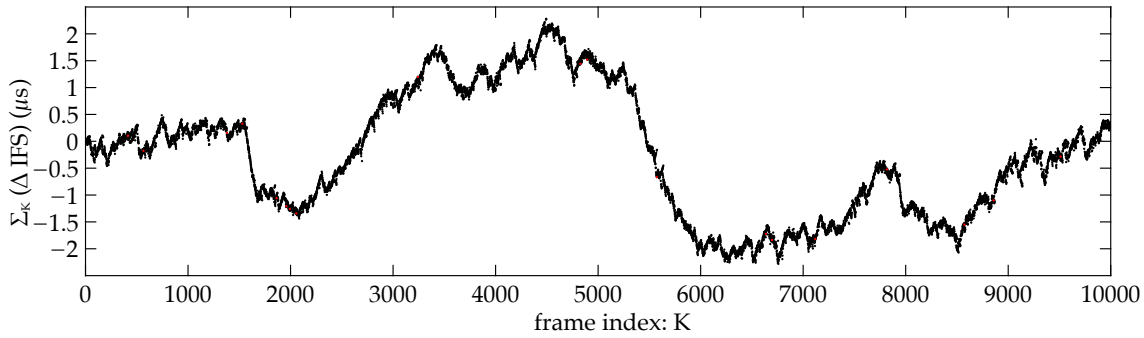
Table 15 in Appendix A lists the FRRs of the Matlab decoder and of the commercial devices in monitor mode, showing that there is not a single device whose reception performance is substantially worse than in the experiment with one node in each group. In fact, Acer-03, Acer-06, Acer-07, EeePC-02, EeePC-08 and EeePC-12 often have better FRRs in Table 15 than in Table 13, indicating that these devices benefit from the additional diversity of the CCTs and that the time and frequency synchronization of the WARP prototype nodes is sufficiently accurate over a long period to make such gains possible. The FRRs of the Matlab decoder are very good and are consistently better than the FRRs of the individual WARP prototype nodes, indicating that the reception performance of the WARP prototype could be further enhanced through better decoding algorithms.



(a) Measured CFO of CCTs, obtained with a CFO containment factor of $v = 0.01$



(b) Measured IFS of CCTs



(c) Accumulation of Δ IFS: global time synchronization error with respect to the flooding initiator

Figure 43: Progression of performance metrics in 5x5 ping-pong experiment at MCS 4 over 1.0 s

The signals received by the WARP SDR with WARPLab are further processed in Matlab to compute performance metrics of the CCTs and to visualize their temporal progressions. With one node per group, the measured CFO has similar characteristics as in Figure 40c. Further, the measured IFS typically ranges from 16.2 µs to 16.3 µs, with a few outliers at 16.15 µs and at 16.35 µs, i.e., its characteristics resemble those in Figure 37.

With five nodes per group, the Matlab decoder processes the received signals as usual and determines a single SOF and a single CFO estimate for each CCT frame. Figure 43 shows the progressions of the CFO and of the IFS, exemplarily for MCS 4. Figure 43a reveals that the CFO tends to drift towards negative frequencies in this experiment and that it is effectively contained at around ~ -6.5 kHz. There are only a few rather small outliers near the curve. Overall, the CFO progresses quite steadily, indicating that the nodes successfully keep up the joint frequency synchronization.

Figure 43b shows that the measured IFS ranges from 16.1 µs to 16.35 µs in most cases, while there are some outliers that are largely received correctly by the Matlab decoder. More precisely, 2.2 % of the detected CCT frames in this experiment run have an IFS of less than 16.1 µs, while 2.1 % of the detected CCT frames have an IFS of more than 16.35 µs. These outliers indicate that some nodes have a temporary time offset (TO) of up to a few hundred nanoseconds with respect to each other and that the SOF detector of the Matlab decoder locks to the signal components of such nodes when processing successive CCTs, so that the TOs become evident in the IFS measurements. Still, the characteristics in terms of the density and the deflection of the outliers is quite consistent throughout the entire experiment run, indicating that the WARP prototype nodes recover from such conditions by resynchronizing in time with respect to each other.

The mean IFS is ~ 16.223 µs in the experiment run with one node per group at MCS 4, whereas it is ~ 16.207 µs in the experiment run with five nodes per group at MCS 4. The fact that the mean IFS is slightly smaller in the experiment with CCTs indicates that the WARP prototype nodes tend to lock their detected SOFs to CCT signal components that arrive early, which mitigates intersymbol interference (ISI) since the other signal components become covered by the guard interval (GI) with a cyclic prefix (CP) [243].

Figure 43c shows the deviation of the expected SOF from the measured SOF for each CCT frame of the experiment run at MCS 4. By denoting the expected average IFS as $\widetilde{T}^{\mathrm{IFS}}$ and the measured IFS before the k-th RTx as T_k^{IFS}, the SOF deviation of the K-th RTx is:

$$\sum\nolimits_K (\Delta\,\mathrm{IFS}) \;=\; \sum_{k=1}^{K} \left(\widetilde{T}^{\mathrm{IFS}} - T_k^{\mathrm{IFS}} \right) \tag{45}$$

This metric is useful to express the error of a global time synchronization mechanism. Note that CCT-based network flooding facilitates global time synchronization of all nodes with respect to the initiator of a flooding process, as demonstrated with Glossy [73]. This principle works also with the WARP prototype when the initiator of a flooding process includes the initial `rtx_global_counter` value in the MSDU of the flooding frame. With this, each network node can determine the number K of elapsed Tx slots in the network by subtracting the received `rtx_global_counter` value from the initial value. Consequently, each node can estimate the TO $\Delta t(K)$ between the SOF of the received RTx and the SOF of the Tx of the initial flooding frame as:

$$\Delta\hat{t}(K) \;=\; K \cdot \left(T^{\mathrm{PPDU}} + \widetilde{T}^{\mathrm{IFS}} \right) \tag{46}$$

Figure 43c shows that the estimated TO $\Delta\hat{t}(K)$ is accurate to about ± 2.3 µs over all 10 000 CCTs in the experiment run at MCS 4. The largest error of $\Delta\hat{t}(K)$ in the experiment with five nodes per group is ± 5.3 µs, which occurs in the experiment run with MCS 6. Note that the accuracy of the global time synchronization mechanism depends on the accuracy of the measured average IFS $\widetilde{T}^{\mathrm{IFS}}$ and on the variability of the IFS T_k^{IFS} over different hops, which in turn depends on the propagation delay at each hop and thus on the network topology. Hence, the global time synchronization mechanism might be less accurate in practical network flooding over multiple hops with a variable node density than in the ping-pong experiment with a static per-hop topology.

4.5.2 *Meeting the URLLC Requirement*

The 3rd Generation Partnership Project (3GPP) defines the requirement of Ultra-Reliable Low-Latency Communications (URLLC) as the end-to-end delivery of a 32-byte frame with a reliability of $1 - 10^{-5}$ and with a latency of 1 ms [3]. Since this thesis proposes dURLLC as a similar paradigm for decentralized networks in Chapter 1, this section explores the feasibility of meeting the URLLC requirement with CCT-based ping-pong transmissions with WARP prototype nodes. Note that the reliability of a flooding process can be tuned through repetitive CCTs of the same flooding frame, as described in Section 1.2.1 and as illustrated in Figure 4. While the low-level MAC protocol provides the `rtx_rep_per_node` field to tune the number of RTx repetitions in multi-hop settings, as described in Section 4.2.1.3, the tuning of the reliability can be emulated by means of ping-pong transmissions by combining the receptions of a certain number of successive CCTs and by treating them like RTx repetitions of one particular message.

Another experiment with two groups of five WARP prototype nodes each is conducted, as illustrated in Figure 42. Each flooding process is triggered and monitored by a WARP SDR with WARPLab [250] placed in LOS in proximity to the WARP prototype nodes. In addition, a second monitoring WARP SDR with WARPLab is placed in a neighboring office at 6 m distance, i.e., behind a concrete wall in strict non-line-of-sight (NLOS). The MSDU of the flooding frame has 32 bytes, according to the URLLC requirement. The experiment is conducted for all MCS indices with a Non-HT PPDU (see Table 2). For each MCS, 100 flooding processes with an initial `rtx_global_counter` value of 10 000 are successively triggered and recorded, making a total of 1 000 000 CCTs. The execution in 100 successive experiment runs overcomes a buffer size limitation of WARPLab and the 16-bit limitation of the `rtx_global_counter` protocol field in group mode.

The decoding results of the received CCTs are stringed together to obtain a sequence of 1 000 000 values for each performance metric. The FRR is calculated over this sequence of decoding results with a variable number of virtual RTx repetitions, which essentially defines the size of a sliding window. If any of the CCTs within the sliding window are received correctly, the corresponding position of the sliding window is considered to result in a successful frame reception. For instance, in case of no virtual RTx repetition, each CCT from a particular group is counted individually for the FRR of this group. In case of one virtual RTx repetition, a reception from a group is considered as successful if the respective CCT or the subsequent CCT generated by this group is received correctly. The FRR is calculated with virtual RTx repetitions for receptions from each of both groups as well as for receptions from any of both groups. With this, the FRR for a variable number of RTx repetitions is emulated by means of a long sequence of CCTs.

Table 16 in Appendix B lists the FRRs of both the LOS and the NLOS receiver for all MCS settings, without virtual RTx repetitions. For receptions from any of both groups, the FRR of the LOS receiver exceeds 99.999 % up to MCS 4, while the FRR of the NLOS receiver exceeds 99.999 % up to MCS 2. Thus, the URLLC reliability requirement can already be met without additional RTx repetitions when a receiver can overhear the CCTs from at least two different groups of nodes that successively forward a flooding frame. Note that in multi-hop network flooding scenarios, each group of nodes that forwards a flooding frame is in communication range to its respective previous-hop group and its respective next-hop group, so that the flooding process propagates through the network.

Device ID	MCS 0			MCS 1			MCS 2			MCS 3		
	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	1	1	0	1	1	0	1	1	0	1	1	0
Matlab NLOS	1	1	0	2	1	0	2	1	0	3	2	1

Device ID	MCS 4			MCS 5			MCS 6			MCS 7		
	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	1	1	0	2	2	1	3	3	2	2	4	1
Matlab NLOS	4	2	1	10	5	3	27	33	10	72	23	14

Table 5: Number of RTx repetitions required to meet the URLLC reliability requirement of $1 - 10^{-5}$

PHY format	MCS 0	MCS 1	MCS 2	MCS 3	MCS 4	MCS 5	MCS 6	MCS 7
Non-HT	120.2	92.2	80.2	64.2	60.2	52.2	48.2	48.2
HT-MF	132.2	92.2	80.2	72.2	68.2	64.2	64.2	60.2

Table 6: Latency per hop with a 32-byte MSDU in microseconds, including a SIFS of 16.2 µs

Hence, a monitoring device is typically in communication range of at least two different groups of nodes, so that it has at least two chances to receive a CCT of a particular flooding frame. Similarly, if a WARP prototype node misses to receive a flooding frame from its previous-hop neighbors, it will have a second chance to receive it from its direct neighbors and a third chance to receive it from its next-hop neighbors if needed. Therefore, the *any* selector in Table 16 reflects the reception performance in multi-hop network flooding scenarios most realistically. Still, in the edge region of a network, a monitoring device may only receive CCTs from the outermost group of nodes.

Table 5 lists the number of virtual RTx repetitions required in addition to a regular flooding frame reception in order to meet the URLLC reliability requirement of $1 - 10^{-5}$. For the LOS receiver and up to MCS 6, the listed numbers for receptions from one particular group are greater by one than the respective numbers for receptions from any of both groups. The FRR of the NLOS receiver is generally better for group ID 1 than for group ID 0, as visible in Table 16. Thus, the number of additional RTx repetitions required with the NLOS receiver varies, especially for higher MCS indices.

Given that the *any* selector reflects the conditions of practical network flooding over multiple hops most realistically, Table 5 indicates that the URLLC reliability requirement might typically be met without additional RTx repetitions for up to MCS 2 and with one additional RTx repetition for up to MCS 4. Note that these results are obtained with the Viterbi algorithm with soft-decision decoding of the Matlab WLAN toolbox [150], which outperforms receivers with suboptimal decoding algorithms [158, Section 12.3].

Table 6 lists the resulting latency per hop with a 32-byte MSDU in microseconds for all MCS indices of the Non-HT and the HT-MF PHY formats, respectively. The URLLC requirement can be met with a latency per hop of 80.2 µs with Non-HT MCS 2 at NLOS conditions or with a latency per hop of 60.2 µs with Non-HT MCS 4 at LOS conditions.

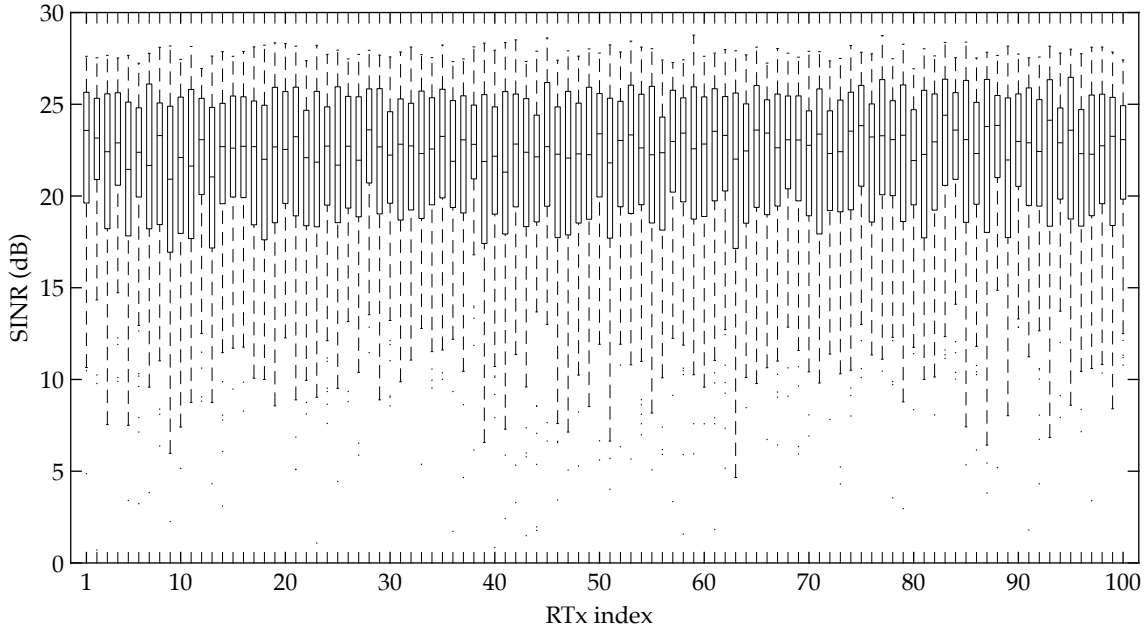


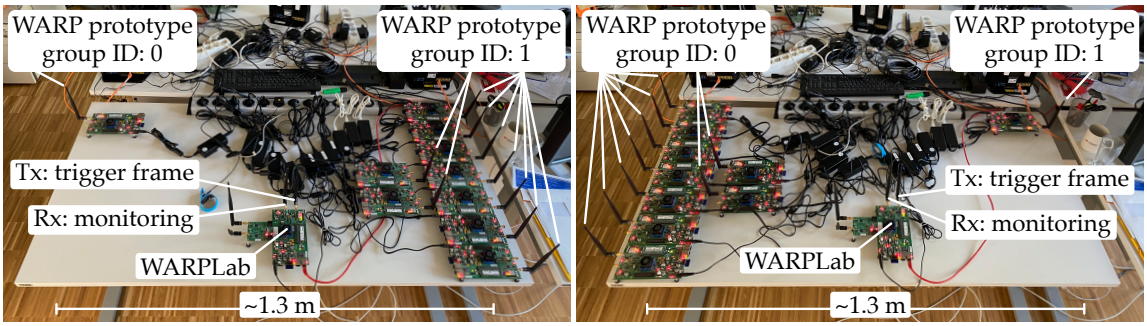
Figure 44: SINR of the first 100 CCTs in 5x5 ping-pong transmissions with a LOS receiver

4.5.3 *SINR over the Retransmission Index*

The signal quality of a CCT depends on the phase offsets (POs), the time offsets (TOs) and the carrier frequency offsets (CFOs) of the transmitters with respect to each other, as described in the analysis in Section 3.4. Since each node synchronizes in time and frequency with respect to a received frame, which in turn typically arrives as a CCT, and since each CCT triggers the generation of another CCT until the respective flooding process terminates, synchronization errors may potentially propagate and grow over time. While the results of the previous experiments show that the signal quality of successive CCTs is quite steady for a long period, it is still unclear how the signal quality evolves right at the beginning of a flooding process and how much it fluctuates.

Figure 44 shows the measured signal-to-interference-plus-noise ratio (SINR) of the first 100 CCTs of 100 runs of ping-pong transmissions between two groups of five nodes each. These results are obtained from the dataset of the experiment presented in Section 4.5.2, while the measured SINR values of successive experiment runs are combined over the respective RTx indices. The SINR characteristics are obtained with MCS 0, but they are similar also for other MCS indices since the SINR is independent of the employed MCS. The plot is generated with the `boxplot` function of Matlab [149].

Figure 44 reveals that the SINR typically ranges around 20 to 25 dB during the first 100 CCTs of a flooding process, indicating that the SINR characteristics of the CCTs are quite consistent from the start of a flooding process. Further, the fact that the SINR characteristics are similar for odd and even RTx indices indicates that the signal quality is equally good for CCTs received from both groups of nodes. Besides receptions with an SINR in the range of 20 to 25 dB, there are also some data points with a slightly higher SINR and some data points with a substantially lower SINR. These variations of the SINR can be explained by the randomness of the POs between the transmitters, so that the interference of a CCT may by chance be constructive, destructive or anything in between.



(a) Group ID 0: 1 node Group ID 1: 9 nodes (b) Group ID 0: 9 nodes Group ID 1: 1 node

Figure 45: Testbed setup with a variable number of nodes per group: (a) first and (b) last setting

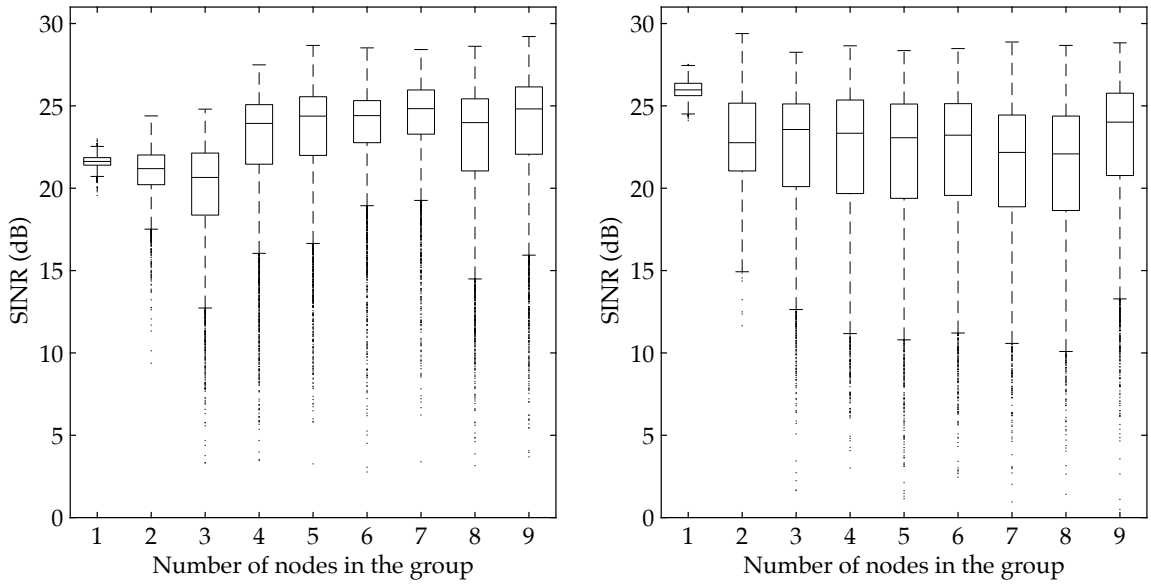
4.5.4 *SINR over the Number of Transmitters*

Figure 45 shows the setup of an experiment in which the CCT signal quality is assessed as a function of the number of concurrent transmitters. The experiment is conducted with nine runs of ping-pong transmissions between two groups of nodes, while the numbers of nodes in the groups are varied between the successive experiment runs.

The experiment starts with the testbed setup shown in Figure 45a, where the left group with the group ID 0 consists of a single node and the right group with the group ID 1 consists of nine nodes. The WARP SDR with WARPLab [250] triggers a flooding process with an initial `rtx_global_counter` value of 10 000 and records the received CCT signals for analysis in Matlab. Afterwards, one node from the right group with the group ID 1 is placed on the left side of the table and is assigned to the group with the group ID 0. Then, another flooding process with 10 000 successive CCTs is triggered and recorded. The procedure is continued until the left group with the group ID 0 comprises nine nodes and the right group with the group ID 1 comprises a single node, as shown in Figure 45b. The flooding frame is a Non-HT PPDU that encapsulates a 32-byte MSDU at MCS 2.

After the experiment, the recorded IQ samples are processed with a decoding script that is based on the Matlab WLAN toolbox [150]. In this process, each decoded CCT is assigned to its group ID by means of the `rtx_global_counter` value in the MAC header. With this, the resulting dataset comprises up to 5 000 CCTs from each of both groups for each number of nodes from 1 to 9, respectively. Note that the number of transmitters that contribute to a CCT is equal to or less than the number of nodes in a group since some nodes may possibly not transmit. More precisely, if a node misses to receive the respective previous RTx from the other group and if it has already generated a retryRTx in the Tx slot before, it does not transmit. Still, due to the good reception performance of the WARP prototype nodes (see Section 4.5.1) and automatic retries (see Section 4.2.1.4), the fraction of nodes of a group that contribute to a CCT is typically high.

Figure 46 shows the measured SINR for each number of nodes for each of both groups, visualized by means of the `boxplot` function of Matlab [149]. With a single transmitter, the overall range of the measured SINR values is quite narrow for each of both groups. This steadiness of the signal quality can be attributed to the static node positions in combination with the fixed signal power of the respective single transmitter. The median SINR values are 21.6 dB for group ID 0 and 26.0 dB for group ID 1, the difference of which can be explained by the different channel conditions of the respective node positions.



(a) Group ID 0 (b) Group ID 1

Figure 46: Measured SINR as a function of the number of concurrent transmitters of a CCT

With two nodes in a group, the overall range of the measured SINR values becomes much wider than with a single transmitter, which applies for both group IDs 0 and 1. While the majority of the received CCTs have an SINR equal to or less than the respective level with a single transmitter, there are also some CCTs that have a slightly higher SINR. Further, with three and more nodes in a group, the range of the measured SINR values gets even wider than with two nodes. For group ID 0, Figure 46a shows that the SINR values with four and more nodes are often a little higher than with one to three nodes. For group ID 1, Figure 46b shows that the SINR values with two and more nodes are often a little lower than with a single node. In addition, with an 800-byte MSDU and with a 1500-byte MSDU, the experiment exhibits similar SINR characteristics.

The wider SINR range of the CCTs in comparison to transmissions from a single node can be explained by the randomness of the POs between the concurrent transmitters. Note that the PO between two concurrent transmitters can be considered to be random due to their uncoupled frequency oscillators. Thus, while the frequency synchronization makes the interference of a CCT coherent, the POs between the concurrent transmitters can make the interference constructive or destructive by chance.

Generally, the signal quality of a CCT can be both a little better or worse than that of a transmission from a single node. The characteristics of the CCT signal quality are quite consistent with an increasing number of transmitters, which makes CCT suitable for mechanisms in which nodes opportunistically join operations, such as network flooding. While the SINR of a CCT may be poor by chance due to the randomness of the POs of the concurrent transmitters with respect to each other, a node typically has at least two opportunities to receive a message in CCT-based network flooding, so that this effect can be overcome reliably, as demonstrated in the experiment presented in Section 4.5.2.

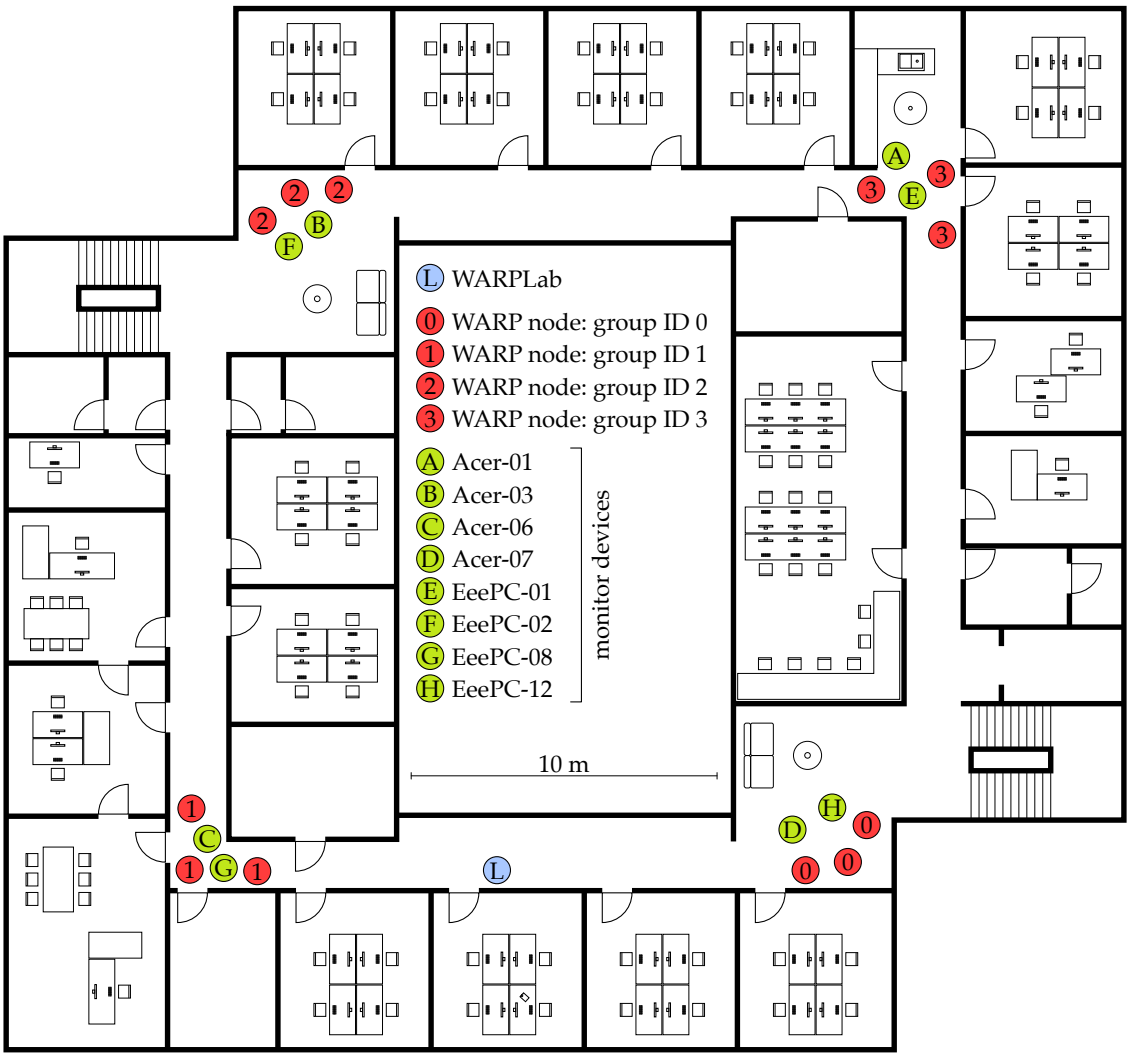


Figure 47: Testbed setup for circular forwarding by four groups of WARP prototype nodes

4.5.5 *Circular Forwarding by Four Groups of Nodes*

Figure 47 illustrates the setup of an experiment in which four groups of three nodes each forward a flooding frame in a circular fashion. This experiment is designed to assess the performance of the WARP prototype under conditions that resemble a CCT-based network flooding scenario more realistically than the ping-pong experiments on a table. The distances between the nodes within each group are $\sim 1 - 2\,\text{m}$, which adds diversity to the channels of the individual CCT signal components. Further, the distances between the groups of nodes are $\sim 15 - 20\,\text{m}$, allowing for rich multi-path effects. Since each flooding frame is forwarded in a circular fashion, successive CCTs propagate through different sections of the office environment, which is another source of diversity.

The WARP SDR with WARPLab [250] is again used to trigger flooding processes and to record the CCTs on the wireless channel for analysis in Matlab. In this experiment, the flooding frame is an HT-MF PPDU comprising 69 OFDM symbols, including the Non-HT PHY preamble with an equivalent length of 4 OFDM symbols (see Section 2.4.2).

MCS	0	1	2	3
MSDU bytes	164	359	554	749

Table 7: Number of MSDU bytes for an HT-MF PPDU with 69 OFDM symbols

The number of OFDM symbols is translated to the maximum possible number of MSDU bytes for the selected MCS, as listed in Table 7. With this, the PPDU has the same length in each experiment run with a different MCS so that the conditions for the frequency synchronization are identical. Still, the MCS might have an effect on the reception performance of the WARP prototype nodes and of the commercial monitor devices. With 69 OFDM symbols, the PPDU duration is 276 µs, so that a flooding process with an initial `rtx_global_counter` value of 10 000 lasts ∼ 2.9 s.

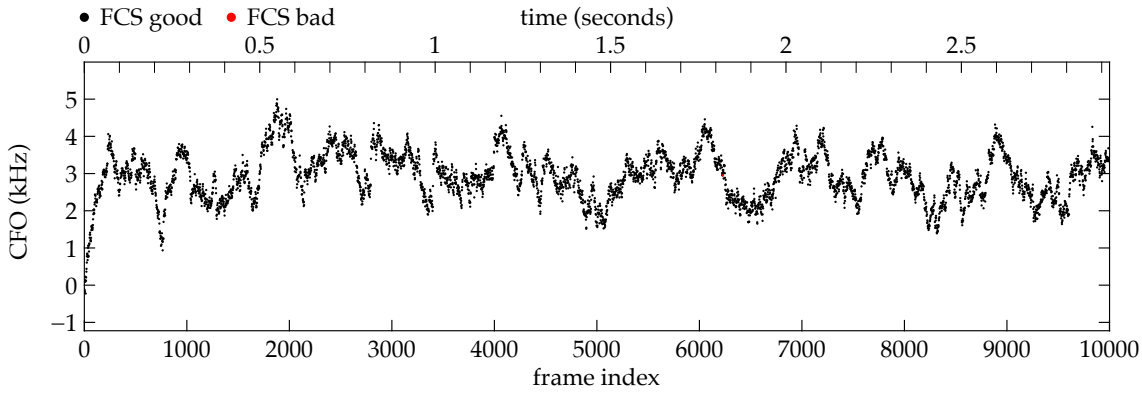
Table 17 in Appendix C lists the FRRs of the individual WARP prototype nodes. Each listed experiment run comprises 10 000 successive CCTs, confirming that network flooding with the WARP prototype nodes works persistently in the office environment. However, with an MCS index higher than 3, a flooding process typically stops before the `rtx_global_counter` value reaches 0, which can be attributed to a reduced SINR due to the relatively large distances between the groups of nodes, leading to frame loss.

Table 18 in Appendix C lists the FRRs of the Matlab decoder and of the commercial devices in monitor mode. Since the WARP SDR with WARPLab is positioned between the groups with the group IDs 0 and 1, it receives the CCTs from these two groups, but not from the groups with the group IDs 2 and 3. Further, each of the commercial monitor devices is placed near one of the four groups, so that it receives CCTs from the respective previous-hop group, from the group nearby and from the next-hop group. The good reception performance of the monitor devices indicates that the time and frequency synchronization of the WARP prototype nodes keeps up accurately over time. In addition, the FRR is also given for receptions from any of the four groups, which mimics the reception performance in a real flooding scenario.

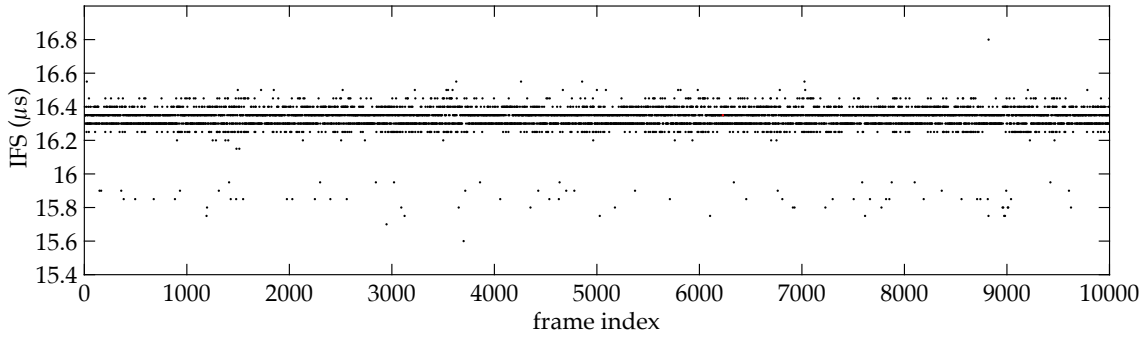
Figure 48 shows the progressions of the CFO and of the IFS, exemplarily for MCS 0. Figure 48a reveals that the CFO tends to drift towards positive frequencies in this experiment and that it is contained at around ∼ 3 kHz. There are a few gaps on the curve, which can be explained by the fact that only CCTs from the groups with the group IDs 1 and 0 are received by the WARP SDR with WARPLab.

Figure 48b shows that most of the IFS measurements range from 16.25 µs to 16.45 µs. The slight increase of the IFS in comparison to Figure 43b can be explained by the larger propagation delay due to a larger distance between the groups of nodes than in the 5x5 ping-pong experiment on a table. In addition, there are some outliers at ∼ 15.9 µs, which are likely caused by retryRTx transmissions. The gap between the IFS of retryRTx frames and RTx frames is rather large in this experiment since the propagation delay is relatively large and since a retryRTx is generated in the fourth Tx slot after the first RTx. The timing of retryRTx frames can potentially be enhanced further through a dynamic calibration by means of propagation delay measurements of the per-hop distance.

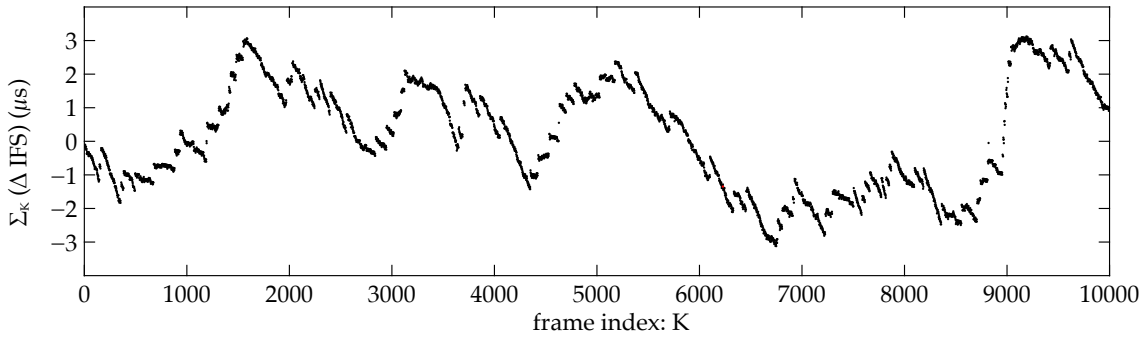
Figure 48c shows that the error of the global time synchronization stays within ∼ ±3 µs. Note that the gaps of the curve are aligned with the occurrences of the IFS outliers with a low value, which are likely caused by retryRTx transmissions.



(a) Measured CFO of CCTs, obtained with a CFO containment factor of $v = 0.01$



(b) Measured IFS of CCTs



(c) Accumulation of Δ IFS: global time synchronization error with respect to the flooding initiator

Figure 48: Progression of performance metrics during circular forwarding at MCS 0 over 2.9 s

4.6 Conclusion

The presented WARP prototype enables CCT-based network flooding in real-time with the OFDM-based IEEE 802.11 Non-HT and HT PHYs. To this end, it precisely synchronizes in time and frequency with respect to a received flooding frame to generate an RTx. With this, multiple nodes can generate a CCT with CI by executing the same procedure, allowing distributed network nodes to join a flooding process opportunistically.

The WARP prototype is based on the IEEE 802.11 reference design for WARP v3 [247] and comprises various extensions and modifications. The frequency synchronization mechanism consists of an enhanced CFO estimator integrated into the PHY Rx core and a frequency shifter integrated into the PHY Tx core [9]. The enhanced CFO estimator combines the L-LTF-based CFO estimate from the preamble with the CPE of the pilot

subcarriers to improve the precision of the CFO estimate, as described in Section 3.6. Figure 41 shows that the $[10\,\%, 90\,\%]$ quantiles of the phase error at the end of a frame are on average accurate to $\sim [-0.021\,\pi, 0.012\,\pi]$ for PPDUs with up to ~ 100 OFDM symbols. The $[10\,\%, 90\,\%]$ quantiles of the relative CFO are accurate to $\sim [-11.8, 14.6]$ Hz with 220 OFDM symbols and change only slightly for longer frames. While the frequency synchronization systems listed in Table 1 are reported to be a little more accurate, the WARP prototype works by means of just a single frame reception.

The time synchronization mechanism employs the preamble-based symbol timing estimation, as described in Section 2.4.3, and hardware timers of the MAC support core. Figure 37 shows that the IFS between successive ping-pong transmissions is consistent.

The WARP prototype employs a low-level MAC protocol whose fields are mapped into the IEEE 802.11 MAC header in three possible formats, allowing commercial devices to receive flooding frames from a network and to inject frames for flooding into a network. The initiator of a flooding process can tune the latency, the reliability and the area coverage by means of the low-level MAC protocol in combination with the MSDU size and the MCS selection. The WARP prototype processes the low-level MAC protocol in real-time so that the behavior of a network is defined by the flooding frame itself. The highest possible PHY data rate with a 20 MHz wide channel is 65 Mbit/s with the HT PHY at MCS 7. The lowest possible latency per hop is 44.2 µs and can be achieved with the Non-HT PHY at MCS 5 or higher. With this, certain performance tradeoffs can be tuned towards specific demands, facilitating the design of dURLLC applications.

The group mode format of the low-level MAC protocol allows a few groups of nodes to forward a flooding frame repeatedly in a cyclic manner, enabling testbed experiments in which a limited number of nodes can generate a large number of successive CCTs, with each CCT triggering the generation of the respective next CCT. In the simplest case, two groups of nodes forward a flooding frame to each other in a ping-pong fashion. Various testbed experiments are conducted in group mode to assess the stability and the temporal performance of CCT-based network flooding with IEEE 802.11.

With five nodes per group, flooding processes with 10 000 successive CCTs can reliably be triggered for various MCS indices and MSDU sizes. Further, commercial devices often exhibit better FRRs for CCTs from five nodes than for transmissions from a single node. These results indicate that the joint synchronization in time and frequency of the WARP prototype nodes keeps up over a large number of successive CCTs, while each CCT serves as the reference for synchronization in time and frequency of the respective next CCT.

The feasibility of the 3GPP URLLC requirement, which is defined as the delivery of a 32-byte MSDU with a reliability of $1 - 10^{-5}$ and with a latency of 1 ms [3], is assessed by means of 1 000 000 CCTs recorded in a 5x5 ping-pong experiment. Since the reliability of a message delivery can be increased through repetitive transmissions, the number of virtual RTx repetitions to obtain an FRR of at least 99.999 % is determined for each MCS. Since different groups of nodes forward a flooding frame successively in multi-hop scenarios, a receiver typically has at least two chances to receive it. Thus, the reception from *any* of both groups in the 5x5 ping-pong experiment is used to emulate two reception opportunities of a node in a WMN. Table 5 shows that the URLLC reliability requirement can be met without additional RTx repetitions with up to Non-HT MCS 2 under NLOS conditions and with up to Non-HT MCS 4 under LOS conditions. Table 6 shows that the latency per hop is 80.2 µs with Non-HT MCS 2 and 60.2 µs with Non-HT MCS 4.

An important prerequisite for meeting the URLLC reliability requirement is that the flooding process itself propagates persistently through the network. The higher the number of nodes in a certain area, the higher is the chance that at least one of these nodes successfully receives a CCT from the previous-hop neighborhood and forwards it. Figure 46 shows that the signal quality of a CCT in terms of the SINR is quite steady for an increasing number of concurrent transmitters. Thus, a CCT-based flooding process propagates most persistently through a network when the node density is high.

Figure 44 shows that the SINR characteristics of successive CCTs are quite consistent right from the start of a flooding process. This finding reveals that the signal quality does not deteriorate due to synchronization errors of individual nodes that potentially propagate further through the network. Instead, the nodes achieve a stable signal quality while they repetitively resynchronize in time and frequency to the CCTs from each other. This result is also supported by Figure 43, which shows that the characteristics of the CFO and of the IFS are quite consistent for 10 000 successive CCTs. Note that the consistency of the SINR characteristics from the start of a flooding process also confirms the validity of the experimental design in Section 4.5.2, in which the decoding results of 100 experiment runs are stringed together to obtain a sequence of 1 000 000 CCTs.

Figure 44 and Figure 46 show that CCTs have a wider SINR range than transmissions from a single node, which is due to the randomness of the POs between the concurrent transmitters. While all CCTs are generated with CI through frequency synchronization, the POs can make the interference of a CCT constructive or destructive by chance. Still, a missed reception of a CCT with a low SINR can be overcome easily as most nodes have at least two chances to receive a flooding frame from different groups of nodes. Further, while the POs may cause a low SINR at a receiver, other receivers may get a higher SINR for the same CCT due to spatial diversity. Thus, a flooding process can still propagate persistently through a network. In addition, RTx repetitions can improve the reliability of flooding and message delivery further. Phase dithering can be employed in future work to randomize the POs in successive CCTs, as described in Section 2.1.7.3.

CCT-based network flooding also allows for global time synchronization of all network nodes with respect to the initiator of a flooding process. In testbed experiments with 5x5 ping-pong transmissions, the global time synchronization error does not exceed $\pm 5.3\,\mu s$ over 10 000 successive CCTs when the expected IFS $\widetilde{T}^{\mathrm{IFS}}$ is accurate. In practical multi-hop settings, the accuracy of the global time synchronization depends on the accuracy of the measured average IFS $\widetilde{T}^{\mathrm{IFS}}$ and on the variability of the IFS T_k^{IFS} over different hops. A node could measure the average IFS $\overline{T}^{\mathrm{IFS}}$ by flooding a probing frame over a certain number of hops that is subsequently flooded in reverse direction back to the source. When the node includes the measured average IFS $\widetilde{T}^{\mathrm{IFS}}$ in the MSDU of a flooding frame, possibly together with an absolute time reference, all nodes can synchronize in time. Alternatively, the propagation delay could also be estimated per hop by an integration of Cooperative Analog and Digital (CANDI) [46], as described in Section 2.3.3.

The experiment presented in Section 4.5.5 demonstrates that CCT-based network flooding is practically feasible with a hop distance of $\sim 17\,m$ in an office environment. Moreover, the low-level MAC protocol of the WARP prototype enables commercial devices to stream a video over multiple hops and to receive it, as discussed in Section 7.3.4. The FRRs of five receivers in an office environment that is populated with several WARP prototype nodes around a video source node range from 99,85 % to 99,97 % [166].

5

COMMODITY HARDWARE PROTOTYPE – ASUS RT-AC86U

This chapter presents another prototype for concurrent cooperative transmission (CCT) with coherent interference (CI) by means of synchronization in time and frequency. The prototype is based on the Asus RT-AC86U wireless router and is created through firmware modifications, showcasing the feasibility of CCT with commodity hardware. This approach essentially leverages capabilities of the integrated IEEE 802.11 chip that are not exposed by the original firmware and that are harnessed through reverse engineering. In particular, the Asus RT-AC86U prototype leverages registers of its IEEE 802.11 chip to extract the carrier frequency offset (CFO) estimated during a frame reception and to tune its carrier frequency by reconfiguring its voltage-controlled oscillator (VCO). Further, the Asus RT-AC86U prototype repurposes the acknowledgment (ACK) engine of its IEEE 802.11 chip to generate a retransmission (RTx) of a received flooding frame with accurate timing. With this, multiple Asus RT-AC86U prototype nodes can synchronize in time and frequency with respect to each other in order to generate CCTs with CI, even with up to four multiple input multiple output (MIMO) spatial streams.

In contrast to the prototype presented in Chapter 4, which is based on the Wireless Open Access Research Platform (WARP) v3, the Asus RT-AC86U prototype extracts four coarse CFO estimates from each received frame with a quantization step size of 2.3 kHz. To overcome this limitation, it computes the average CFO of multiple received frames, which enhances the accuracy of the estimated CFO with respect to a particular sender. Therefore, a frequency synchronization procedure is executed before generating CCTs, allowing multiple Asus RT-AC86U prototype nodes to pre-synchronize with respect to a common frequency reference by means of a couple of tuning frame receptions. Afterwards, the syntonized nodes can generate CCTs with CI in testbed experiments. The Asus RT-AC86U prototype particularly allows for ping-pong transmissions between two groups of nodes by assigning each node to a specified group and by repurposing the Duration/ID field of the IEEE 802.11 medium access control (MAC) header as a counter. While this mode of operation resembles the group mode of the low-level MAC protocol of the WARP prototype, as described in Section 4.2.2.3, the Asus RT-AC86U prototype does not feature the entire low-level MAC protocol presented in Section 4.2.1.

This chapter is organized as follows. Section 5.1 provides background information on the design methodology through reverse engineering and firmware modifications. Section 5.2 describes the Asus RT-AC86U prototype implementation, covering the CFO estimation during frame receptions and the VCO tuning for frequency synchronization as well as the utilization of the ACK engine for time synchronization. Section 5.3 presents functional validations of the Asus RT-AC86U prototype regarding its time and frequency synchronization mechanisms as well as the stability of its carrier frequency over time. Section 5.4 presents the evaluation of the Asus RT-AC86U prototype in various testbed experiments, comprising runs of 10 000 successive CCTs with up to nine concurrent transmitters and with up to four MIMO spatial streams at 40 MHz channel bandwidth. Section 5.5 provides concluding remarks.

5.1 Background on Firmware Modifications

While firmware modifications have the potential to enable features and mechanisms that are not envisaged by product manufacturers, implementing new features by such means is an ambitious endeavor. Chip and device manufacturers share detailed documents of their products only with customers, but they do not disclose them publicly nor do they offer open-source firmwares, so that the functions of hardware components and their registers must be reverse-engineered to identify suitable primitives for new features. However, there is no guarantee that sought-after functions exist or that certain values are exposed through registers as desired. To increase the chance of success, a thorough analysis of the extracted firmware as well as the design of custom tools to trigger and to observe the behavior of a device under test (DUT) under certain conditions is required.

IEEE 802.11 chips can be subdivided into the categories SoftMAC and FullMAC [204]. While both categories of chips process time-critical tasks of the physical layer (PHY) and of the MAC layer themselves, they differ in how they manage tasks that are less time-critical. On the one hand, the MAC sublayer management entity (MLME) [104, Section 4.9] of a SoftMAC chip is implemented as a driver of the host operating system, i.e., the host processes received frames and the host can transmit arbitrary frames. On the other hand, a FullMAC chip comprises the MLME within itself and provides an Ethernet bridge to exchange frames with the host in both directions. Most modern mobile devices like smartphones are equipped with a FullMAC chip due to certain advantages over a SoftMAC chip. A FullMAC chip is easier to integrate in a product than a SoftMAC chip since it readily includes an MLME. In addition, a chip vendor does not have to release drivers or to disclose code. Further, a FullMAC chip can save energy in comparison to a SoftMAC chip as it does not employ the central processing unit (CPU) of the host for management tasks, i.e., the CPU may stay in a sleep state while the FullMAC chip processes received IEEE 802.11 frames, until application data for the host arrive.

SoftMAC chips have the advantage that they allow new link-layer functions to be implemented in the device driver employed by the host operating system. For instance, various Linux device drivers for wireless network interfaces implement the IEEE 802.11s standard and expose their functionality through the `mac80211` driver interface, allowing wireless mesh networks to be set up without infrastructure, like Freifunk [76, 152, 182]. However, low-latency mechanisms require time-critical functions to be executed quickly by the IEEE 802.11 chip in order to respond to a received frame in a timely manner. In particular, in CCT-based network flooding, an RTx of a flooding frame should be generated a short interframe space (SIFS) duration of $\sim 16\,\mu s$ after the reception, with consistent timing. Thus, firmware modifications of the IEEE 802.11 chip are necessary.

The Asus RT-AC86U wireless router is equipped with the FullMAC chips BCM4365E and BCM4366E from Broadcom for communications through IEEE 802.11a/b/g/n and IEEE 802.11ac, respectively. The primary goal of prototyping with this device is to facilitate CCT with CI with IEEE 802.11g/n, allowing for comparisons with the WARP prototype presented in Chapter 4. However, since the Asus RT-AC86U wireless router readily supports IEEE 802.11n for both 20 MHz and 40 MHz wide channels in the 2.4 GHz and in the 5 GHz bands for all modulation and coding schemes (MCSs), another goal is to explore the feasibility of CCT with CI also with up to four MIMO spatial streams. To meet these goals, firmware modifications of the BCM4365E chip are required.

FullMAC chips from Broadcom have a common architecture with components such as an embedded ARM processor for less time-critical tasks, a D11 core for time-critical MAC processing as well as digital baseband, analog frontend and radio components [204]. Since CCT-based network flooding requires event processing in real-time, the D11 core of the BCM4365E chip is particularly relevant for developing the Asus RT-AC86U prototype. The D11 core comprises a programmable state machine (PSM) that quickly responds to changes of condition registers set by the PHY engine and that sets special purpose registers of the PHY engine to handle both frame receptions and transmissions [204]. The D11 core has separate memories for code and data, the latter of which is referred to as the shared memory since it can also be accessed by external components [204]. The PSM executes the so-called *ucode*, which resides in the code memory during runtime. The ucode is contained in the firmware of the embedded ARM processor and is loaded into the code memory during the initialization of the D11 core [204]. Besides the memories for code and data, the template RAM is able to buffer in-phase quadrature (IQ) samples of received frames and of arbitrary signals for transmissions [204].

Nexmon [204] is a firmware patching framework for Broadcom chips that allows writing firmware patches for the embedded ARM processor in C and for the D11 core in assembler, while the ucode can be edited by means of a disassembler/assembler. Nexmon provides various tools for automating the process of firmware analysis and patching and showcases the potential of modified firmwares with applications such as a monitor mode for smartphones, reactive jamming and a PHY-based covert channel. Nexmon is not used to automate the firmware patching of the Asus RT-AC86U prototype, but it reveals many useful insights into this complicated subject [204].

5.2 Implementation

To facilitate the development process and the operation of the Asus RT-AC86U prototype, various functions of the BCM4365E chip are exposed to the user space of the device. These functions are wrapped into shell scripts that may either be called interactively by a user logged in via Secure Shell (SSH) over Ethernet or by an external controller that automates tasks across multiple devices. The latter mode of operation particularly allows the automation of reverse-engineering processes and the execution of experiments. To speed up command invocations from an external controller in comparison to SSH and to improve the robustness of control automation over Ethernet, a command server is employed that listens for incoming commands on a network socket and executes them.

The implementation of the Asus RT-AC86U prototype comprises the following features. Section 5.2.1 presents the CFO estimation by means of four PHY registers, each of which exposes a coarse CFO estimate of each received frame. The addresses of these registers are identified through an automated reverse engineering process that employs a WARP software-defined radio (SDR) with WARPLab [246, 250]. Section 5.2.2 presents the tuning of the VCO through a radio register, which allows the carrier frequency to be shifted. Section 5.2.3 presents the generation of an RTx of a received frame through the D11 core. Section 5.2.4 describes the integration of these features, which allows multiple nodes to be pre-synchronized with respect to a common frequency reference so that two groups of nodes can generate successive CCTs in a ping-pong fashion.

5.2.1 *Rx: CFO Estimation*

IEEE 802.11 receiver implementations estimate and compensate the CFO with respect to the transmitter of a frame to mitigate the impact of intercarrier interference (ICI), as described in Section 2.4.4. Therefore, there is a chance that the PHY engine of the BCM4365E chip exposes its CFO estimate of a received frame through some registers. However, a manual inspection of several PHY registers with an educated guess of some promising addresses does not reveal any values representing the CFOs of received frames. Therefore, an automated reverse engineering process is conducted to search the entire 16-bit address space of the PHY engine for registers that expose the CFO.

The automated reverse engineering process is executed in two steps. In the first step, a dataset with several dumps of the 16-bit address space of the PHY engine is generated, while each dump is created when the Asus RT-AC86U device receives a trigger frame that is transmitted with an artificial CFO by a WARP SDR with WARPLab [246, 250]. In this process, the WARP SDR sends a trigger frame repeatedly with different CFOs. In the second step, the dataset is processed, while each address gets assigned a score that indicates to what extent the respective sequence of read PHY register values correlates with the sequence of known CFOs. The addresses with the highest absolute scores are the most promising candidates for PHY registers exposing the CFO of a received frame. Both steps are described in detail below, followed by a presentation of the results.

During the first step of the reverse engineering process, a Matlab script controls both a WARP SDR with WARPLab [246, 250] and an Asus RT-AC86U device over Ethernet. The Asus RT-AC86U device runs a modified firmware on the BCM4365E chip so that the D11 core exposes the contents of specified PHY registers after each reception of a trigger frame, which in turn is detected by means of a specific destination MAC address. When a trigger frame arrives, the D11 core reads the addresses of the PHY registers from specific locations of the shared memory and extracts the contents of each PHY register by calling a routine that selects its address and by reading from a special purpose register. Then, the D11 core transfers the read values into the MAC service data unit (MSDU) of the received frame, i.e., it overwrites the MSDU with the read values to expose them. The D11 core can process up to 108 addresses at a time in this procedure.

To inspect the entire 16-bit address space of the PHY engine through this procedure, the Matlab script processes successive slices of the address space in a loop of 607 cycles. In each cycle, the Matlab script first configures the Asus RT-AC86U device by writing the addresses of the slice into the specific locations of the shared memory of the D11 core. Further, it starts `tcpdump` on the Asus RT-AC86U device to capture the trigger frames with the exposed values from the PHY registers and to store them into a `pcap` file [241]. Then, the Matlab script uses the WARP SDR to transmit a trigger frame several times, while shifting the carrier frequency of the individual transmissions through baseband processing so that the Asus RT-AC86U device is exposed to a sequence of artificial CFOs. At the end of each cycle, the Matlab script stops `tcpdump` on the Asus RT-AC86U device and downloads the generated `pcap` file, besides a few other operational tasks.

In the second step of the automated reverse engineering process, the values in the `pcap` files are extracted with a tool that is based on `libpcap` [241]. Then, a score is computed for each address of the PHY engine that indicates to what extent the corresponding sequence of extracted values correlates with the actual sequence of CFOs.

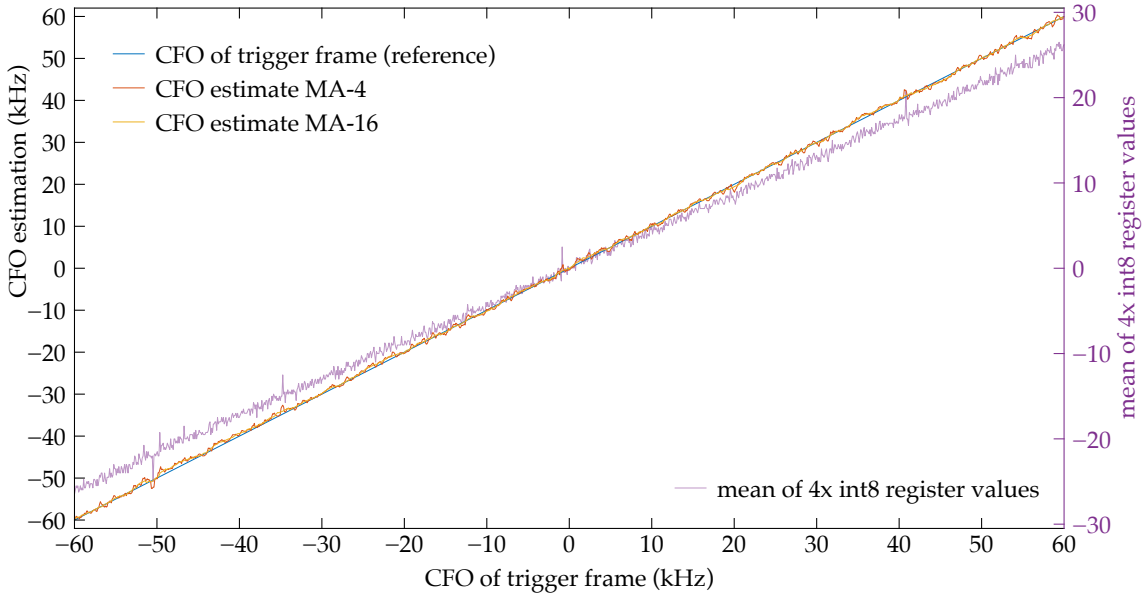


Figure 49: CFO estimation through four 8-bit fields of two PHY registers of the BCM4365E chip

The automated reverse engineering process is first executed with a sequence of artificial CFOs that comprises a sweep from −20 kHz to 20 kHz in steps of 4 kHz, followed by a sweep from 20 kHz to −20 kHz in steps of 4 kHz, while the physical CFO between the WARP SDR and the Asus RT-AC86U device is compensated according to an estimate. Note that a CFO sweep in only one direction would make the score peak also for PHY registers whose values increase or decrease over time due to other reasons than the CFOs of the received trigger frames, such as counters. The score is found to peak particularly for the addresses $\texttt{0x0152} + k \cdot \texttt{0x2000}$ and $\texttt{0x0159} + k \cdot \texttt{0x2000}$, $k \in \{0, \ldots, 7\}$.

The automated reverse engineering process is executed another time, while only the contents of the PHY registers with the aforementioned suspected addresses are dumped. As the number of suspected addresses is smaller than 108, only a single run is required, so that all sequences of values are read during the same receptions of trigger frames. In this run, the artificial CFO is swept from 60 kHz to −60 kHz in steps of 100 Hz. A manual inspection of the extracted values reveals that the sequences of values at the addresses $\texttt{0x0152} + k \cdot \texttt{0x2000}$, $k \in \{0, \ldots, 7\}$ are all identical and that the sequences of values at the addresses $\texttt{0x0159} + k \cdot \texttt{0x2000}$, $k \in \{0, \ldots, 7\}$ are all identical, respectively. This finding indicates that the three uppermost address bits might not be wired, so that addresses with an offset of $k \cdot \texttt{0x2000}$ effectively point to the same memory resources. A manual inspection of the two different sequences of values reveals that the lower eight bits and the upper eight bits of each value are often identical or very close to each other. This finding indicates that each 16-bit value comprises two different 8-bit CFO estimates. Thus, the PHY engine of the BCM4365E chip exposes four different 8-bit CFO estimates upon each frame reception, which can be read from the PHY registers with the 16-bit addresses `0x0152` and `0x0159`. These four CFO estimates have a step size of ~ 2.3 kHz.

Figure 49 shows the mean value of the four PHY registers, which is still quite noisy. A multiplication by a scaling factor of 2.3 kHz yields the corresponding CFO estimate. Applying a moving average over multiple values enhances the CFO estimation accuracy.

5.2.2 *Tx: VCO Tuning*

To explore means to tune the carrier frequency of the Asus RT-AC86U wireless router, various registers are checked manually by writing different values into them while observing the spectrum of transmitted signals. The firmware of the BCM4365E chip particularly comprises a channel table that is used to set up the radio through 62 registers during an initialization procedure. One of these radio registers is found to tune the VCO of the radio. Measurements with Universal Software Radio Peripheral (USRP) SDRs [229] and different Asus RT-AC86U devices show that the VCO register can be used to tune the carrier frequency of transmissions linearly with a quantization step size of $\sim 115\,\text{Hz}$.

5.2.3 *Tx: Generation of RTx*

To generate an RTx of a received flooding frame a SIFS duration of $\sim 16\,\mu\text{s}$ after the frame reception and with consistent timing, the respective processing steps must be carried out by the D11 core. Thus, the ucode is modified such that a received flooding frame can be identified through a specific destination MAC address. If this address matches, the received frame is copied to the shared memory of the D11 core during the reception. The ACK engine of the D11 core is repurposed to retransmit the frame subsequently from the shared memory. In this process, however, the frame check sequence (FCS) is found to be correct only if the MAC protocol data unit (MPDU) has at most 255 bytes. To overcome this limitation, the MPDU is additionally copied from the shared memory to the template RAM of the D11 core and is then transmitted from there. With this, an RTx can be generated with a correct FCS also for MPDUs with more than 255 bytes.

To allow for experiments with ping-pong transmissions between two groups of nodes, the Asus RT-AC86U prototype generates an RTx of a received flooding frame only if the value in the Duration/ID field of the IEEE 802.11 MAC header is either odd or even, depending on the group membership of the device, and if this value does not exceed a preconfigured maximum value. In addition, the Asus RT-AC86U prototype increments the value in the Duration/ID field before generating an RTx of a received flooding frame.

5.2.4 *Integration*

The three functional components of CFO estimation, VCO tuning and RTx generation are integrated into the Asus RT-AC86U prototype to facilitate CCT with CI through multiple devices. The frequency synchronization mechanism is implemented as a separate procedure, so that multiple devices have to be pre-synchronized to a common reference frequency before they can be used to generate CCTs with CI, which is for two reasons. First, the four PHY registers that expose the CFO estimates of a received frame have a coarse quantization step size of $\sim 2.3\,\text{kHz}$. In addition to computing an average of these four values, the accuracy of the CFO estimation can be enhanced further by computing the average of the CFO estimates of multiple frame receptions from the same sender. Second, the VCO register is set during an initialization procedure and cannot be altered by the D11 core during the processing of a received frame. All functions for the device configuration and the execution of the frequency synchronization are wrapped into shell scripts that can be called either interactively or by an automated process via Ethernet.

5.3 Functional Validation

To ensure that the Asus RT-AC86U prototype functions properly and to characterize its time and frequency synchronization mechanisms, various testbed experiments are conducted with a WARP SDR running WARPLab [246, 250] and an Asus RT-AC86U prototype node as a DUT. The two devices are placed in line-of-sight (LOS) at a distance of ~ 1 m from each other. Each experiment comprises two essential phases.

First, the frequency synchronization procedure is conducted, i.e., the DUT synchronizes to the carrier frequency of the WARP SDR while the latter sends multiple tuning frames. Besides sending with its native carrier frequency, the WARP SDR may also introduce an artificial CFO to make the DUT tune to a frequency relative to the native carrier frequency of the WARP SDR. The frequency synchronization procedure is done when the DUT has successfully received a specified number of tuning frames, i.e., the WARP SDR sends additional tuning frames if the DUT misses some receptions.

Second, after the DUT has synchronized to the carrier frequency of the WARP SDR, the WARP SDR sends trigger frames, to each of which the DUT responds with an RTx. The WARP SDR, in turn, receives an RTx from the DUT by means of a second antenna and transfers the recorded IQ samples to a Matlab workspace for processing and analysis. In this process, the WARP SDR may pause for a small amount of time after each reception of an RTx and before sending the next trigger frame, so that the time span of taking measurements can be extended while keeping the amount of produced data manageable. With this, the characteristics of the carrier frequency of the DUT can also be observed over a long period of time after executing the frequency synchronization procedure.

5.3.1 *Frequency Stability*

The stability of the carrier frequency of the Asus RT-AC86U prototype is characterized in a testbed experiment over a time period of ~ 108 minutes. A DUT is first synchronized to the native carrier frequency of a WARP SDR by means of 1024 tuning frame receptions. Then, the WARP SDR periodically sends a data frame that triggers the DUT to respond with an RTx of the frame. In addition, the WARP SDR sends two request to send (RTS) frames immediately before each data frame to prevent interference by other devices. If the WARP SDR does not receive an RTx from the DUT after its data frame transmission, it instantly retries to trigger a response by the DUT until it receives an RTx. The Matlab script pauses for one second after receiving the IQ samples of an RTx and before sending the data frame for the next time. In addition, the transfers of IQ samples over Ethernet and the data processing in Matlab take ~ 86 ms. The experiment comprises 6000 such trigger-response cycles. The experiment is conducted on channel 157 in the 5 GHz band, i.e., on a 20 MHz wide channel with a center frequency of 5785 MHz.

For the evaluation, the enhanced CFO estimation technique presented in Section 3.6 is employed to obtain an accurate CFO estimate for each RTx received from the DUT. The data frame is a Non-HT PHY protocol data unit (PPDU) at MCS 4 (see Table 2) that encapsulates an MSDU of 766 bytes. With this, the PPDU consists of a sequence of 68 orthogonal frequency-division multiplexing (OFDM) symbols plus the PHY preamble with an equivalent length of 4 OFDM symbols (see Section 2.4.2), so that the enhanced CFO estimator can leverage the common phase error (CPE) of 68 OFDM symbols.

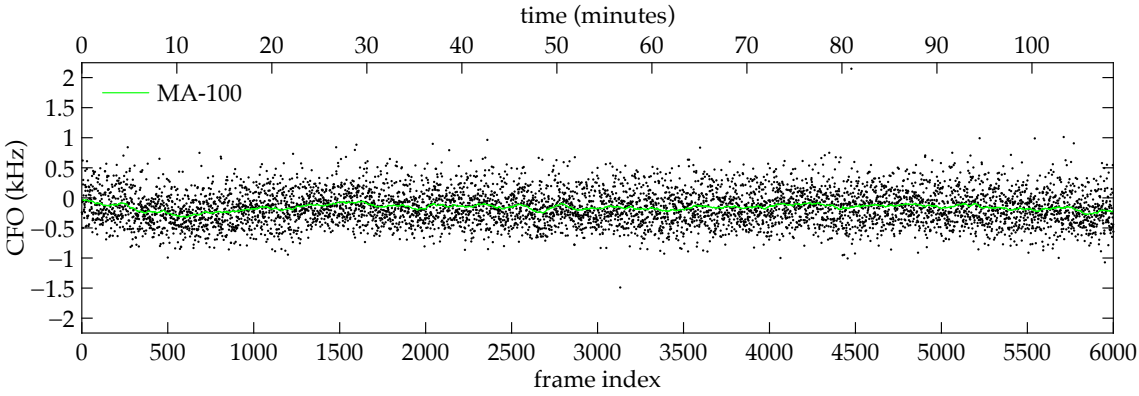


Figure 50: The initial frequency synchronization remains effective over more than one hour.

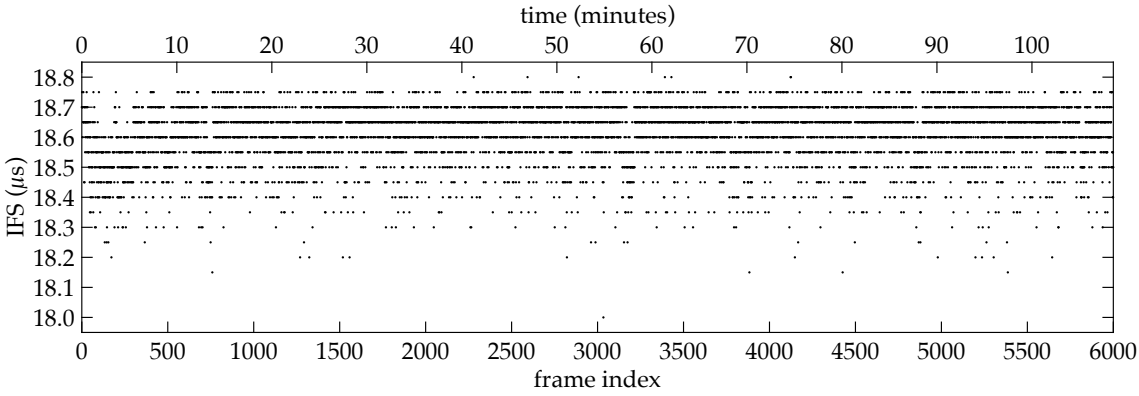


Figure 51: The measured IFS typically ranges from ∼ 18.4 µs to ∼ 18.75 µs, with some outliers.

Figure 50 shows the measured CFOs of the 6000 RTx frames from the DUT with respect to the native carrier frequency of the WARP SDR. The CFOs of the RTx frames are randomly scattered by ∼ ±500 Hz around a mean value, indicating small frequency variations that may occur on both devices due to oscillator jitter (see Section 2.2.2). In addition to the CFO measurements of the individual frames, Figure 50 also shows the moving average of 100 successive CFO values. The moving average starts at −35 Hz, confirming the effectiveness of the preceding frequency synchronization procedure. In the course of the experiment, the moving average ranges from −29 Hz to −322 Hz, indicating that the mean carrier frequency of the DUT is quite stable over a long time. However, this CFO stability is found to occur only at a constant room temperature.

5.3.2 *Timing Consistency*

Figure 51 shows the measured interframe space (IFS) values between the 6000 pairs of trigger frames from the WARP SDR and the respective RTx frames from the DUT, which are obtained from the dataset of the previous experiment presented in Section 5.3.1. Most IFS values range from ∼ 18.4 µs to ∼ 18.75 µs, i.e., their spread is narrower than the guard interval (GI) of 800 ns of the Non-HT PHY. Thus, intersymbol interference (ISI) can be avoided at least for a few successive CCTs, indicating that CCT-based network flooding is feasible. The deviation from the specified SIFS duration of 16 µs is uncritical.

5.3.3 *Frequency Synchronization Accuracy*

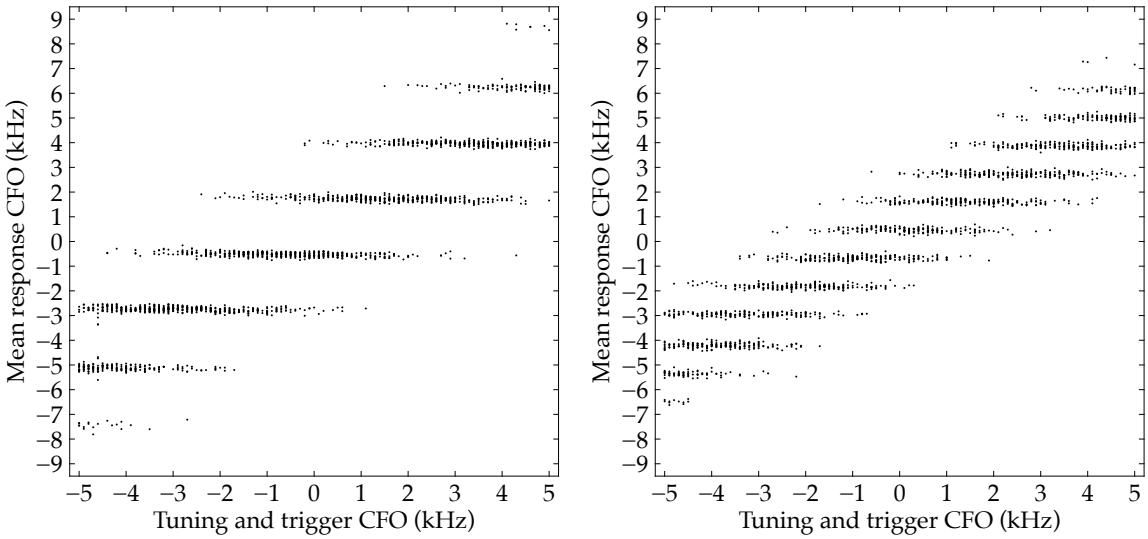
The accuracy of the frequency synchronization of the Asus RT-AC86U prototype depends on both the accuracy of its CFO estimation mechanism on the receive (Rx) side and the accuracy of its mechanism to tune the carrier frequency on the transmit (Tx) side. On the Rx side, each CFO estimate extracted from the four PHY registers during a frame reception has a coarse quantization step size of $\sim 2.3\,\text{kHz}$, as described in Section 5.2.1. This limitation can be overcome by averaging the CFO estimates of multiple frames, i.e., the CFO estimation accuracy can be enhanced by using more tuning frames during the frequency synchronization procedure. On the Tx side, there are two limitations. First, the carrier frequency fluctuates by up to $\sim \pm 500\,\text{Hz}$ around a mean value during transmissions, as illustrated in Figure 50. There are no means to mitigate this effect. Second, the mean carrier frequency can only be tuned in steps of $\sim 115\,\text{Hz}$ by means of the VCO register, as described in Section 5.2.2.

The accuracy of the frequency synchronization of the Asus RT-AC86U prototype is characterized in a testbed experiment in which a DUT is synchronized to a specified carrier frequency by means of a specified number of tuning frames that are transmitted by a WARP SDR with WARPLab [246, 250]. The carrier frequency is defined in relation to the native carrier frequency of the WARP SDR, i.e., the tuning frames are transmitted with an artificial CFO through baseband processing in Matlab. After the frequency synchronization procedure, the WARP SDR sends a specified number of data frames, each of which triggers the DUT to generate an RTx. Since the CFOs of the RTx frames from the DUT may fluctuate by up to $\sim \pm 500\,\text{Hz}$ around a mean value, the estimates of them are averaged to approximate the mean CFO of the DUT. The entire procedure is repeated a specified number of times to obtain different instances of the mean response CFO of the DUT after different executions of the frequency synchronization procedure. This entire procedure, in turn, is executed with a parameter sweep of the artificial CFO from $-5\,\text{kHz}$ to $5\,\text{kHz}$ in steps of $100\,\text{Hz}$ and with different numbers of tuning frames. The experiment is conducted on channel 157 in the 5 GHz band, i.e., at 5785 MHz.

Figure 52 shows the mean response CFO of the DUT with respect to the native carrier frequency of the WARP SDR as a function of the artificial CFO of the tuning frames. Each data point is the mean value of the CFO estimates of 20 RTx frames from the DUT. Further, each subplot comprises 20 such data points for each artificial CFO.

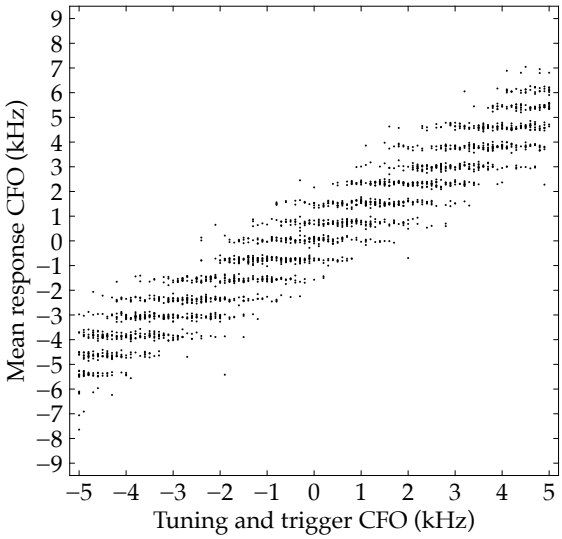
Figure 52a shows that the mean response CFO values of the DUT are aggregated in clusters with a distance of $\sim 2.3\,\text{kHz}$ from each other when a single tuning frame is used. The clusters overlap such that applying a particular artificial CFO during the frequency synchronization procedure may result in a mean response CFO that belongs to one of up to four clusters. These findings reflect the fact that the PHY registers expose noisy CFO estimates of a received frame with a large quantization step size of $\sim 2.3\,\text{kHz}$.

Figure 52b shows that when two tuning frames are used, the cluster distances are halved compared to a single tuning frame. In fact, computing the average of the CFO estimates of two tuning frames allows the DUT to obtain a CFO estimate that lies in the middle of the values that can be represented by the encoding of the PHY registers. Further, the clusters in Figure 52b are narrower than in Figure 52a, so that the range of the mean response CFO values is also narrower than with a single tuning frame.

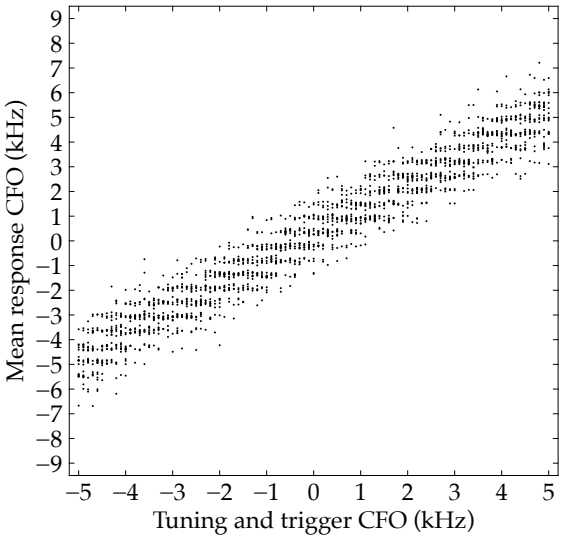


(a) Number of tuning frames: 1

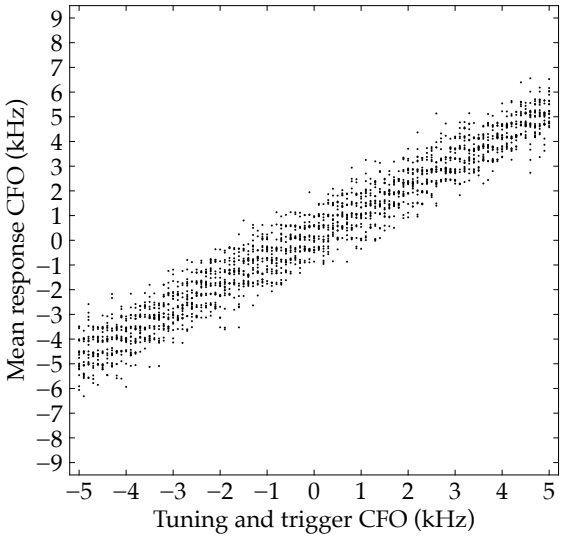
(b) Number of tuning frames: 2



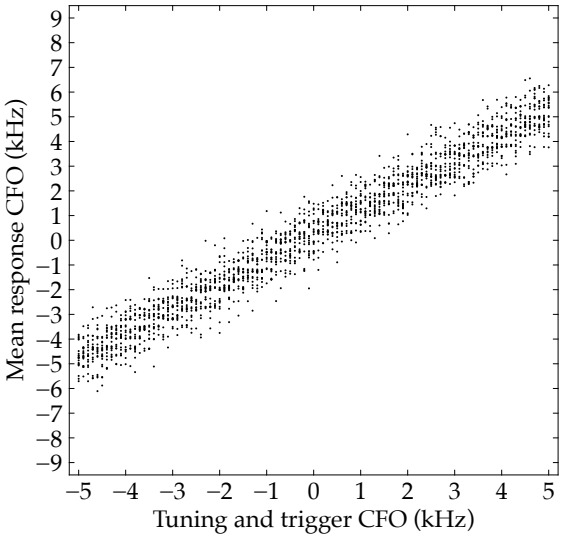
(c) Number of tuning frames: 3



(d) Number of tuning frames: 4



(e) Number of tuning frames: 5



(f) Number of tuning frames: 6

Figure 52: Mean response CFO with respect to the native carrier frequency of the WARP SDR

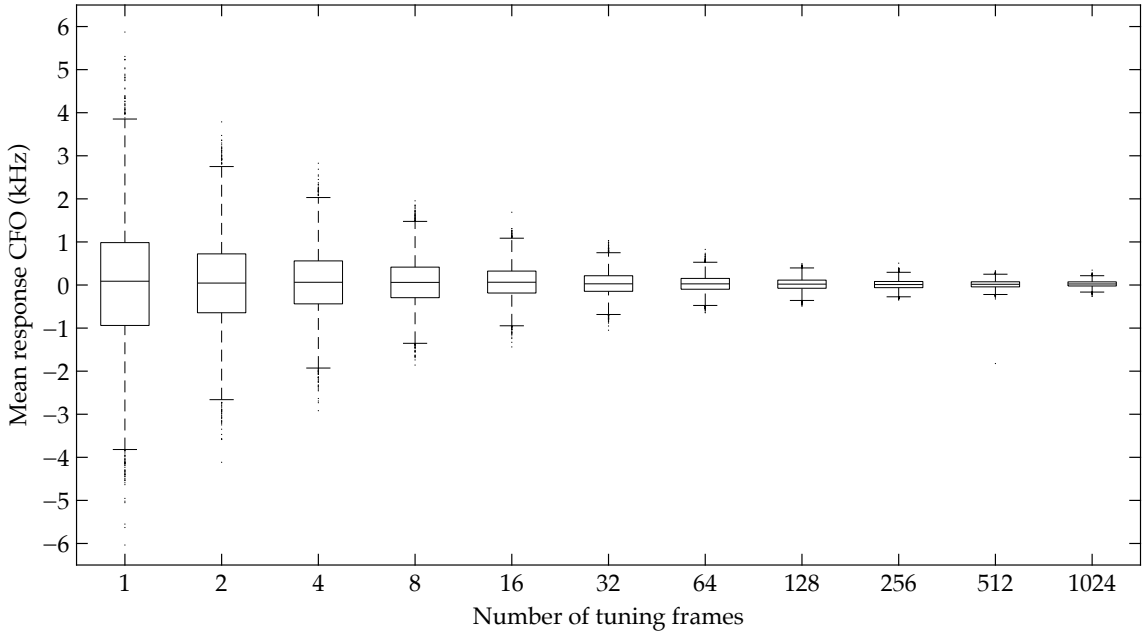


Figure 53: Mean response CFO with respect to the absolute reference frequency

Figure 52 further illustrates the characteristics of the mean response CFO values when three to six tuning frames are used. The distances between the clusters become narrower in Figure 52c and in Figure 52d until the clusters merge in Figure 52e and in Figure 52f. Hence, the ranges of the mean response CFO values become narrower with an increasing number of tuning frames, i.e., the frequency synchronization becomes more accurate.

To assess the accuracy of the frequency synchronization procedure for a larger number of tuning frames, the experiment is conducted once again, while the number of tuning frames is swept exponentially from 1 to 1024 with a ratio of 2 between successive steps. The artificial CFO is again swept linearly from $-5\,\text{kHz}$ to $5\,\text{kHz}$ in steps of $100\,\text{Hz}$. For each of these steps, 100 mean response CFO values are collected, each of which, in turn, is obtained by averaging the CFO estimates of 100 RTx frames from the DUT, after synchronizing the DUT in frequency with the respective number of tuning frames.

Figure 53 shows the deviation of the mean response CFO values from the respective artificial CFO to which the DUT is synchronized. For each number of tuning frames, all steps of the parameter sweep of the artificial CFO are combined, so that the plot effectively shows the characteristics of the mean response CFO of the DUT with respect to the absolute reference frequency, while incorporating a range of reference frequencies. The plot is generated by means of the `boxplot` function of Matlab [149].

With a single tuning frame, Figure 53 shows that the majority of the mean response CFO values are accurate to $\sim \pm 1\,\text{kHz}$ and that most of the values are accurate to $\sim \pm 3.9\,\text{kHz}$. In contrast, with 1024 tuning frames, the majority of the mean response CFO values reach an accuracy of $[-22, 73]\,\text{Hz}$, while most of the values are accurate to $[-164, 216]\,\text{Hz}$.

In other runs of the experiment in which the DUT has an initial CFO of tens of kilohertz with respect to the WARP SDR, the median of the mean response CFO values may have a bias of a few hundred hertz after executing the frequency synchronization procedure, even if a large number of tuning frames is used. As a workaround, the frequency synchronization procedure can be executed twice in succession.

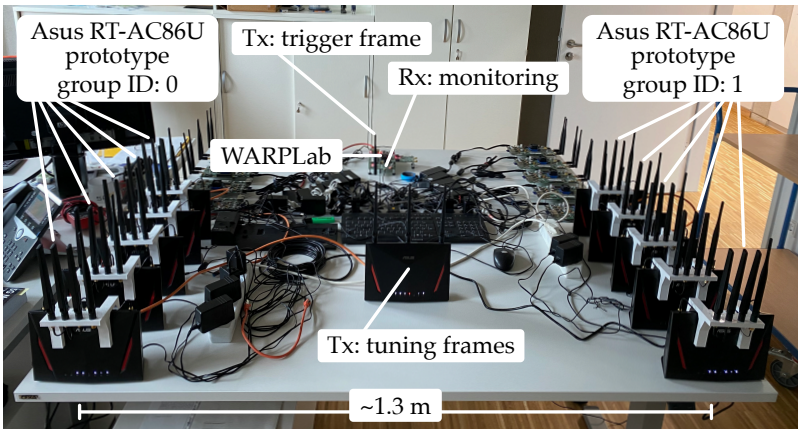


Figure 54: Testbed setup for ping-pong transmissions between two groups of five nodes

5.4 Evaluation

To assess the feasibility of CCT-based network flooding with commercial off-the-shelf IEEE 802.11 devices, testbed experiments are conducted in which two groups of Asus RT-AC86U prototype nodes generate successive CCTs with CI in a ping-pong fashion. The experiments are conducted in the same room and under the same conditions as the ping-pong experiments with the WARP prototype nodes presented in Section 4.5. The Asus RT-AC86U prototype nodes are configured to generate 10 000 successive CCTs in each experiment run in order to check if they persistently keep up a CCT-based flooding process. As each CCT triggers the generation of the next CCT by the other group, respectively, the temporal progression of performance metrics can be observed.

The experiments presented in Section 5.4.1, in Section 5.4.2 and in Section 5.4.3 are conducted in the same way as the experiments with the WARP prototype nodes presented in Section 4.5.2, in Section 4.5.3 and in Section 4.5.4, respectively. For the sake of brevity, the following subsections only highlight the main differences of the experiments with the Asus RT-AC86U prototype nodes in comparison to the respective experiments with the WARP prototype nodes and discuss the results. Furthermore, Section 5.4.4 explores the feasibility and the performance of CCT with up to four MIMO spatial streams.

5.4.1 *Meeting the URLLC Requirement*

The Ultra-Reliable Low-Latency Communications (URLLC) requirement of delivering a 32-byte MSDU with a reliability of $1 - 10^{-5}$ and with a latency of 1 ms [3] is assessed in a testbed experiment through CCT-based ping-pong transmissions with two groups of five Asus RT-AC86U prototype nodes each. The experiment is conducted in the same way as the corresponding experiment with WARP prototype nodes presented in Section 4.5.2. While the frequency synchronization of the Asus RT-AC86U prototype is typically less accurate than that of the WARP prototype, it might be sufficient to generate CCTs with CI due to the small MSDU size of 32 bytes employed according to the URLLC requirement. With this, the longest Non-HT PPDU consists of 26 OFDM symbols at MCS 0, including the PHY preamble with an equivalent length of 4 OFDM symbols (see Section 2.4.2). The experiment is conducted on channel 157 in the 5 GHz band, i.e., at 5785 MHz.

Device ID	MCS 0			MCS 1			MCS 2			MCS 3		
	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	1	1	0	1	1	0	0	0	0	0	1	0
Matlab NLOS	2	1	0	2	2	1	1	1	0	2	5	1

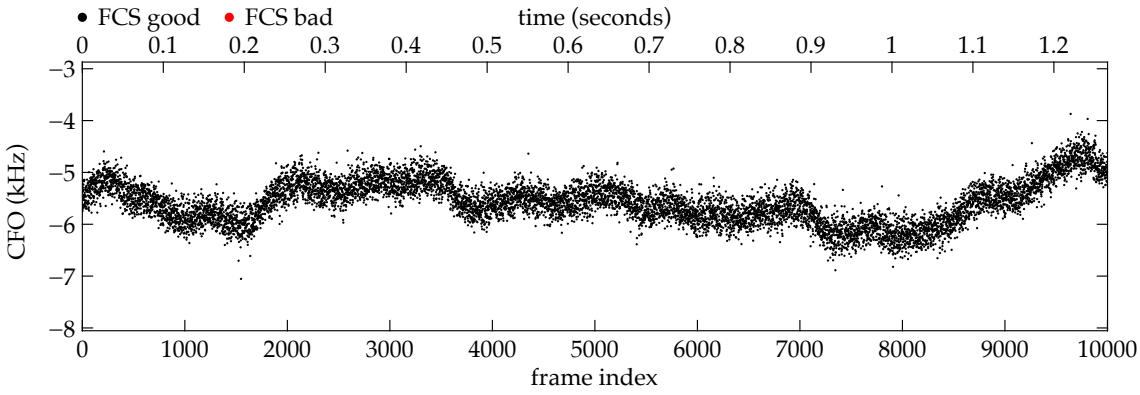
Device ID	MCS 4			MCS 5			MCS 6			MCS 7		
	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	1	1	0	1	2	0	1	4	0	3	9	1
Matlab NLOS	5	13	2	11	20	5	34	64	26	N/A	N/A	N/A

Table 8: Number of RTx repetitions required to meet the URLLC reliability requirement of $1 - 10^{-5}$

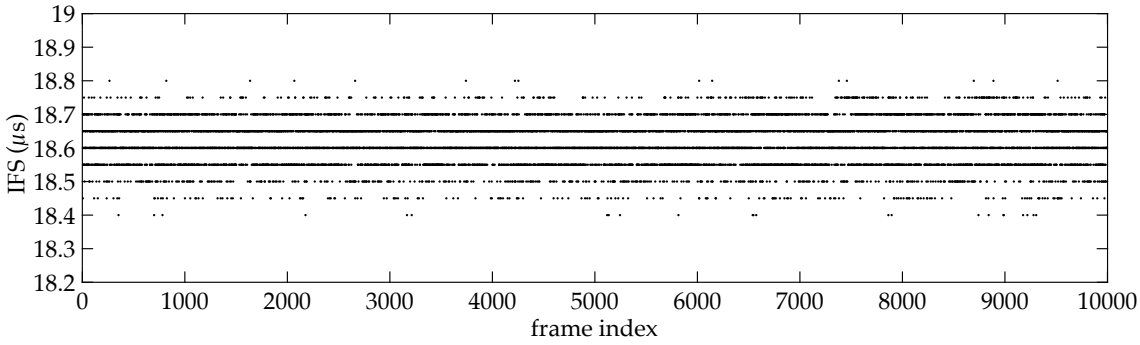
Figure 54 shows the testbed setup with five Asus RT-AC86U prototype nodes in each of both groups. The Asus RT-AC86U device in the middle of the table is used to send tuning frames during the frequency synchronization procedure so that the surrounding Asus RT-AC86U prototype nodes synchronize to the reference frequency of this device. The frequency synchronization procedure is executed twice with 1024 tuning frames at the beginning of the experiment, while several experiment runs are conducted afterwards. In each experiment run, the WARP SDR with WARPLab [250], which is in LOS, is used to trigger a flooding process with 10 000 successive CCTs and to record the received CCT signals for analysis in Matlab. In addition, a second monitoring WARP SDR with WARPLab is placed in a neighboring office at 6 m distance in non-line-of-sight (NLOS). Flooding processes with 10 000 successive CCTs can be successfully triggered in test runs for all MCS indices with a Non-HT PPDU (see Table 2), confirming that CCT-based network flooding is feasible with the Asus RT-AC86U prototype nodes. As the Non-HT PHY format employs a single spatial stream, the Asus RT-AC86U prototype nodes send the same signal on all four attached antennas, respectively. While the additional spatial diversity makes the channel state information (CSI) of the CCT signals even more frequency selective, it apparently does not impede the reception of the CCTs.

The experiment is conducted for all MCS indices with a Non-HT PPDU. For each MCS, 100 flooding processes with 10 000 CCTs are successively triggered and recorded to obtain 1 000 000 CCTs in total. The frame reception rate (FRR) is calculated with a virtual number of RTx repetitions over these 1 000 000 CCTs to emulate the tuning of the reliability through RTx repetitions in multi-hop settings, as described in Section 4.5.2.

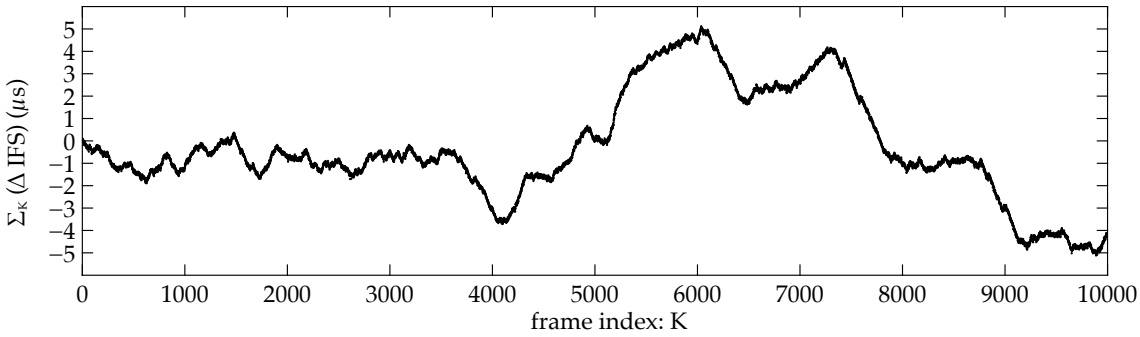
Table 19 in Appendix D lists the FRRs of both the LOS and the NLOS receiver for all MCS indices, without virtual RTx repetitions. For receptions from any of both groups, the FRR of the LOS receiver exceeds 99.999 % up to MCS 6, while the FRR of the NLOS receiver exceeds 99.999 % for MCS 0 and for MCS 2. Further, Table 8 lists the number of virtual RTx repetitions required in addition to the first flooding frame reception to meet the URLLC reliability requirement of $1 - 10^{-5}$. The fact that the reception performance of the LOS receiver is slightly better for MCS 2 than for MCS 0 indicates that the accuracy of the frequency synchronization varies a little across different executions of the experiment. Still, the results confirm that the URLLC requirement can be met.



(a) Measured CFO of CCTs



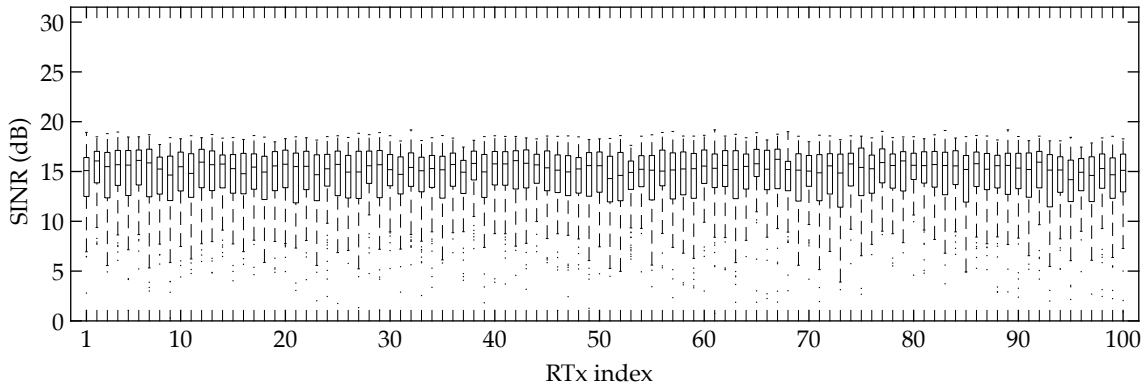
(b) Measured IFS of CCTs



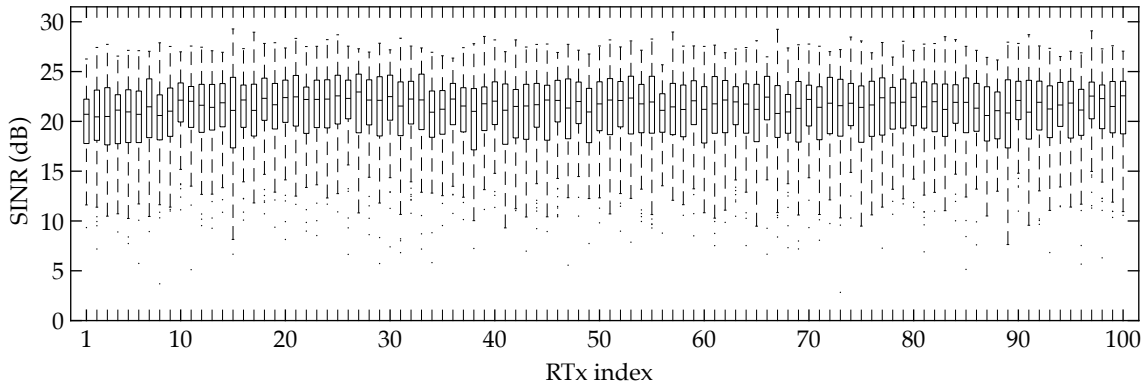
(c) Accumulation of Δ IFS: global time synchronization error with respect to the flooding initiator

Figure 55: Progression of performance metrics in 5x5 ping-pong experiment at MCS 0 over 1.2 s

Figure 55 shows the progressions of the CFO and of the IFS, exemplarily for an experiment run at MCS 0. Note that the CFO estimation in Matlab is most accurate for MCS 0 as the resulting PPDU has the largest number of OFDM symbols (see Section 3.6). Figure 55a shows that the measured CFO fluctuates by $\sim \pm 500$ Hz around a local mean value, which in turn varies by up to $\sim \pm 1$ kHz over the course of the experiment run. While the Asus RT-AC86U prototype nodes are expected to cause short-term variations, the rather slow variations of the local mean value are probably due to oscillator jitter of the monitoring WARP SDR. Figure 55b shows that the IFS ranges from 18.4 µs to 18.8 µs, so the IFS spread is narrower than the GI duration of 800 ns of the Non-HT PHY, which is a prerequisite to avoid ISI. With an accurate estimate of the average IFS $\widetilde{T}^{\text{IFS}}$, the global time synchronization error does not exceed ± 5 µs, as shown in Figure 55c. Other experiment runs with MCS 0 exhibit similar characteristics.



(a) SINR during the execution of the experiment with MCS 0

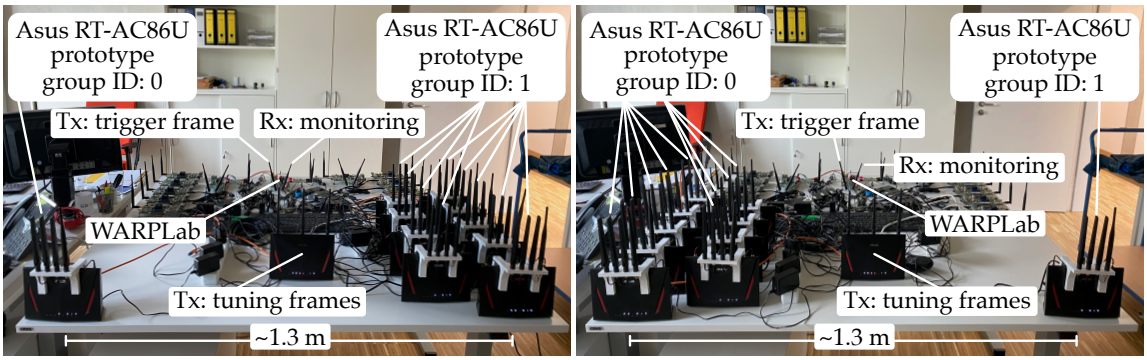


(b) SINR during the execution of the experiment with MCS 4

Figure 56: SINR of the first 100 CCTs in 5x5 ping-pong transmissions with a LOS receiver

5.4.2 *SINR over the Retransmission Index*

To investigate the evolution of the signal quality of successive CCTs over the course of a flooding process with ping-pong transmissions, the dataset of the previous experiment presented in Section 5.4.1 is processed according to the method described in Section 4.5.3, i.e., for each MCS index, the measured signal-to-interference-plus-noise ratio (SINR) values of all 100 experiment runs are considered jointly over the respective RTx indices. Figure 56 shows the measured SINR values of the first 100 CCTs of the 100 different experiment runs of ping-pong transmissions, exemplarily for the experiments conducted with MCS 0 and with MCS 4, respectively. The plots are generated with the `boxplot` function of Matlab [149]. In each of both plots, the characteristics of the SINR are quite consistent across all RTx indices, which indicates that the CCT signal quality does not change significantly after the start of a flooding process with ping-pong transmissions. However, the median SINR is $\sim$15 dB in Figure 56a, while it is $\sim$21 dB in Figure 56b. This finding cannot be attributed to the MCS itself since the SINR is independent of it. Instead, this finding indicates that the accuracy of the frequency synchronization varies a little across the different executions of the experiment with different MCS indices. Overall, the median SINR ranges from $\sim$15 dB to $\sim$23 dB across the different executions. The relatively wide range of the whiskers above and below the boxes is due to the random phase offsets (POs) between the transmitters, as described in Section 4.5.3.

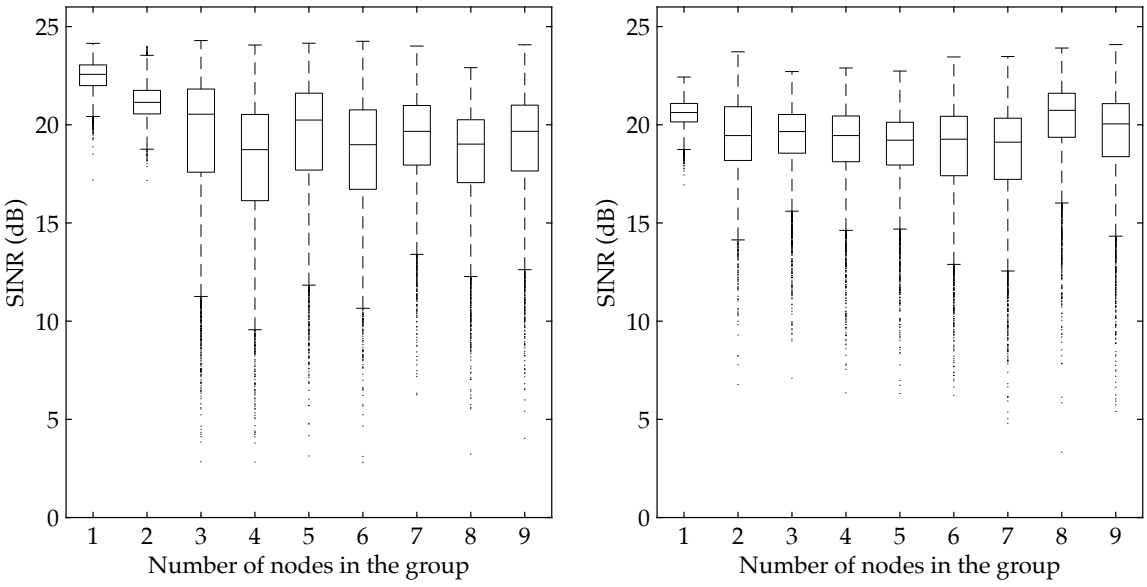


(a) Group ID 0: 1 node Group ID 1: 9 nodes (b) Group ID 0: 9 nodes Group ID 1: 1 node

Figure 57: Testbed setup with a variable number of nodes per group: (a) first and (b) last setting

5.4.3 *SINR over the Number of Transmitters*

Figure 57 shows the setup of an experiment in which the CCT signal quality is assessed as a function of the number of concurrent transmitters. This experiment is conducted in accordance with the experiment with WARP prototype nodes presented in Section 4.5.4. The frequency synchronization procedure is executed twice with 1024 tuning frames. Then, nine runs of ping-pong transmissions between both groups of nodes are executed, while the numbers of nodes in the groups are varied between the experiment runs. As in Section 4.5.4, the flooding frame is a Non-HT PPDU that encapsulates an MSDU with 32 bytes at MCS 2, so that each run of 10 000 ping-pong transmissions lasts ~ 0.8 s. Figure 58 shows the measured SINR values, visualized by means of the `boxplot` function of Matlab [149]. The SINR characteristics in Figure 58 are similar to those in Figure 46, so the conclusions of Section 4.5.4 also apply to the results of this experiment.



(a) Group ID 0 (b) Group ID 1

Figure 58: Measured SINR as a function of the number of concurrent transmitters of a CCT

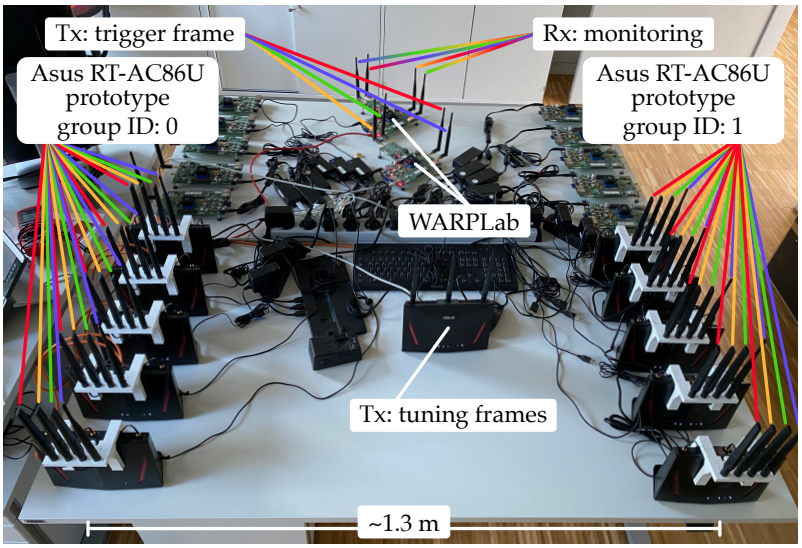


Figure 59: Testbed setup for ping-pong transmissions with up to four MIMO spatial streams

5.4.4 *Performance Gains through MIMO Spatial Streams*

Since the BCM4365E chip supports MCS 0–31 of the HT PHY, the Asus RT-AC86U prototype is employed to explore the feasibility and the performance of CCT-based network flooding also with up to four MIMO spatial streams. Note that multiple MIMO spatial streams are not supported by the WARP prototype presented in Chapter 4 since the IEEE 802.11 reference design for WARP v3 [247] only supports the Non-HT PHY and MCS 0–7 of the HT PHY. In addition, the Asus RT-AC86U prototype can operate the HT PHY with 40 MHz wide channels in the 5 GHz band, allowing the PHY data rate to be increased by more than a factor of 2 compared to a 20 MHz wide channel.

It is to be expected that accurate synchronization in time and frequency facilitates CCT with MIMO spatial streams since the superimposed signals can be decoded as usual when the signal components of the spatial streams interfere coherently with each other. In this case, the spatial diversity of the concurrent transmitters makes the CSI of each spatial stream just more frequency selective, which is not an impediment for decoding.

Figure 59 shows the testbed setup of an experiment with two groups of five Asus RT-AC86U prototype nodes each. In contrast to the testbed setup shown in Figure 54, two different WARP SDRs with WARPLab [250] are used to transmit the trigger frame and to monitor the CCTs generated by the Asus RT-AC86U prototype nodes, respectively. With this, each of both WARP SDRs can process up to four spatial streams by means of its different antennas and radio frequency (RF) chains. The plain-colored annotation lines in Figure 59 indicate the assignments of antennas to four different spatial streams during transmissions, while the multi-colored annotation lines indicate that the antennas of the monitoring WARP SDR receive different superpositions of the spatial streams. The frequency synchronization procedure is carried out twice with 1024 tuning frames at the beginning of each execution of an experiment. The following experiments are conducted with an HT-mixed format (HT-MF) PPDU with a GI of 800 ns on channel 159 in the 5 GHz band, i.e., on a 40 MHz wide channel at a center frequency of 5795 MHz. The MCS parameters are listed in Table 20 in Appendix E.

5.4.4.1 *Reception Performance for MSDUs with 32, 128 and 512 Bytes*

Testbed experiments with ping-pong transmissions between the two groups of nodes are conducted with MSDU sizes of 32, 128 and 512 bytes for all MCS indices, respectively. Each execution of an experiment with a particular MSDU size comprises the frequency synchronization procedure, followed by 100 runs of 100 successive CCTs for each MCS. On the one hand, this execution mode allows the FRR to be calculated over 10 000 received CCTs for each MCS. On the other hand, it allows the characterization of other performance metrics, such as the SINR, over the first 100 CCTs of 100 flooding processes. The IQ samples from the WARP SDR with WARPLab, which is in LOS, are processed by a decoder that is based on the Matlab wireless LAN (WLAN) toolbox [150].

Table 21, Table 22 and Table 23 in Appendix E list the FRR of the Matlab decoder for the experiments with an MSDU size of 32 bytes, 128 bytes and 512 bytes, respectively. Flooding processes with 100 successive CCTs can be triggered for all MCS indices with a 32-byte and with a 128-byte MSDU and for most MCS indices with a 512-byte MSDU, which confirms that CCT works with MIMO spatial streams. While the Matlab decoder has a poor FRR for some of the MCS indices with a high data rate, these data rates are substantially higher than the ones used in the previous experiments (cf. Table 2, Table 20). Notably, the frequency synchronization is sufficiently accurate for generating CI even with an MSDU size of 512 bytes for various MCS indices, despite certain disturbing effects on the frequency accuracy, such as oscillator jitter, as described in Section 5.3. For instance, with an MSDU size of 512 bytes and MCS 0, the PPDU length corresponds to 90 OFDM symbols of 4 µs each, including the PHY preamble and the SIGNAL field symbols with an equivalent length of 9 OFDM symbols.

Since frequency synchronization errors translate to phase errors growing over time, their impact is small when the PPDU is short. Thus, the experiment with an MSDU size of 32 bytes is best suited to identify performance gains through MIMO spatial streams. Some of the data rates of the HT PHY are provided through different MCS indices by means of different combinations of the modulation scheme, the coding rate and the number of spatial streams (see Table 20). For instance, the MCS indices 3, 9 and 24 represent a PHY data rate of 54 Mbit/s by means of 1, 2 and 4 spatial streams, respectively. Similarly, the MCS indices 5, 11 and 25 represent a PHY data rate of 108 Mbit/s by means of 1, 2 and 4 spatial streams, respectively. PPDUs generated with the MCS indices from one of these sets have DATA portions with an identical number of OFDM symbols, so that the CCT reception performance can be compared for different numbers of spatial streams under the same conditions in terms of their exposure to noise and interference. Table 21 shows that the FRR with MCS 24 is higher than the FRR with MCS 9 and that the FRR with MCS 9 is higher than the FRR with MCS 3, i.e., at a PHY data rate of 54 Mbit/s, the reception performance is better with a higher number of spatial streams. The same relation applies also at a PHY data rate of 108 Mbit/s, i.e., the FRR is higher with MCS 25 than with MCS 11 and the FRR is higher with MCS 11 than with MCS 5. However, at a PHY data rate of 162 Mbit/s, MCS 19 with 3 spatial streams has a higher FRR than MCS 12 with 2 spatial streams, whereas MCS 26 with 4 spatial streams has a slightly lower FRR than MCS 19 with 3 spatial streams.

Most of the FRR comparisons indicate that a higher number of spatial streams offers a performance gain. Still, practical experimentation also incurs distorting effects, such as oscillator jitter of the devices and fading, causing performance variations.

5.4.4.2 *Meeting the URLLC Requirement*

The URLLC requirement of delivering a 32-byte MSDU with a reliability of $1 - 10^{-5}$ and with a latency of 1 ms [3] is assessed specifically for the MCS indices {3, 9, 24} and for the MCS indices {5, 11, 25}, allowing FRR comparisons between one, two and four spatial streams at the PHY data rates of 54 Mbit/s and 108 Mbit/s, respectively. Each execution of an experiment with a particular MCS index comprises the frequency synchronization procedure, followed by 100 runs of 10 000 successive CCTs. Otherwise, the methodology of the experiment presented in Section 4.5.2 applies.

Table 24 in Appendix E lists the FRRs of the Matlab decoder, which processes the IQ samples from the monitoring WARP SDR in LOS, without virtual RTx repetitions. At both PHY data rates of 54 Mbit/s and 108 Mbit/s, the reception performance is better with a larger number of spatial streams. For receptions from any of both groups, the FRR exceeds 99.999 % with MCS 9, with MCS 24 and with MCS 25, i.e., the URLLC reliability requirement is met without additional RTx repetitions with these MCS indices. Furthermore, Table 9 lists the number of virtual RTx repetitions required in addition to the first flooding frame reception to meet the URLLC reliability requirement of $1 - 10^{-5}$. Still, the better reception performance with spatial streams comes at the expense of a slightly larger PPDU due to the additional HT long training field (HT-LTF) symbols that are employed for channel estimation, i.e., while the DATA portions of the PPDUs with different numbers of spatial streams have the same duration when the PHY data rates are identical, the entire PPDU durations differ slightly. Table 10 lists the resulting latency per hop with a 32-byte MSDU for the six MCS indices.

	HT PHY at 54 Mbit/s			HT PHY at 108 Mbit/s		
	MCS 3	MCS 9	MCS 24	MCS 5	MCS 11	MCS 25
	($N_{SS} = 1$)	($N_{SS} = 2$)	($N_{SS} = 4$)	($N_{SS} = 1$)	($N_{SS} = 2$)	($N_{SS} = 4$)
Group ID 0	2	2	1	3	2	1
Group ID 1	2	2	1	3	2	1
Any group	1	0	0	1	1	0

Table 9: Number of RTx repetitions required to meet the URLLC reliability requirement of $1 - 10^{-5}$ at the PHY data rates of 54 Mbit/s and 108 Mbit/s with $N_{SS} \in \{1, 2, 4\}$ spatial streams, with a 40 MHz wide channel and a GI of 800 ns in the HT-MF PHY format (see Table 20)

HT PHY at 54 Mbit/s			HT PHY at 108 Mbit/s		
MCS 3	MCS 9	MCS 24	MCS 5	MCS 11	MCS 25
($N_{SS} = 1$)	($N_{SS} = 2$)	($N_{SS} = 4$)	($N_{SS} = 1$)	($N_{SS} = 2$)	($N_{SS} = 4$)
64 µs	68 µs	76 µs	60 µs	64 µs	72 µs

Table 10: Latency per hop with a 32-byte MSDU in microseconds, including a SIFS of 16 µs, at the PHY data rates of 54 Mbit/s and 108 Mbit/s with $N_{SS} \in \{1, 2, 4\}$ spatial streams, with a 40 MHz wide channel and a GI of 800 ns in the HT-MF PHY format (see Table 20)

5.5 Conclusion

The presented Asus RT-AC86U prototype enables CCT-based ping-pong transmissions between two groups of nodes with the OFDM-based IEEE 802.11 Non-HT and HT PHYs in real-time. With this, CCT-based network flooding is demonstrated to be feasible also with commercial devices by means of a modified firmware.

The Asus RT-AC86U wireless router is converted into a prototype for CCT with CI through firmware modifications. In particular, the ucode is modified in order to adapt the real-time response behavior of the embedded BCM4365E chip upon frame receptions. Section 5.2.1 shows how a modified firmware can be employed to facilitate an automated reverse engineering process in which a DUT receives frames with an artificial CFO from a WARP SDR to identify PHY registers of the chip that expose its CFO estimates. In contrast, the possibility to tune the carrier frequency through the VCO register is found by trial and error, as described in Section 5.2.2. Generally, technical knowledge and intuition play an important role in reverse engineering, but custom tools and methods tailored to specific problems may help to find functionalities that are hard to reveal.

A frequency synchronization procedure is employed to pre-synchronize multiple Asus RT-AC86U prototype nodes to a common reference frequency by means of multiple tuning frames transmitted by a particular device. Afterwards, the syntonized nodes are employed in testbed experiments to generate CCTs with CI while their respective oscillators run freely. This mode of frequency synchronization is chosen for two reasons. First, the four CFO estimates of the PHY registers are quantized with a step size of 2.3 kHz. Thus, multiple tuning frames are employed to enhance the accuracy of the CFO estimation. The frequency synchronization procedure can be configured to use a specific number of tuning frames, so that its accuracy can be characterized as a function of this parameter. Second, the radio registers of the BCM4365E chip can only be set up by the embedded ARM processor through an initialization procedure, so the VCO register could not even be written by the D11 core during the processing of a received frame.

The overall accuracy of the frequency synchronization of the Asus RT-AC86U prototype depends on three factors. First, the CFO estimation accuracy depends on the number of tuning frames employed to average over the PHY register values with a quantization step size of 2.3 kHz, as described in Section 5.3.3. Second, the carrier frequency can only be tuned in steps of ~ 115 Hz by means of the VCO register, as described in Section 5.2.2. Third, the free-running oscillator is subject to jitter, so that the carrier frequency fluctuates by up to $\sim \pm 500$ Hz during individual frame transmissions, as described in Section 5.3.1. Still, since the maximum CFO-induced phase error during a transmission depends on the PPDU duration, CCT with CI is at least feasible with rather short frames.

Figure 51 shows that the measured IFS values of an Asus RT-AC86U prototype node typically range from ~ 18.4 µs to ~ 18.75 µs. Since the spread of the IFS values is narrower than the GI of 800 ns of the Non-HT PHY, ISI can be avoided in reception-triggered CCTs.

Testbed experiments confirm that two groups of five Asus RT-AC86U prototype nodes can reliably generate 10 000 successive CCTs with CI in a ping-pong fashion. In the experiment presented in Section 5.4.1, the FRR of a LOS receiver for a frame with a 32-byte MSDU is consistently better than 99 % up to Non-HT MCS 6 (see Table 2), as listed in Table 19. Further, Figure 55 shows that the quality of the measured CFO and IFS values is consistent during the CCT-based ping-pong transmissions.

The URLLC requirement of delivering a 32-byte MSDU with a reliability of $1 - 10^{-5}$ and with a latency of 1 ms [3] is met for receptions from any of both groups up to Non-HT MCS 6 under LOS conditions and with Non-HT MCS 0 and 2 under NLOS conditions, respectively. Additional RTx repetitions can be employed to meet the URLLC reliability requirement also with other parameter combinations, as listed in Table 8.

The SINR characteristics of CCTs generated by the two groups of Asus RT-AC86U prototype nodes are independent of the RTx index, i.e., the signal quality does not change significantly during CCT-based ping-pong transmissions, as shown in Figure 56. Further, the SINR characteristics of CCTs with different numbers of transmitters are also similar to each other, as shown in Figure 58. With this, the characteristics of CCTs generated by the Asus RT-AC86U prototype nodes are similar to the characteristics of CCTs generated by the WARP prototype nodes, so the corresponding conclusions presented in Section 4.6 also apply to the results of this chapter.

The Asus RT-AC86U prototype is further employed to explore the feasibility and the performance of CCT-based network flooding with multiple MIMO spatial streams. Section 5.4.4 presents testbed experiments that are conducted with MCS 0–31 of the HT PHY on a 40 MHz wide channel in the 5 GHz band, showing that CCT is not only feasible with multiple spatial streams but that spatial diversity gains can also be exploited. In particular, Table 21, Table 22 and Table 23 show that a high FRR is achieved with various MCS indices with up to four spatial streams for frames with MSDU sizes of 32 bytes, 128 bytes and 512 bytes, respectively. For receptions from any of both groups, for instance, a LOS receiver has an FRR of 99.5 % for CCTs with a 32-byte MSDU at a PHY data rate of 324 Mbit/s through MCS 28 (see Table 20 and Table 21), which is not achieved with less than four spatial streams. Besides achieving a higher PHY data rate, spatial diversity gains also allow for an improved reliability at a given PHY data rate. The results with a 32-byte MSDU are best suited to identify such gains since the resulting PPDUs are quite short, so that the impact of CFO-induced phase errors is fairly small. Considering the PHY data rates of 54 Mbit/s, 108 Mbit/s and 162 Mbit/s, Table 21 shows that a higher number of spatial streams most often results in a higher FRR.

The URLLC requirement of delivering a 32-byte MSDU with a reliability of $1 - 10^{-5}$ and with a latency of 1 ms [3] is also assessed in an experiment with MIMO spatial streams. Table 24 confirms the result that the reception performance is better with a higher number of spatial streams at the PHY data rates of 54 Mbit/s and 108 Mbit/s, respectively. For receptions from any of both groups, the URLLC reliability requirement of $1 - 10^{-5}$ is met with MCS 9, with MCS 24 and with MCS 25, respectively. Table 9 lists the number of additional RTx repetitions required to meet the URLLC reliability requirement also with other parameter combinations. Due to the additional HT-LTF symbols employed for channel estimation, PPDUs with more spatial streams have a slightly longer duration and thus a slightly larger latency per hop, as listed in Table 10.

While the Asus RT-AC86U wireless router also supports 80 MHz wide channels, the corresponding IEEE 802.11ac standard is operated by the BCM4366E chip, so further firmware modifications are required to enable CCT with an 80 MHz wide channel.

So far, the Asus RT-AC86U prototype supports CCT-based ping-pong transmissions between two groups of nodes. To enable CCT-based network flooding also over multiple hops and to support real applications, future work might comprise an integration of the low-level MAC protocol of the WARP prototype that is presented in Section 4.2.1.

Part III

CONCURRENT COOPERATIVE TRANSMISSION WITH INCOHERENT INTERFERENCE

This part presents the exploration of CCT with incoherent interference that results from a lack of accurate synchronization in time and frequency between multiple senders. Chapter 6 introduces time-variant zero-forcing (TVZF), an equalization technique that reconstructs interference patterns changing over time so that a receiver can decode CCTs that would usually cause frame loss. With this, TVZF increases a receiver's robustness to synchronization errors and facilitates CCT schemes with relaxed synchronization requirements.

6 TIME-VARIANT ZERO-FORCING

Time-variant zero-forcing (TVZF) is an equalization technique that deals with incoherent interference caused by imperfect frequency synchronization of the transmitters of a concurrent cooperative transmission (CCT). With this, TVZF relaxes the requirement for precise frequency synchronization in a CCT, enabling at least two different uses. First, TVZF enhances the robustness of syntonized CCT systems to incidental frequency synchronization errors of the concurrent transmitters. In particular, if a receiver detects different carrier frequency offsets (CFOs) in a received CCT frame, it can apply TVZF to recover the CCT frame, the reception of which would otherwise fail with high probability. Second, TVZF facilitates CCT schemes in which the transmitters cannot be wirelessly synchronized in frequency due to the lack of common frame receptions at all nodes. Such schemes may comprise but are not limited to the extensive broadcasting scenario presented in Section 1.2.2. Since TVZF relaxes the requirement for precise frequency synchronization in a CCT, the concurrent transmitters can ideally operate with the native frequencies of their respective frequency oscillators. Further, the impact of intersymbol interference (ISI) in a CCT can be absorbed by the guard interval (GI) of the employed orthogonal frequency-division multiplexing (OFDM) physical layer (PHY), since it is filled with a cyclic prefix (CP) [243]. Therefore, TVZF also offers some room for timing inaccuracies in a CCT, i.e., the time offsets (TOs) between the CCT signal components may take values within the GI duration reduced by the channel delay spread (CDS), i.e., up to 800 ns with the IEEE 802.11 Non-HT PHY [104, Section 17.3] (see Section 2.4). Particularly in case of CCTs triggered over wired connections, such as in the extensive broadcasting scenario presented in Section 1.2.2, time synchronization errors do not propagate further since each CCT is triggered separately.

While TVZF is essentially an equalization technique, it comprises several processing steps on the receiver side to reconstruct the time-variant interference over the entire bandwidth of an unsyntonized CCT frame. Hence, the term TVZF does not only refer to the equalization step itself but also to the preceding signal processing steps that facilitate the zero-forcing equalization according to the reconstructed time-variant interference.

This chapter is organized as follows. Section 6.1 discusses the impact of intercarrier interference (ICI) in a CCT with imperfect frequency synchronization of the transmitters. Section 6.2 introduces the system model of TVZF equalization. Section 6.3 presents a TVZF receiver design with several signal processing steps for unsyntonized CCTs with the IEEE 802.11 Non-HT PHY (see Section 2.4.2). In particular, Section 6.3.1 introduces an enhanced symbol timing estimation technique that ensures that the GI covers all CCT signal components. Section 6.3.2 describes the detection and parameter estimation of the individual CCT signal components. Section 6.3.3 presents an interpolation-based technique to estimate the channel coefficients of the individual CCT signal components. Section 6.3.4 shows how TVZF equalization combines the different pieces of information from the preceding processing steps. Section 6.4 presents an evaluation of the TVZF receiver design for the IEEE 802.11 Non-HT PHY in testbed experiments by means of

Wireless Open Access Research Platform (WARP) v3 software-defined radios (SDRs) [246] with combined parameter sweeps of the TO and the CFO between concurrent transmitters. The evaluation comprises experiments with two, three and four concurrent transmitters. Section 6.5 presents testbed experiments with unsyntonized CCTs from Asus RT-AC86U wireless routers operating at their native carrier frequencies. Section 6.6 explores the impact of the relative signal power levels of a CCT on the reception performance with and without TVZF, respectively. Section 6.7 provides concluding remarks on TVZF.

6.1 Intercarrier Interference

OFDM receivers typically estimate and compensate the CFO of a transmitter to avoid ICI when applying a fast Fourier transform (FFT) on received OFDM symbols [13, 183]. The IEEE 802.11 Non-HT PHY facilitates this operation through the PHY preamble, as described in Section 2.4.4 [104, Section 17.3.3]. However, in a CCT with multiple unsyntonized transmitters, the superimposed signal comprises several different CFOs. As the effects of these CFOs superimpose, the interference of the CCT is incoherent, which disallows the CFOs to be estimated through conventional processing techniques. Moreover, even if these CFOs could be estimated correctly, they could not be compensated altogether through frequency shifting of the received baseband signal. Consequently, TVZF postpones the CFO and channel estimation steps *after* the FFT-based demodulation of OFDM symbols. In doing so, TVZF deliberately introduces a certain amount of ICI.

ICI occurs in OFDM systems when a receiver is tuned to a slightly different carrier frequency than a transmitter and when it does not compensate for the CFO. In this case, the orthogonal subcarriers of the transmitter are slightly shifted in the spectrum with respect to the locations at which the receiver extracts them, so that the zero-crossings of the subcarriers on the transmitter side are not exactly aligned with the respective neighboring subcarriers on the receiver side. While a subcarrier on the receiver side still obtains the largest amount of energy from its corresponding subcarrier on the transmitter side, all the other subcarriers on the transmitter side also leak a certain amount of energy into the considered subcarrier on the receiver side since each subcarrier has a larger bandwidth than the subcarrier spacing [13, 159]. ICI generally increases with a larger CFO and its effects superimpose in a CCT with multiple transmitters.

The IEEE 802.11 standard requires devices to meet the center frequency of a 20 MHz wide channel with an accuracy of ± 20 parts per million (ppm) [104, Section 17.3.9.5], i.e., the carrier frequency of a device may deviate from the specified frequency by up to $\sim \pm 50$ kHz in the 2.4 GHz band (see Section 2.2.2). Note that such high CFOs would severely reduce the signal-to-interference-plus-noise ratio (SINR) of a CCT due to ICI. Nevertheless, some devices may have much more accurate carrier frequency oscillators. For instance, 6 different Asus RT-AC86U wireless routers are found to have CFOs in the range from -15 kHz to $+10$ kHz in the 2.4 GHz band with respect to a WARP v3 SDR. These devices are not included in Table 4 and reside at the University of Brescia in Italy. Table 3 shows that the carrier frequencies of 14 different WARP v3 SDRs [246] in the 2.4 GHz band do not deviate by more than ± 3 kHz with respect to each other, which can be attributed to their temperature-compensated crystal oscillators (TCXOs).

The impact of ICI on the SINR of received frames is assessed in simulations with a single transmitter and with an ideal channel. To this end, IEEE 802.11 Non-HT frames are

generated with a random payload by means of the custom Matlab encoder that is also used in several simulations in Chapter 3 and in the testbed experiments in Section 3.4.3. The CFO between the transmitter and the receiver is tuned to different values while the receiver's CFO compensation is disabled. Table 11 shows the average observed error vector magnitude (EVM) and the corresponding SINR values for various CFO levels. For instance, a CFO of $\Delta f = 5\,\text{kHz}$ causes an EVM of 4.68% $\approx -26.6\,\text{dB}$, with small variations of less than 0.01 % between successive frames. This amount of ICI reflects excellent conditions and still allows the IEEE 802.11 Non-HT PHY to be operated at its highest modulation and coding scheme (MCS), i.e., MCS 7 (see Table 2) [104, Table 17-17]. Therefore, a CFO of up to 5 kHz can be considered to introduce a tolerable amount of ICI that might have less impact than noise and multi-path fading effects in practice. Particularly the WARP SDRs achieve CFOs below 3 kHz due to their TCXOs (see Table 3), opening up the potential for unsyntonized CCTs with high data rates. However, also commercial devices with CFOs of up to 20 kHz might generate unsyntonized CCTs with moderate data rates that can be decoded by means of TVZF.

CFO	1 kHz	2.5 kHz	5 kHz	10 kHz	20 kHz	48 kHz
EVM	0.92%	2.37%	4.68%	9.33%	19.47%	59.88%
SINR	40.7 dB	32.5 dB	26.6 dB	20.6 dB	14.2 dB	4.5 dB

Table 11: Impact of an uncompensated CFO on the IEEE 802.11 Non-HT PHY

6.2 TVZF System Model

This section extends the PHY-independent CCT system model introduced in Section 3.2 to describe a CCT with an arbitrary number of concurrent transmitters from the perspective of a TVZF receiver. To this end, the TVZF system model introduces an index $s \in \{1, ..., S\}$ to denote the individual senders of a CCT and their respective CCT signal components. While the S different CCT signal components encode the same data frame, they may have different TOs and CFOs with respect to each other and with respect to the receiver. Hence, the TVZF system model uses a notation that assigns an index s to TO and CFO variables to indicate to which sender of a CCT these variables respectively belong to. Further, TO and CFO variables have an annotation that indicates whether the respective values relate to a transmit (Tx) reference or a receive (Rx) reference. For instance, Δf_s^{Rx} is the CFO of a transmitter with index s with respect to a TVZF receiver, whereas Δf_s^{Tx} is a CFO with respect to a common reference of all transmitters of a CCT. The latter notation expresses the CFOs between concurrent transmitters independently of a specific receiver, which is useful in the testbed experiments presented in Section 6.4. In these experiments, the common reference is the native carrier frequency of a WARP SDR [246] that drives multiple concurrent transmitters, allowing the true CFO Δf_s^{Tx} to be accurately adjusted in software through baseband processing. Moreover, the CFO Δf_s^{Rx} can also be expressed by taking the CFO $\Delta f_{\text{Tx}}^{\text{Rx}}$ between the Tx reference and the Rx reference into account:

$$\Delta f_s^{\text{Rx}} = \Delta f_{\text{Tx}}^{\text{Rx}} + \Delta f_s^{\text{Tx}} \tag{47}$$

Note that the system model introduced in Section 3.2 is designed to analyze the interference in a CCT with two concurrent transmitters and does thus not take a specific receiver into account. Hence, a CFO Δf in Chapter 3 corresponds to a CFO Δf_2^{Tx} in the TVZF system model, while the CFO Δf_1^{Tx} is 0 Hz. Further, a TO Δt in Chapter 3 corresponds to a TO Δt_2^{Tx}, while the TO Δt_1^{Tx} is 0 s. The CFO $\Delta f_{\mathrm{Tx}}^{\mathrm{Rx}}$ is not considered in Section 3.4 since it would affect both signal components equally, so it would be compensated by a phase tracking mechanism, as described in Section 3.4.2.1.

After computing the FFT of the received OFDM symbols, a CFO causes a phase drift of all subcarriers over time [159], as expressed by the last exponential factor in Equation 9. This phase drift translates to a continuous rotation of the received OFDM symbols in the in-phase quadrature (IQ) plane, as exemplarily shown for the four pilot subcarriers of an IEEE 802.11 Non-HT frame in Figure 61a. In case of a CCT, a symbol received on a subcarrier is the result of the superposition of the signal components contributed by all the concurrent transmitters on this subcarrier. Since the phases of the signal components drift with different CFOs Δf_s^{Rx}, the interference is time-variant and may alter both the phase and the amplitude of the received symbols over time, as shown for up to four concurrent transmitters in Figure 61. The vector of the received symbols $\mathbf{y}(q)$ can be expressed as a function of the OFDM symbol index q and the OFDM symbol duration T_c:

$$\mathbf{y}(q) = \sum_{s=1}^{S} \left(\mathrm{diag}(\mathbf{h}_s) \cdot \mathbf{x}(q) \cdot \mathrm{e}^{j\,2\pi\,\Delta f_s^{\mathrm{Rx}}\,(q-r)T_c} + \mathbf{n}_s^{\mathrm{Tx}} \right) + \mathbf{n}^{\mathrm{Rx}} \quad \forall\, q \in \{r, r+1, ..., Q-1\} \tag{48}$$

The vector of the transmitted symbols $\mathbf{x}(q)$ is identical for all the S senders of the CCT and contains the symbols $C_q(n)$ from Equation 9 for all used subcarriers with indices n. The vector $\mathbf{h}_s$ contains the channel coefficients of the sender with index s at the reference OFDM symbol that is used for channel training in transmissions with a single sender. With the IEEE 802.11 Non-HT PHY, the reference OFDM symbol has the index $r = 3$, pointing to the second half of the legacy long training field (L-LTF) (see Section 2.4.2), in accordance with the analysis in Section 3.4.2. The channel coefficients $\mathbf{h}_s$ drift at the pace of the CFO Δf_s^{Rx} between the sender with index s and the receiver. Note that the exponential factors depending on the TO Δt in Equation 9 do not occur in Equation 48, i.e., the effect of a frequency-dependent phase shift caused by a TO of a particular sender with index s is implicitly included in the channel coefficients $\mathbf{h}_s$. The vector $\mathbf{n}_s^{\mathrm{Tx}}$ denotes the ICI caused by the sender with index s, and $\mathbf{n}^{\mathrm{Rx}}$ is a vector of noise at the receiver. TVZF recovers the symbols $\mathbf{x}(q)$ of the q-th OFDM symbol by component-wise division:

$$\hat{\mathbf{x}}(q) = \mathrm{diag}(\mathbf{h}_\Sigma(q))^{-1} \cdot \mathbf{y}(q) \quad \forall\, q \in \{r, r+1, ..., Q-1\} \tag{49}$$

This equalization is referred to as *time-variant* since the vector $\mathbf{h}_\Sigma(q)$ varies with the OFDM symbol index q, i.e., each OFDM symbol has a specific set of channel coefficients:

$$\mathbf{h}_\Sigma(q) = \sum_{s=1}^{S} \mathbf{h}_s\, \mathrm{e}^{j\,2\pi\,\Delta f_s^{\mathrm{Rx}}\,(q-r)T_c} \quad \forall\, q \in \{r, r+1, ..., Q-1\} \tag{50}$$

The channel coefficients $\mathbf{h}_\Sigma(q)$ may appear to be highly frequency-selective, i.e., the values on neighboring subcarriers can vary widely, as illustrated exemplarily in Figure 63. While such conditions may typically require more complex equalization techniques, $\mathbf{h}_\Sigma(q)$ is the result of interfering narrow-band subcarriers that are flat-fading according to their corresponding *physical* channels. Hence, zero-forcing equalization is applicable.

Further, the TVZF system model assumes the individual channel components $\mathbf{h}_s$ and their corresponding CFOs Δf_s^{Rx} to be *invariant* over the duration of a frame transmission. The time span during which a channel component $\mathbf{h}_s$ is virtually invariant is referred to as its coherence time, which is known to be proportional to the inverse of the Doppler spread [218]. Hence, TVZF best applies in immobile environments, but it may also work in scenarios with a moderate amount of mobility. For instance, the channel coherence time is 1.61 ms at a mobility of 50 km/h [109], potentially allowing for a Non-HT frame with a PHY service data unit (PSDU) of up to 1182 bytes at MCS 0 (see Section 2.4.6). Regarding the CFOs Δf_s^{Rx}, WARP v3 SDRs [246] and Asus RT-AC86U wireless routers both have stable carrier frequencies over the duration of a frame, as shown in Section 3.5. Thus, the requirements of the TVZF system model for unsyntonized CCTs can be met by both WARP SDRs and commercial devices. The remaining challenge is to estimate the channel components $\hat{\mathbf{h}}_s$ and the CFOs $\Delta \hat{f}_s^{\mathrm{Rx}}$ for all the S senders of a CCT, for which a solution is subsequently presented for the IEEE 802.11 Non-HT PHY.

6.3 Receiver Design

This section presents a TVZF-based receiver design that decodes unsyntonized CCTs with the IEEE 802.11 Non-HT PHY [104, Section 17.3] (see Section 2.4). The TVZF receiver assumes an unknown number S of transmitters that start transmitting the same frame approximately at the same time with a TO of up to the length of the GI, which is 800 ns. Time synchronization at this scale can be well achieved with hardware timers, as shown in Section 3.1. The TVZF receiver works as follows:

1. Symbol timing estimation: The TVZF receiver determines the start of frame (SOF) by means of successive interference cancellation (SIC) on the PHY preamble to detect also weak CCT signal components that arrive early.
2. Detection and parameter estimation of CCT signal components: The TVZF receiver detects the concurrent transmitters by means of the received superposition of their signal components and estimates their respective CFOs $\Delta \hat{f}_s^{\mathrm{Rx}}$. To this end, it applies a Fourier analysis of the received symbol traces on the four pilot subcarriers in combination with a technique called iterative interference cancellation (IIC). Optionally, the result of the IIC-based Fourier analysis can be further enhanced through optimization. In the process, the TVZF receiver also obtains channel coefficient estimates $\hat{\mathbf{h}}_s^P$ on the four pilot subcarriers for each CCT signal component.
3. Interpolation of channel coefficients: The TVZF receiver generates interpolated channel coefficients $\hat{\mathbf{h}}_s^\zeta$ for each CCT signal component by leveraging the channel coefficients $\hat{\mathbf{h}}_s^P$ from the previous processing step and the channel coefficients $\tilde{\mathbf{h}}_\Sigma^{\mathrm{LTF}}$ from the L-LTF (see Section 2.4.2). Further, the interpolated channel coefficients $\hat{\mathbf{h}}_s^\zeta$ of the CCT signal components can be jointly optimized by fitting them to $\tilde{\mathbf{h}}_\Sigma^{\mathrm{LTF}}$.
4. TVZF equalization: The TVZF receiver finally reconstructs the time-variant channel coefficients $\hat{\mathbf{h}}_\Sigma(q)$, applies TVZF equalization and continues processing the frame by symbol demapping, deinterleaving, Viterbi decoding and descrambling.

The four processing steps of the TVZF receiver design are described in more detail below.

6.3.1 *SIC-Based Symbol Timing Estimation*

IEEE 802.11 receivers leverage the PHY preamble to detect an incoming frame in a stream of IQ samples from a radio frequency (RF) interface and to determine the corresponding SOF through symbol timing estimation, as described in Section 2.4.3. In the common case of a single transmitter, a signal may propagate on multiple paths from the transmitter to the receiver, so that the receiver gets multiple slightly delayed copies of the same signal. In particular, a line-of-sight (LOS) signal component takes a direct path to the receiver, whereas non-line-of-sight (NLOS) signal components may be reflected several times and may also be subject to other propagation effects like scattering and attenuation [200]. With this, a LOS signal component arrives earlier and typically with a higher power than NLOS signal components. Hence, the SOF estimate of a receiver usually points to one of the earliest arriving signal components. As the OFDM symbols after the PHY preamble are prepended by a GI with a CP (see Section 2.4.2), the signal components subject to multi-path propagation effects are typically covered by the GI, so that the OFDM symbols can be demodulated through an FFT without ISI.

However, to avoid ISI also in a CCT with multiple transmitters, a receiver must explicitly ensure that all CCT signal components are covered by the GI. Note that a weak CCT signal component may possibly arrive earlier at a receiver than a strong CCT signal component since a far transmitter may start sending slightly earlier than a near transmitter. As the SOF estimate of a conventional receiver design, as described in Section 2.4.3, points to the strongest CCT signal component, a weaker CCT signal component might not be covered by the GI and cause ISI. Therefore, the TVZF receiver design employs an enhanced symbol timing estimation to address such conditions.

A simple method to detect weak early CCT signal components would be to search also for peak pairs with a low magnitude in the output of the matched filter that detects occurrences of the long training sequence (LTS) in the received IQ samples. For instance, a receiver could search for lower peak pairs in a window of up to the length of the GI before the maximum peak pair. However, this method is found to be rather inaccurate in practical tests with WARP SDRs. Hence, the TVZF receiver design draws on SIC [233] to enhance the accuracy of the detection of weak CCT signal components.

Figure 60 illustrates four steps of the SIC-based symbol timing estimation of the TVZF receiver in a simulation with the Matlab WLAN toolbox [150]. The CCT comprises three signal components with Tx gains of 0.6, 1.0 and 0.8 and TOs of 0 ns, 200 ns and 400 ns, respectively, so that the CCT signal component that arrives first has the lowest power. All three CCT signal components are processed through a simulated TGn Model-B channel with additive white Gaussian noise (AWGN) at 30 dB signal-to-noise ratio (SNR). The TVZF receiver first determines the maximum absolute value of the matched filter output in the region in which the L-LTF is expected. In the topmost graph of Figure 60, this value is found at sample index 560. Then, the TVZF receiver calculates a dynamic detection threshold as a certain fraction of this maximum value, shown as a horizontal red line in the topmost graph of Figure 60. In practical tests with WARP SDRs, a useful value for this fraction is found to be $\sim 80\,\%$. Since the L-LTF contains two LTS symbols prepended with a large GI (see Section 2.4.2), the occurrence of a CCT signal component is expected to generate two peaks with a distance of exactly 3.2 µs. If the TVZF receiver identifies any such peak pairs that exceed the threshold in both locations, it detects the

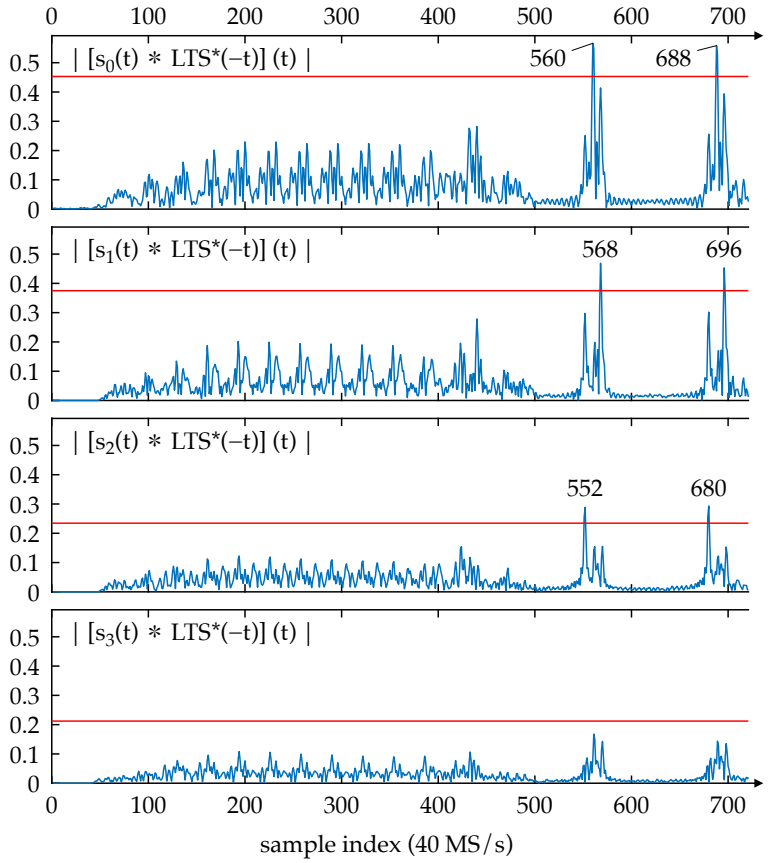


Figure 60: SIC-based symbol timing estimation: the TVZF receiver accurately detects the symbol timings of three CCT signal components by applying a dynamic threshold on the matched filter output in combination with SIC. Channel model: TGn Model-B with AWGN at 30 dB SNR; Tx gains: 0.6, 1.0, 0.8; TOs: 0 ns, 200 ns, 400 ns.

next CCT signal component at the location of the peak pair with the highest magnitude. In the top graph of Figure 60, this peak pair has the sample indices 560 and 688.

After having detected a CCT signal component, the TVZF receiver removes it from the received IQ samples through SIC, i.e., it multiplies a clean PHY preamble with the complex-valued correlation coefficient and subtracts the result from the IQ samples. Then, the TVZF receiver adapts the dynamic threshold according to the new maximum value of the matched filter output and continues searching for peak pairs. In this process, the TVZF receiver accepts an additional CCT signal component only if the largest TO does not exceed the GI duration. With this, it ensures that weaker CCT signal components are only considered if they can also be covered by the GI. In Figure 60, the TVZF receiver detects another peak pair at the sample indices 568 and 696. After that, it detects the early CCT signal component through the peak pair at the sample indices 552 and 680.

As a termination criterion, the dynamic detection threshold is prevented to fall below a certain level. Practical tests with WARP SDRs show that the TVZF receiver design is sufficiently sensitive when the detection threshold does not fall below $\sim$18 % of the maximum value of the matched filter output of the original IQ samples. In Figure 60, this fraction is 38 % for the purpose of illustration. Note that there are still small spikes in the residual IQ samples after three applications of SIC, which can be attributed to the

TGn Model-B channel taps. In fact, the number of CCT signal components detected by the SIC-based symbol timing estimation does usually not reflect the actual number of concurrent transmitters. An estimate of the actual number of CCT signal components and their corresponding parameters is obtained in the following processing step, while ISI is avoided by means of the SIC-based symbol timing estimation.

6.3.2 *IIC-Based Detection of CCT Signal Components*

After determining the SOF, the TVZF receiver removes the GI, computes an FFT of the received OFDM symbols and extracts the symbol traces of the pilot subcarriers by reverting the polarity of the fixed binary phase-shift keying (BPSK) sequence that they encode [104, Section 17.3.5.10]. Figure 61 shows examples of such symbol traces with up to four concurrent transmitters. The measured symbol traces on the four pilot subcarriers with the indices $P = \{-21, -7, 7, 21\}$ are technically time-variant channel coefficients:

$$\tilde{\mathbf{h}}_{\Sigma}^{P}(q) \simeq \sum_{s=1}^{S} \mathbf{h}_{s}^{P} \, \mathrm{e}^{j\,2\pi\,\Delta f_{s}^{\mathrm{Rx}}\,(q-r)T_{c}} \quad \forall\, q \in \{r, r+1, ..., Q-1\} \tag{51}$$

For each of the senders with an index $s \in \{1, ..., S\}$, the vector $\mathbf{h}_{s}^{P}$ consists of four channel coefficients h_{s}^{p}, each of which represents one of the four pilot subcarriers with an index $p \in P$. Note that all channel coefficients contained in $\mathbf{h}_{s}^{P}$ are equally affected by the CFO $\Delta f_{s}^{\mathrm{Rx}}$ between the respective sender and the receiver. Further, a channel coefficient h_{s}^{p} can be separated into an amplitude $a_{s}^{p} = \left|h_{s}^{p}\right|$ and a phase $\phi_{s}^{p} = \angle h_{s}^{p}$.

Next, the TVZF receiver uses the symbol traces $\tilde{\mathbf{h}}_{\Sigma}^{P}(q)$ to compute the estimates $\Delta\hat{f}_{s}^{\mathrm{Rx}}$ and $\hat{h}_{s}^{p}$ for each detected CCT signal component with an index $s \in \{1, ..., \hat{S}\}$ and for each pilot subcarrier index $p \in P$, respectively. To this end, it performs a Fourier analysis of the symbol traces $\tilde{\mathbf{h}}_{\Sigma}^{P}(q)$. Note that it may seem unusual to perform a Fourier analysis on frequency-domain data. The rationale is that $\tilde{\mathbf{h}}_{\Sigma}^{P}(q)$ is time-variant and changes as a function of discrete frequencies that shall be estimated. Since there is a chance that several concurrent transmitters have the same CFO $\Delta f_{s}^{\mathrm{Rx}}$ or very close CFOs, the estimates denoted as $\Delta\hat{f}_{s}^{\mathrm{Rx}}$ and $\hat{h}_{s}^{p}$ may potentially cover multiple CCT signal components with indices $\mathbb{S}_{s} \subseteq \{1, ..., S\}$. The estimates $\Delta\hat{f}_{s}^{\mathrm{Rx}}$ and $\hat{h}_{s}^{p}$ are correspondingly indexed according to the number of detected CCT signal components $\hat{S}$:

$$\begin{aligned} \Delta\hat{f}_{s}^{\mathrm{Rx}} &\simeq \Delta f_{\sigma}^{\mathrm{Rx}} \quad \forall\, \sigma \in \mathbb{S}_{s} \quad \forall\, s \in \{1, ..., \hat{S}\} \\ \hat{h}_{s}^{p} &\simeq \sum_{\sigma \in \mathbb{S}_{s}} h_{\sigma}^{p} \quad \forall\, s \in \{1, ..., \hat{S}\} \end{aligned} \tag{52}$$

In the Fourier analysis, the TVZF receiver leverages the different realizations of the CFOs $\Delta f_{s}^{\mathrm{Rx}}$ on the pilot subcarriers by combining the FFTs of the symbol traces $\tilde{\mathbf{h}}_{\Sigma}^{P}(q)$. To extract the estimates $\Delta\hat{f}_{s}^{\mathrm{Rx}}$ and $\hat{h}_{s}^{p}$, it uses a periodogram with a rectangular window function according to the length $Q - r$ of the symbol traces. In doing so, each CFO is represented as a *sinc* pulse in the FFT spectrum. Note that such a *sinc* pulse is wider for shorter frames, which has the undesired effect that multiple *sinc* pulses can overlap in the FFT spectrum such that stronger CCT signal components mask weaker ones. Therefore,

Algorithm 1 Detection and parameter estimation of CCT signal components through IIC

1: $\eta \leftarrow$ relative detection threshold (e.g.: 0.18)
2: $\kappa \leftarrow$ number of IIC cycles (e.g.: 4)
3: $s \leftarrow 0$ (*count of detected CCT signal components*)
4: $\tilde{h}_{\Sigma}^{p}(q) \leftarrow$ received pilot symbol traces $(q = r, r+1, ..., Q-1) \quad \forall\, p \in P$
5: $\tilde{H}_{\Sigma}^{p}(jf) \leftarrow \text{fft}\left(\tilde{h}_{\Sigma}^{p}(q)\right) \quad \forall\, p \in P$
6: $\theta = \eta \cdot \max\left(\sum_{p \in P} \left|\tilde{H}_{\Sigma}^{p}(jf)\right|\right)$
7: $\tilde{G}_{s+1}^{p}(jf) \leftarrow \tilde{H}_{\Sigma}^{p}(jf) \quad \forall\, p \in P$
8: **while** $\max\left(\sum_{p \in P} \left|\tilde{G}_{s+1}^{p}(jf)\right|\right) \geq \theta$ **do**
9: $\quad s \leftarrow s+1$ (*next CCT signal component detected*)
10: $\quad [m_s, \Delta\hat{f}_{s}^{\text{Rx}}] \leftarrow \max\left(\sum_{p \in P} \left|\tilde{G}_{s}^{p}(jf)\right|\right)$
11: $\quad \hat{a}_{s}^{p} = \frac{1}{Q-r}\left|\tilde{G}_{s}^{p}(jf)\right|_{f=\Delta\hat{f}_{s}^{\text{Rx}}}\left| \quad \forall\, p \in P\right.$
12: $\quad \hat{\phi}_{s}^{p} = \angle\, \tilde{G}_{s}^{p}(jf)\Big|_{f=\Delta\hat{f}_{s}^{\text{Rx}}} \quad \forall\, p \in P$
13: $\quad \hat{h}_{s}^{p} = \hat{a}_{s}^{p} \mathrm{e}^{j\hat{\phi}_{s}^{p}} \quad \forall\, p \in P$
14: $\quad$ **for** $k = 1 \rightarrow \kappa$ **do** (*IIC optimization of parameter estimates*)
15: $\quad\quad$ **for** $l = 1 \rightarrow s$ **do** (*for each detected CCT signal component*)
16: $\quad\quad\quad \tilde{G}_{l}^{p}(jf) \leftarrow \tilde{H}_{\Sigma}^{p}(jf) - \text{fft}\left(\sum_{\substack{v=1 \\ v \neq l}}^{s} \hat{h}_{v}^{p} \mathrm{e}^{j2\pi\, \Delta\hat{f}_{v}^{\text{Rx}}\,(q-r)T_c}\right) \quad \forall\, p \in P$
17: $\quad\quad\quad [m_l, \Delta\hat{f}_{l}^{\text{Rx}}] \leftarrow \max\left(\sum_{p \in P} \left|\tilde{G}_{l}^{p}(jf)\right|\right)$
18: $\quad\quad\quad \hat{a}_{l}^{p} = \frac{1}{Q-r}\left|\tilde{G}_{l}^{p}(jf)\right|_{f=\Delta\hat{f}_{l}^{\text{Rx}}}\left| \quad \forall\, p \in P\right.$
19: $\quad\quad\quad \hat{\phi}_{l}^{p} = \angle\, \tilde{G}_{l}^{p}(jf)\Big|_{f=\Delta\hat{f}_{l}^{\text{Rx}}} \quad \forall\, p \in P$
20: $\quad\quad\quad \hat{h}_{l}^{p} = \hat{a}_{l}^{p} \mathrm{e}^{j\hat{\phi}_{l}^{p}} \quad \forall\, p \in P$
21: $\quad \tilde{G}_{s+1}^{p}(jf) \leftarrow \tilde{H}_{\Sigma}^{p}(jf) - \text{fft}\left(\sum_{v=1}^{s} \hat{h}_{v}^{p} \mathrm{e}^{j2\pi\, \Delta\hat{f}_{v}^{\text{Rx}}\,(q-r)T_c}\right) \quad \forall\, p \in P$
22: Return $\Delta\hat{f}_{s}^{\text{Rx}}$ and $\hat{h}_{s}^{p} \quad \forall\, s \in \{1, ..., \hat{S}\} \quad \forall\, p \in P$

the TVZF receiver employs SIC to cancel CCT signal components in $\tilde{\mathbf{h}}_{\Sigma}^{P}(q)$ after detecting them, allowing it to detect also weaker CCT signal components. However, when applying SIC successively, the estimates of the stronger CCT signal components, which are first detected, are still subject to interference by the weaker CCT signal components, leading to degradations of all estimates. Hence, the TVZF receiver employs a variation of SIC that is subsequently referred to as iterative interference cancellation (IIC).

Algorithm 1 gives a formalized description of the IIC-based Fourier analysis of $\tilde{\mathbf{h}}_{\Sigma}^{P}(q)$. The TVZF receiver computes an FFT of each of the four pilot symbol traces (lines 4 and 5). Further, the TVZF receiver defines a relative detection threshold η (line 1) by which it calculates an absolute detection threshold θ as a function of the sum of the absolute FFT spectra of the pilot symbol traces (line 6), allowing for variations of the received signal strength (RSS) across different CCT frame receptions. A lower value of η makes the detection of CCT signal components more sensitive but may also lead to false detections. Practical tests with WARP SDRs show that a value of $\eta \simeq 18\,\%$ provides an accurate

detection performance in various settings. Further, the number of IIC cycles κ (line 2) can be adjusted to tune the tradeoff between processing speed and the parameter estimation accuracy. The number of detected CCT signal components is initialized to $s = 0$ (line 3). The TVZF receiver maintains a set of FFT spectra to detect the next CCT signal component, which initially contains the original FFT spectra of the pilot symbol traces (line 7).

When the sum of the absolute FFT spectra exceeds the threshold θ (line 8), the TVZF receiver increments the counter s (line 9) and processes the newly detected CCT signal component. To this end, it extracts the CFO $\Delta\hat{f}_s^{\mathrm{Rx}}$ by means of the location of the maximum of the combined FFT spectrum (line 10). It further extracts the magnitude $\hat{a}_s^p$ (line 11) and the phase $\hat{\phi}_s^p$ (line 12) of the detected CCT signal component for each pilot subcarrier from the corresponding FFT spectrum at the location of the CFO $\Delta\hat{f}_s^{\mathrm{Rx}}$. The latter two values are also combined to a complex-valued channel coefficient $\hat{h}_s^p$ (line 13).

After processing a newly detected CCT signal component, the TVZF receiver optimizes the parameter estimates of all CCT signal components through IIC to mitigate the impact of interference by the newly detected CCT signal component. This optimization is executed κ times (line 14) for each CCT signal component (line 15). For each considered CCT signal component with index l, the TVZF receiver reconstructs the contributions of all other CCT signal components and subtracts them from the received pilot symbol traces (line 16). Then, it refines the CFO $\Delta\hat{f}_l^{\mathrm{Rx}}$ (line 17), the magnitude $\hat{a}_l^p$ (line 18), the phase $\hat{\phi}_l^p$ (line 19) and the complex-valued channel coefficient $\hat{h}_l^p$ (line 20).

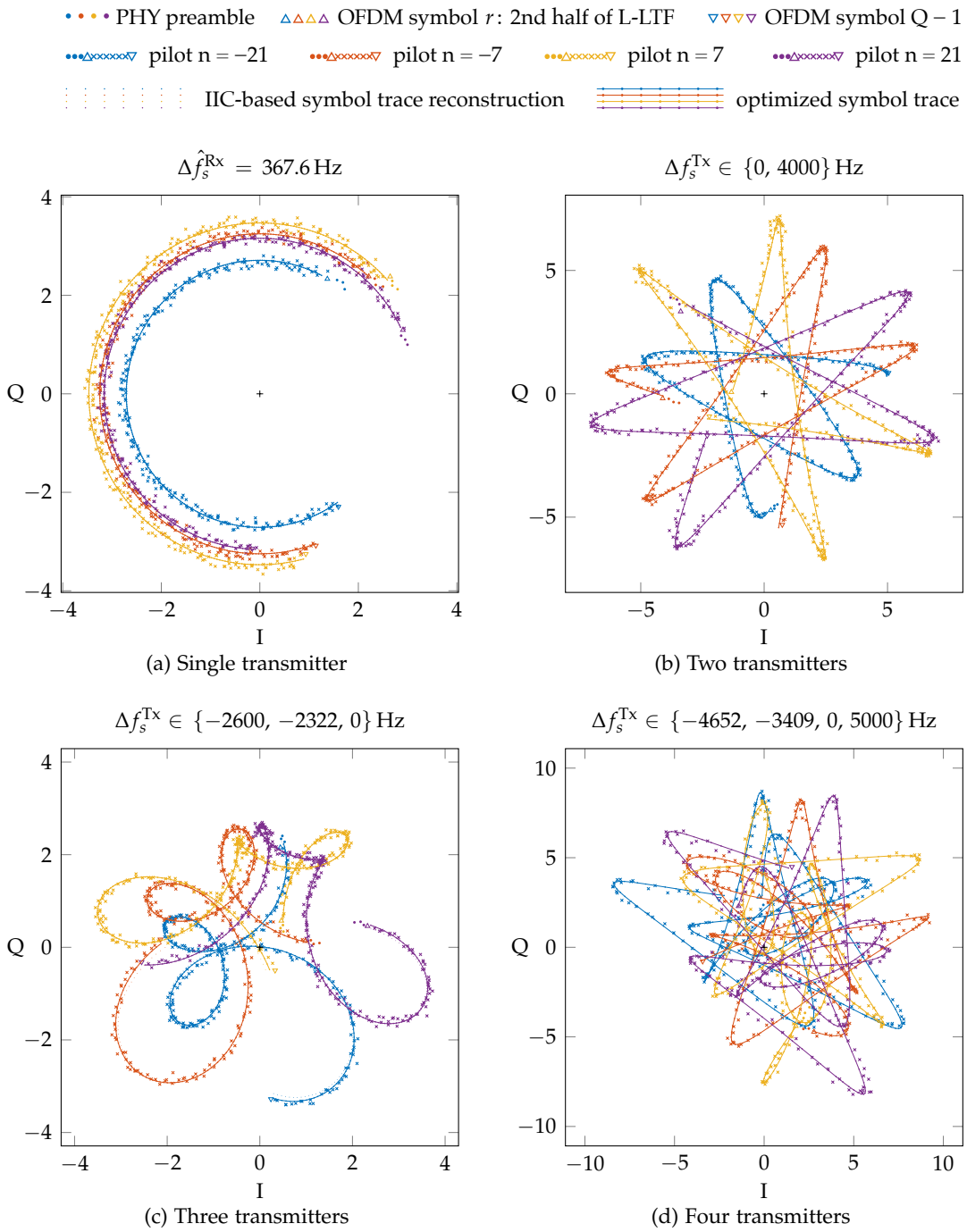
After κ iterations of IIC (line 14), the TVZF receiver reconstructs all CCT signal components and subtracts them from the FFT spectra of the received pilot symbol traces to obtain the set of FFT spectra for the next detection cycle (line 21). When no further CCT signal components are detected (line 8), Algorithm 1 returns the $\hat{S}$ CFOs $\Delta\hat{f}_s^{\mathrm{Rx}}$ and the $\hat{S} \cdot |P|$ channel coefficients $\hat{h}_s^p$ of the detected CCT signal components (line 22). With this, the symbol traces $\tilde{\mathbf{h}}_{\Sigma}^{P}(q)$ of the pilot subcarriers P can be reconstructed as:

$$\hat{\mathbf{h}}_{\Sigma}^{P}(q) = \sum_{s=1}^{\hat{S}} \hat{\mathbf{h}}_s^P \mathrm{e}^{j\,2\pi\,\Delta\hat{f}_s^{\mathrm{Rx}}\,(q-r)T_c} \quad \forall\, q \in \{r, r+1, ..., Q-1\} \tag{53}$$

To assess the accuracy of IIC, the CFO estimates $\Delta\hat{f}_s^{\mathrm{Rx}}$ and the channel estimates $\hat{h}_s^p$ are further optimized by fitting them to the received channel coefficient traces $\tilde{h}_{\Sigma}^{p}(q)$ by means of the least squares method, i.e., by minimizing the mean square error (MSE):

$$\mathrm{MSE}_{\mathbf{h}} = \frac{1}{|P|} \sum_{p \in P} \frac{1}{Q-r} \sum_{q=r}^{Q-1} \left| \hat{h}_{\Sigma}^{p}(q) - \tilde{h}_{\Sigma}^{p}(q) \right|^2 \tag{54}$$

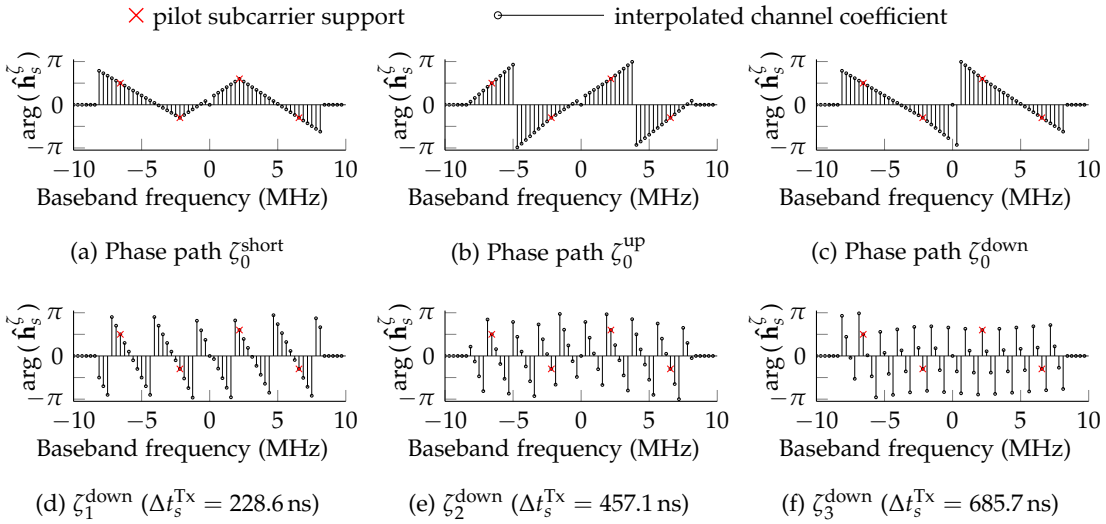
The MSE is minimized by means of the downhill simplex algorithm in Matlab, which is also known as the Nelder-Mead method and which works without derivatives [122]. While this optimization is time-consuming and impractical for real-time designs, it allows assessing the accuracy of the parameter estimation through Algorithm 1. Test runs show that the optimization leads to rather insignificant improvements in most cases, which indicates a good estimation performance of IIC. Figure 61 presents the values obtained through IIC as thin dotted lines, whereas the optimized values are shown as solid lines. In most regions of the four subplots, the thin dotted lines are hidden behind the solid lines, which confirms the effectiveness of IIC. Small differences are visible in Figure 61c.



(a) Single transmitter (b) Two transmitters

(c) Three transmitters (d) Four transmitters

Figure 61: Symbol traces of the four pilot subcarriers in CCTs with WARP SDRs: the received symbol traces $\tilde{\mathbf{h}}_{\Sigma}^{P}(q)$ are analyzed and reconstructed as $\hat{\mathbf{h}}_{\Sigma}^{P}(q)$ through IIC for CCT frames with $Q = 261$ OFDM symbols. The optimized symbol trace reconstructions mostly cover the corresponding IIC-based reconstructions, indicating good estimation performance of IIC. (a) $\Delta\hat{f}_s^{\mathrm{Rx}}$ is the estimated CFO between a single transmitter and the receiver, whereas (b–d) Δf_s^{Tx} are the true CFOs between concurrent transmitters.



(a) Phase path ζ_0^{short} (b) Phase path ζ_0^{up} (c) Phase path ζ_0^{down}

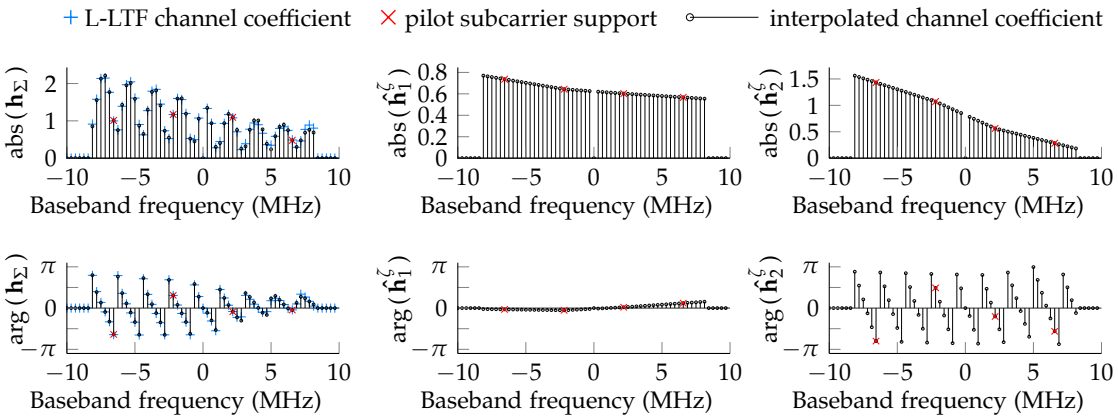
(d) ζ_1^{down} ($\Delta t_s^{\text{Tx}} = 228.6\,\text{ns}$) (e) ζ_2^{down} ($\Delta t_s^{\text{Tx}} = 457.1\,\text{ns}$) (f) ζ_3^{down} ($\Delta t_s^{\text{Tx}} = 685.7\,\text{ns}$)

Figure 62: The phase values of the four subcarrier support points can be interpolated in various ways: in a flat-fading channel, one of the phase paths (a) ζ_0^{short}, (b) ζ_0^{up}, or (c) ζ_0^{down} is almost always a good match. A TO Δt_s^{Tx} introduces a frequency-selective phase shift, causing additional phase rotations by 2π between neighboring support points (d–f).

6.3.3 *Interpolation of Channel Coefficients*

After detecting the CCT signal components and estimating their respective parameters, the TVZF receiver uses the channel coefficients $\hat{\mathbf{h}}_s^P$ of the pilot subcarriers as support points to interpolate and extrapolate the channel coefficients of the data subcarriers. The difficulty is that the complex-valued channel coefficients around the four support points can be interpolated in various ways. The TVZF receiver interpolates the amplitude and the phase linearly and separately from each other as this method yields the best results in test experiments, as compared to other interpolation methods, such as spline interpolation in the complex plane. While linear interpolation does not generate smooth curves, it also does not generate undesired deflections, as spline and cubic interpolation occasionally do. In addition, the linear interpolation may serve as a starting point for further optimizations of the channel coefficients of the CCT signal components.

While there is only one possibility to connect the magnitude values of the pilot subcarrier support points through linear interpolation, the corresponding phase values can be interpolated in various different ways since the phase is an angular quantity. Without additional phase rotations by multiples of 2π, a direct connection between two phase values can take two different directions. Hence, there are $2^3 = 8$ ways to connect the four pilot subcarrier support points. However, practical tests show that there are three patterns that almost always match with the channel coefficients of physical channels. These patterns are subsequently called phase paths and are illustrated in Figure 62a–c. Figure 62a shows the phase path ζ_0^{short} which connects two neighboring support points such that the unwrapped angular distance between them is less than or equal to π. Figure 62b shows the phase path ζ_0^{up} which connects two neighboring support points such that the unwrapped phase at the higher frequency takes a higher value than the phase value at the lower frequency. Figure 62c shows the phase path ζ_0^{down} which connects two neighboring support points in the opposite direction as ζ_0^{up}.



(a) Congruence of $\tilde{\mathbf{h}}_{\Sigma}^{\text{LTF}}$ and $\hat{\mathbf{h}}_{\Sigma}^{\zeta}$ (b) $\hat{\mathbf{h}}_{1}^{\zeta}$ ($\Delta t_1^{\text{Tx}} = 0\,\text{ns}$) (c) $\hat{\mathbf{h}}_{2}^{\zeta}$ ($\Delta t_2^{\text{Tx}} = 550\,\text{ns}$)

Figure 63: Decomposition of superimposed channel coefficients: the TVZF receiver estimates a channel component $\hat{\mathbf{h}}_s^{\zeta}$ for each CCT signal component. The sum of the channel components $\hat{\mathbf{h}}_{\Sigma}^{\zeta} = \hat{\mathbf{h}}_1^{\zeta} + \hat{\mathbf{h}}_2^{\zeta}$ is congruent with the L-LTF-based channel estimate $\tilde{\mathbf{h}}_{\Sigma}^{\text{LTF}}$.

In addition, a CCT signal component with a TO with respect to the SOF may have a phase path with phase rotations of a multiple of 2π between the subcarrier support points. This effect occurs since a TO causes a frequency-dependent phase shift, as expressed by the second exponential factor of the second CCT signal component in Equation 9. In particular, a phase rotation of 2π between two pilot subcarriers corresponds to a TO of $\Delta t = 228.6\,\text{ns}$, as illustrated in Figure 22b and c. Thus, to address a TO of up to the GI duration of 800 ns, $\lfloor \frac{800\text{ns}}{228.6\text{ns}} \rfloor = 3$ additional phase slopes have to be taken into account. Note that the fractional part of the division is already covered by the phase path ζ_0^{down}. These additional phase slopes are illustrated in Figure 62d–f for the phase path ζ_0^{down} and are denoted as ζ_1^{down}, ζ_2^{down} and ζ_3^{down}.

The TVZF receiver has to decide which phase path and which phase slope to apply for each of the $\hat{S}$ sets of pilot subcarrier support points $\hat{\mathbf{h}}_s^{P}$, respectively, to obtain the interpolated channel coefficients $\hat{\mathbf{h}}_s^{\zeta}$ for all subcarriers. To this end, it searches through various combinations of phase paths and slopes and compares the sum of the interpolated channel coefficients $\hat{\mathbf{h}}_s^{\zeta}$ with the superposition $\tilde{\mathbf{h}}_{\Sigma}^{\text{LTF}}$ from the L-LTF:

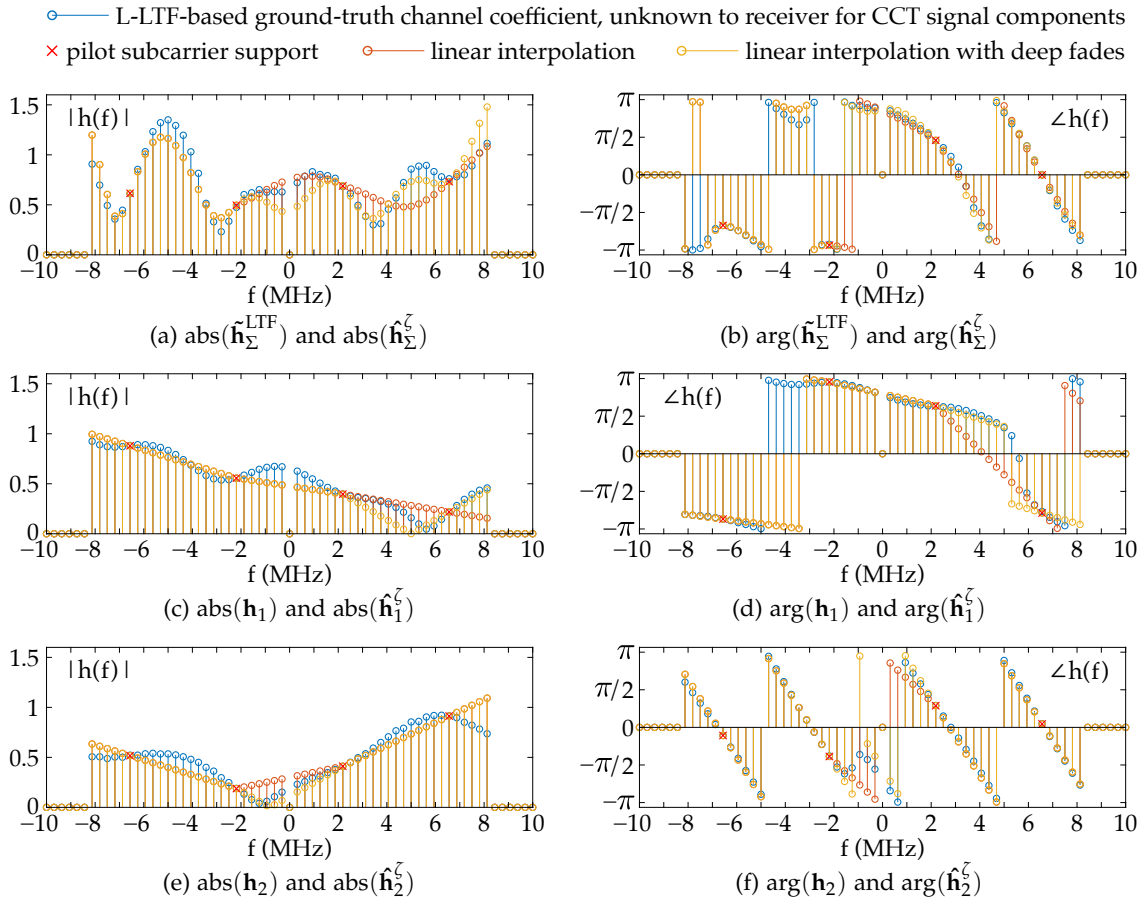
$$\text{MSE}_{\text{LTF}} = \frac{1}{52} \sum_{\substack{n=-26 \\ n\neq 0}}^{26} \left| \tilde{\mathbf{h}}_{\Sigma}^{\text{LTF}}[n] - \sum_{s=1}^{\hat{S}} \hat{\mathbf{h}}_s^{\zeta}[n] \right|^2 \tag{55}$$

The search space of this procedure grows exponentially with the number $\hat{S}$ of detected CCT signal components. However, since $\zeta_i^{\text{down}} = \zeta_{i+1}^{\text{up}}$ holds, the search space can be reduced to two phase paths with 4 and 5 phase slopes, respectively. Optionally, the interpolated channel coefficients $\hat{\mathbf{h}}_s^{\zeta}$ can be jointly optimized so that their sum approaches the superposition $\tilde{\mathbf{h}}_{\Sigma}^{\text{LTF}}$ from the L-LTF by making readjustments at the subcarrier indices $n \in \{-26, -14, 0, 14, 26\}\ \forall\ s \in \{1, ..., \hat{S}\}$ through the downhill simplex algorithm [122]. However, this joint optimization can potentially also lead to degradations.

Figure 63 illustrates the estimation of the channel components $\hat{\mathbf{h}}_s^{\zeta}$ in a CCT with two concurrent transmitters in which the second CCT signal component is delayed by $\Delta t_2^{\mathrm{Tx}} = 550\,\mathrm{ns}$ with respect to the first one. Figure 63b shows that the phase values of the estimated channel component $\hat{\mathbf{h}}_1^{\zeta}$ are quite flat, indicating that the TVZF receiver is synchronized in time with the earlier CCT signal component. Therefore, the SIC-based symbol timing estimation, as described in Section 6.3.1, is effective. On the contrary, Figure 63c shows that the estimated channel component $\hat{\mathbf{h}}_2^{\zeta}$ has a rather steep phase slope, indicating that the TVZF receiver recognizes the unknown TO Δt_2^{Tx}. Figure 63a shows that the sum of both channel components $\hat{\mathbf{h}}_{\Sigma}^{\zeta}$ is congruent with the L-LTF-based channel estimate $\tilde{\mathbf{h}}_{\Sigma}^{\mathrm{LTF}}$, indicating good estimation performance of the CCT channel components and their respective parameters by the TVZF receiver.

In addition to linear interpolation of the separated magnitude and phase values of the pilot subcarrier support points, the TVZF receiver can also address deep fades occurring on one or several OFDM subcarriers. Such deep fades particularly occur in environments with multi-path propagation effects. In contrast to the temporal deep fades of a CCT analyzed in Section 3.4.2, the spectral deep fades are static and affect a narrow subset of neighboring subcarriers. Still, a spectral deep fade also causes a phase jump of π in the channel coefficients of the neighboring subcarriers at lower and higher frequencies. To model a deep fade, the TVZF receiver interpolates the channel coefficients such that a zero-crossing with a phase jump of π occurs between two subcarrier support points. To this end, the TVZF receiver inverts the sign of the magnitude values of specific pilot subcarrier support points while shifting the corresponding phase values by π. To detect deep fades in the channel components of a CCT, the TVZF receiver searches through various combinations with and without a deep fade between any two pilot subcarrier support points for each of the $\hat{S}$ CCT signal components and selects the combination of channel components $\hat{\mathbf{h}}_s^{\zeta}$ that minimizes the MSE in Equation 55.

Figure 64 illustrates an example of a CCT with two transmitters in which each of both CCT signal components is processed through a simulated High Throughput Task Group (TGn) Model-C channel in Matlab [150]. Figure 64c shows that the first channel component $\mathbf{h}_1$ has a deep fade on the subcarrier with index $n = 18$, i.e., at 5625 kHz, whereas Figure 64e shows that the second channel component $\mathbf{h}_2$ has a deep fade on the subcarrier with the index $n = -4$, i.e., at -1250 kHz. The conventional linear interpolation clearly deviates from the true channel coefficients around the deep fades since it connects the magnitude of the pilot subcarrier support points directly with each other, whereas the linear interpolation with deep fades matches better with the true channel coefficients. The PSDU of the CCT frame has 1536 bytes and is decoded with 9 bit errors by the conventional linear interpolation, whereas the interpolation with deep fades recovers the frame correctly without bit errors. While this example shows that the TVZF receiver can deal with deep fades, it also shows that the interpolation-based channel estimation is not ideal for frequency-selective fading since the estimated channel coefficients deviate from the true channel coefficients on some subcarriers. Generally, the TVZF receiver might work best in flat-fading environments due to the interpolation-based channel estimation but it can still address deep fades that may occasionally occur also in such environments.



(a) abs($\tilde{\mathbf{h}}_{\Sigma}^{\mathrm{LTF}}$) and abs($\hat{\mathbf{h}}_{\Sigma}^{\zeta}$)
(b) arg($\tilde{\mathbf{h}}_{\Sigma}^{\mathrm{LTF}}$) and arg($\hat{\mathbf{h}}_{\Sigma}^{\zeta}$)
(c) abs($\mathbf{h}_1$) and abs($\hat{\mathbf{h}}_1^{\zeta}$)
(d) arg($\mathbf{h}_1$) and arg($\hat{\mathbf{h}}_1^{\zeta}$)
(e) abs($\mathbf{h}_2$) and abs($\hat{\mathbf{h}}_2^{\zeta}$)
(f) arg($\mathbf{h}_2$) and arg($\hat{\mathbf{h}}_2^{\zeta}$)

Figure 64: Interpolation-based channel component estimation with and w/o deep fade detection: each CCT signal component is processed with a simulated TGn Model-C channel.

6.3.4 *TVZF Equalization*

The TVZF receiver has $\hat{S}$ channel estimates $\hat{\mathbf{h}}_s^{\zeta}$ with channel coefficients on all subcarriers and $\hat{S}$ corresponding CFO estimates $\Delta\hat{f}_s^{\mathrm{Rx}}$. It finally combines these estimates to compute the time-variant channel coefficients for equalization, in accordance to Equation 50:

$$\hat{\mathbf{h}}_{\Sigma}(q) = \sum_{s=1}^{\hat{S}} \hat{\mathbf{h}}_s^{\zeta}\, \mathrm{e}^{j\,2\pi\,\Delta\hat{f}_s^{\mathrm{Rx}}\,(q-r)T_c} \quad \forall\, q \in \{r, r+1, ..., Q-1\} \tag{56}$$

6.4 Testbed Evaluation: CCTs with Controlled TOs and CFOs

To assess the CCT reception performance of the TVZF receiver, testbed experiments with combined parameter sweeps of the TO and the CFO are conducted with up to four concurrent transmitters. The Tx antennas are attached to a common WARP SDR [246] to leverage the shared clock so that the TOs Δt_s^{Tx} and the CFOs Δf_s^{Tx} of the senders with indices $s \in \{1, ..., S\}$ can be precisely tuned in software through baseband processing. IEEE 802.11 Non-HT frames are generated with the custom Matlab encoder also used for various simulations in Section 3.4 and for the CCT testbed experiments in Section 3.4.3.

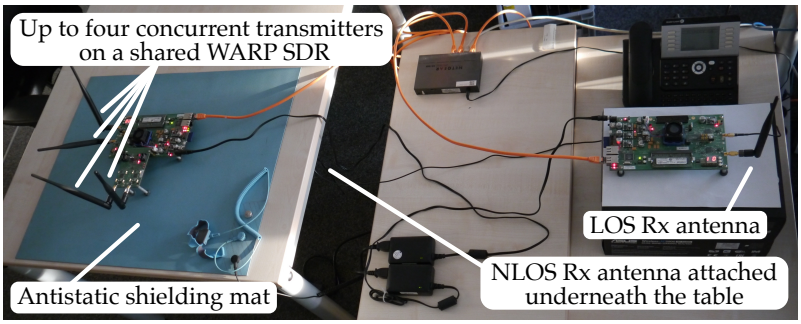


Figure 65: TVZF testbed setup with two WARP SDRs running WARPLab

The generated IQ samples are transferred to the WARP SDR via Ethernet by means of WARPLab [250]. To generate strong interference and to avoid the capture effect, each Tx antenna sends one CCT signal component with the same power. Note that this method differs from the testbed experiments presented in Section 3.3.3 and Section 3.4.3 in which the CCT signal components are mixed and transmitted on a single antenna. Using separate Tx antennas ensures that the different CCT signal components propagate through different physical channels so that the channel components $\mathbf{h}_s$ may differ.

Figure 65 shows the experimental setup with two WARP SDRs in an office environment. The Tx WARP SDR has a distance of about 1.5 m, 2 m, 2.5 m and 4 m to the office walls. The Rx WARP SDR has two antennas, one of which is placed in LOS at a distance of approximately 1.5 m to the Tx antennas, whereas the other Rx antenna is placed in NLOS underneath the table in proximity to the Tx antennas. The table is covered with a conducting antistatic mat to prevent signals from passing directly through the table. The IQ samples of the received CCT frames are transferred via Ethernet by means of WARPLab [250] to a Matlab workspace where they are processed by the TVZF receiver.

To compare the CCT reception performance of the TVZF receiver with the reception performance of commercial devices, three notebooks with an Atheros AR928X wireless network adapter are set up in the same room at a distance of about two to three meters to the Tx antennas in LOS. As in Section 3.3.3 and in Section 3.4.3, the IEEE 802.11 Non-HT encoder writes the CCT parameter settings into the MAC service data unit (MSDU) so that the receivers can extract these values from correctly received frames, allowing the frame reception rate (FRR) to be calculated for each CCT parameter combination.

To avoid collisions with other devices, the experiments are conducted on channel 14, for which a license is required in Germany. In addition, each CCT frame is prepended with a request to send (RTS) frame to postpone pending transmissions of other devices. Each CCT frame is a Non-HT PHY protocol data unit (PPDU) that encapsulates an MSDU with 1500 bytes at MCS 2, i.e., with a PHY data rate of 12 Mbit/s (see Table 2). With this, each CCT frame has 261 OFDM symbols, including the PHY preamble with an equivalent length of 4 OFDM symbols (see Section 2.4.2), resulting in a total duration of 1044 µs.

The TOs Δt_s^{Tx}, the CFOs Δf_s^{Tx} and the phase offsets (POs) $\Delta \phi_s^{\mathrm{Tx}}$ of the senders with indices $s \in \{1, ..., S\}$ are precisely controlled in software to systematically assess the decoding performance of the TVZF receiver. In particular, the first sender with index $s = 1$ is the reference, i.e., $\Delta t_1^{\mathrm{Tx}} = 0\,\mathrm{s}$, $\Delta f_1^{\mathrm{Tx}} = 0\,\mathrm{Hz}$ and $\Delta \phi_1^{\mathrm{Tx}} = 0$. Further, the TO Δt_2^{Tx} of the second sender is swept from $-1\,\mu\mathrm{s}$ to $1\,\mu\mathrm{s}$ in steps of 25 ns. The CFO Δf_2^{Tx} of the second sender is swept from $-5\,\mathrm{kHz}$ to $5\,\mathrm{kHz}$ in steps of 100 Hz in combination

with the TO sweep. The PO $\Delta\phi_2^{\mathrm{Tx}}$ of the second sender is drawn randomly from a uniform distribution in $]-\pi, \pi]$ for each CCT frame to emulate the effect of uncoupled carrier frequency oscillators. The testbed experiments are conducted with two, three and four senders, i.e., $S \in \{2, 3, 4\}$. In case of more than two senders, the CFOs Δf_3^{Tx} and Δf_4^{Tx} are randomly drawn from a uniform distribution in $[-|\Delta f_2^{\mathrm{Tx}}|, |\Delta f_2^{\mathrm{Tx}}|]$, allowing for CFOs with different signs. The TOs Δt_3^{Tx} and Δt_4^{Tx} are randomly drawn from a uniform distribution in $[0, \Delta t_2^{\mathrm{Tx}}]$. The combined parameter sweep is repeated 100 times to gain statistical significance, making a total of 818 100 CCT frame transmissions.

The TVZF receiver is configured to use $\kappa = 64$ IIC cycles and a 2^{16}-point FFT in Algorithm 1 to ensure that these parameters do not limit the decoding performance. Nevertheless, test runs indicate that the TVZF receiver achieves very similar performance also with much lower values, e.g., with $\kappa = 4$ IIC cycles and with a 2^{12}-point FFT. The recorded datasets are evaluated with and without the additional optimizations described in Section 6.3.2 and in Section 6.3.3 to quantify their effect. Since the additional optimization is time-consuming, the dataset processing is accelerated by using 80 000 core hours of the Lichtenberg compute cluster at Technische Universität Darmstadt.

6.4.1 *Two Concurrent Transmitters*

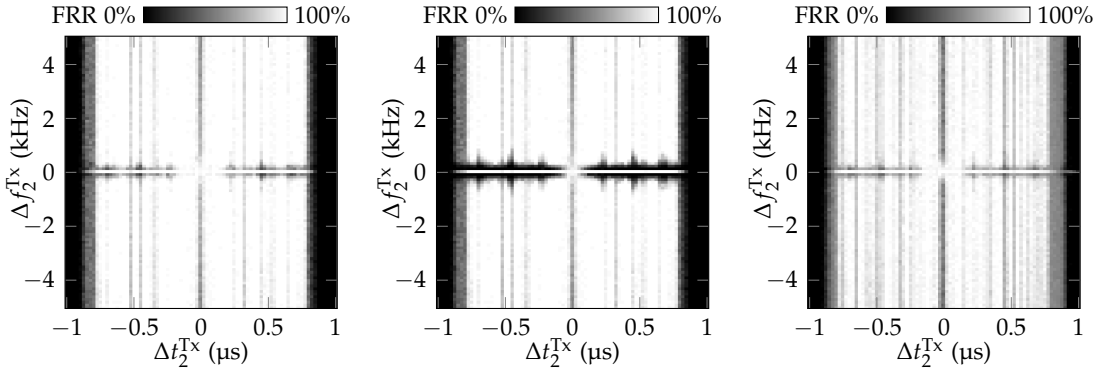
Figure 66 shows various heat maps of the TVZF reception performance for the combined effects of the TO Δt_2^{Tx} and the CFO Δf_2^{Tx} in CCTs with $S = 2$ concurrent transmitters. Irrespective of the evaluation metric, all heat maps show better performance as brighter regions with a linear scale, as in Chapter 3.

The FRR metric considers a frame as received if all 1500 bytes of the MSDU are correct. Figure 66a–c have large regions with a high FRR, which indicates that the TVZF receiver copes with a large range of parameter combinations of the TO Δt_2^{Tx} and the CFO Δf_2^{Tx}. In contrast, Figure 66f shows that the FRR of an Atheros AR928X wireless network adapter is only high for very small absolute values of the CFO Δf_2^{Tx}. The other two Atheros AR928X devices perform very similarly. With a TO Δt_2^{Tx} of up to the GI duration, the average FRR in Figure 66a, b, c and f is 97 %, 93 %, 92 % and 4 %, respectively.

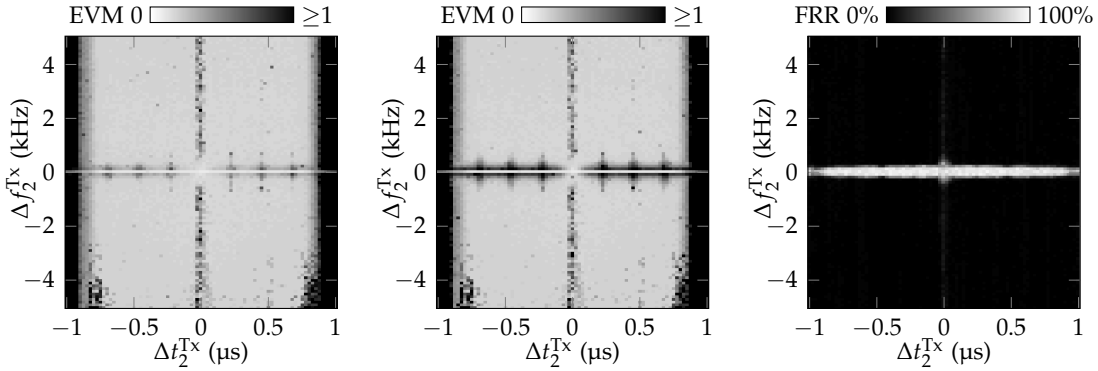
Figure 66a–c also display vertical dark bands starting at $\Delta t_2^{\mathrm{Tx}} \simeq \pm 800\,\mathrm{ns}$, indicating that the TVZF receiver cannot decode CCT frames with a TO exceeding the GI duration. This characteristic is expected due to ISI. Furthermore, since the maps are equally bright for positive and negative TOs Δt_2^{Tx} of up to the duration of the GI, the SIC-based symbol timing estimation described in Section 6.3.1 is effective when any of both concurrent transmitters sends first, i.e., it avoids ISI by finding the earlier CCT signal component.

Figure 66a additionally displays a few vertical gray lines, which can be attributed to the static physical channel characteristics during the experiment, leading to recurring multi-path propagation effects. Figure 66c has even more such artefacts of vertical gray lines, which can be explained by a higher frequency selectivity of the NLOS channel.

Moreover, the most dominant vertical gray line in each FRR plot of the TVZF receiver occurs at a TO of $\Delta t_2^{\mathrm{Tx}} = 0\,\mathrm{s}$. The analysis in Section 3.4.1 shows that a conventional receiver can achieve the highest tolerable CFO without symbol errors at a TO of $\Delta t_2^{\mathrm{Tx}} = 0\,\mathrm{s}$. This is because a TO of $\Delta t_2^{\mathrm{Tx}} = 0\,\mathrm{s}$ does not introduce a frequency-dependent phase shift in the second CCT signal component, so that the phase of its channel coefficients is flat or, in case of a physical channel, at least very similar for several neighboring subcarriers.



(a) [LOS] TVZF w/ optimization (b) [LOS] TVZF w/o optimization (c) [NLOS] TVZF w/ optimization

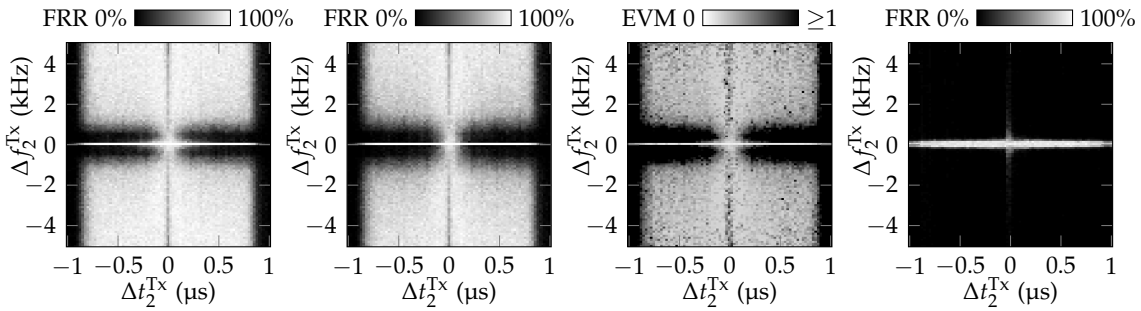


(d) [LOS] TVZF w/ optimization (e) [LOS] TVZF w/o optimization (f) [LOS] Atheros AR928X

Figure 66: CCT testbed results with $S = 2$ concurrent transmitters: each data point covers 100 CCT frames at MCS 2 (see Table 2) with a 1500-byte MSDU. Brighter means better.

However, with a CFO Δf_2^{Tx}, the superposition of the CCT signal components may become subject to a deep fade occurring on several or even all subcarriers at about the same time. Since the PO $\Delta\phi_2^{\mathrm{Tx}}$ is random, such a deep fade occurs in a CCT with a certain probability, depending on the CFO Δf_2^{Tx} and the total number of OFDM symbols Q of the PPDU, as illustrated exemplarily in Figure 20. When a deep fade occurs, it causes several symbol errors within the same OFDM symbol, which cannot be recovered by the TVZF receiver. Therefore, the FRR of the TVZF receiver is moderately degraded at $\Delta t_2^{\mathrm{Tx}} = 0\,\mathrm{s}$.

Figure 66a shows the FRR of the TVZF receiver with the additional optimizations described in Section 6.3.2 and in Section 6.3.3, whereas these optimizations are disabled in the results shown in Figure 66b. A comparison of Figure 66a and b shows that the TVZF receiver achieves very similar FRR characteristics with and without optimization for large parameter ranges, except for small CFOs of up to roughly $\pm 300\,\mathrm{Hz}$. In particular, the FRR is reduced in parameter regions with a TO of $\Delta t_2^{\mathrm{Tx}} = k \cdot 228.6\,\mathrm{ns}$, $k \in \{\pm 1, \pm 2, \pm 3\}$. Note that these TO values cause a phase shift of a multiple of 2π between two neighboring pilot subcarriers, as described in Section 6.3.3. An inspection of CCT frame receptions in these parameter regions with a small CFO Δf_2^{Tx} reveals that the estimation accuracy of the channel coefficients $\hat{h}_s^p$ of the detected senders are degraded also when all CFOs are detected, so that the optimization enhances these channel coefficient estimates. Moreover, the IIC-based detection of CCT signal components according to Algorithm 1 may sometimes miss one of the two senders when their carrier frequencies are too close.



(a) [LOS] TVZF w/ opt. (b) [LOS] TVZF w/o opt. (c) [LOS] TVZF w/ opt. (d) [LOS] AR928X

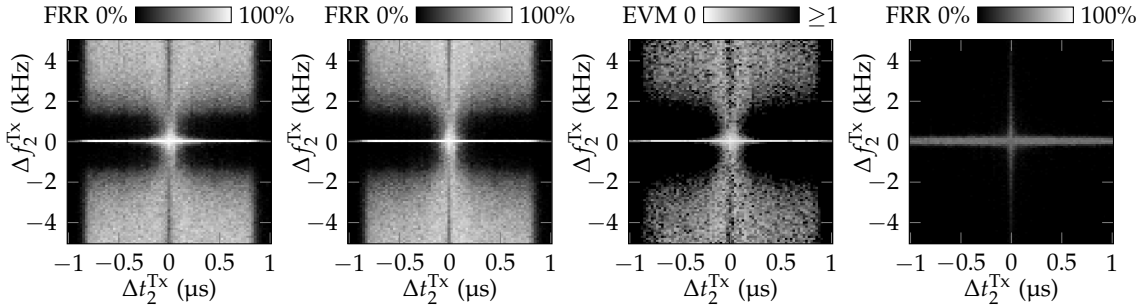
Figure 67: CCT testbed results with $S = 3$ concurrent transmitters: the TO Δt_3^{Tx} and the CFO Δf_3^{Tx} are drawn randomly such that $\Delta t_3^{\mathrm{Tx}} \in [0, \Delta t_2^{\mathrm{Tx}}] \wedge \Delta f_3^{\mathrm{Tx}} \in [-|\Delta f_2^{\mathrm{Tx}}|, |\Delta f_2^{\mathrm{Tx}}|]$.

In this case, the undetected sender causes interference and degrades the TVZF reception performance. Note that the Atheros AR928X tolerates a slightly larger range of CFOs around $\Delta f_2^{\mathrm{Tx}} \simeq 0\,\mathrm{Hz}$ across a wide range of TOs, which might be due to algorithms that deal with the Doppler effect in mobile scenarios. An integration of such algorithms into the TVZF receiver may possibly improve the FRR also for small CFOs.

Figure 66d and Figure 66e show the average EVM of the TVZF receiver with and without additional optimization, respectively. Since the EVM is computed by means of the demapped symbols, it is independent of the employed modulation scheme and the forward error correction (FEC). These EVM maps have a homogeneous shade for wide parameter ranges that corresponds to an SINR of roughly 17 dB. The average EVMs are very similar for the cases with and without optimization, except for the narrow range of small CFOs of up to roughly $\pm 300\,\mathrm{Hz}$. Further, there is no visible EVM degradation towards higher CFOs, which indicates that ICI indeed has a minor effect in the investigated parameter range. There are also some dark dots in the EVM maps that do not show up in the FRR maps. Inspections reveal that these dots are caused by a few corrupted frames that lead to symbol errors with large EVM values.

6.4.2 *Three Concurrent Transmitters*

Figure 67 shows the performance of the TVZF receiver for CCTs with $S = 3$ concurrent transmitters. In particular, the maps in Figure 67a and b show the FRR for both cases with and without the additional optimizations described in Section 6.3.2 and in Section 6.3.3. The main difference to CCTs with $S = 2$ concurrent transmitters is that the FRR is low for a broader range of small absolute values of the CFO Δf_2^{Tx}. Figure 67c shows that the EVM is subject to degradations in the same parameter regions. Since the CFO Δf_3^{Tx} is randomly drawn from a uniform distribution in $[-|\Delta f_2^{\mathrm{Tx}}|, |\Delta f_2^{\mathrm{Tx}}|]$, the TVZF receiver has to detect three CFOs that are close to each other when Δf_2^{Tx} takes a low value. If the TVZF receiver misses the detection of one of the CFOs, the corresponding transmitter causes interference and degrades the SINR. A comparison of Figure 67a and b shows that the additional optimizations enlarge the region with high FRR values slightly at the center of the map. Figure 67d shows that an Atheros AR928X wireless network adapter has similar FRR characteristics as for $S = 2$ in Figure 66f. For TOs Δt_2^{Tx} and Δt_3^{Tx} of up to the GI duration, the average FRR in Figure 67a and b is 78 % and 72 %, respectively.



(a) [LOS] TVZF w/ opt. (b) [LOS] TVZF w/o opt. (c) [LOS] TVZF w/ opt. (d) [LOS] AR928X

Figure 68: CCT testbed results with $S = 4$ concurrent transmitters: the TO Δt_s^{Tx} and the CFO Δf_s^{Tx} are drawn randomly such that $\Delta t_s^{\mathrm{Tx}} \in [0, \Delta t_2^{\mathrm{Tx}}] \wedge \Delta f_s^{\mathrm{Tx}} \in [-|\Delta f_2^{\mathrm{Tx}}|, |\Delta f_2^{\mathrm{Tx}}|] \ \forall s \in \{3, 4\}$.

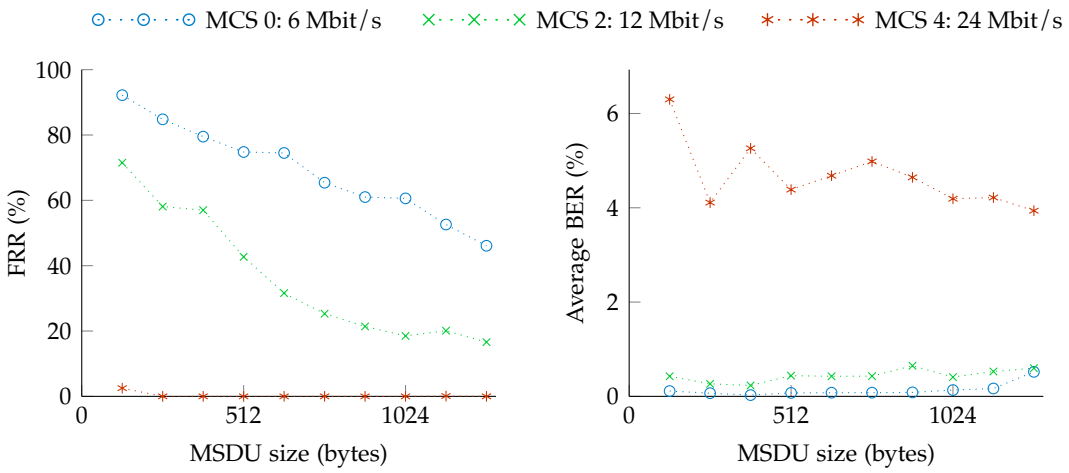
6.4.3 *Four Concurrent Transmitters*

Figure 68 shows the performance of the TVZF receiver for CCTs with $S = 4$ concurrent transmitters. The FRR shown in Figure 68a–b and the EVM shown in Figure 68c are degraded further in comparison to the experiment with $S = 3$ concurrent transmitters. Further, Figure 68d shows the FRR of an Atheros AR928X wireless network adapter. For TOs Δt_2^{Tx}, Δt_3^{Tx} and Δt_4^{Tx} of up to the GI duration, the average FRR in Figure 68a and b is 49 % and 46 %, respectively. Nevertheless, these numbers include the degradations at small CFOs, while the FRR of the TVZF receiver is higher at $|\Delta f_2^{\mathrm{Tx}}| > \sim 2\,\mathrm{kHz}$.

6.5 TESTBED EVALUATION: CCTS FROM COMMERCIAL DEVICES

This section presents a testbed experiment that assesses the performance of the TVZF receiver for IEEE 802.11 Non-HT with CCTs from three unsyntonized commercial devices. This particularly allows quantification of TVZF performance metrics for CCT parameter combinations that occur in a practical implementation. The employed devices are Asus RT-AC86U wireless routers with Broadcom BCM4365E/BCM4366E chips for IEEE 802.11. The Asus RT-AC86U devices are configured through a modified firmware to respond to a specific trigger frame by retransmitting the received frame with a fixed scrambler seed. Technically, the retransmission (RTx) is triggered through the acknowledgment (ACK) engine of the devices so that a received trigger frame is retransmitted immediately after a short interframe space (SIFS), i.e., 16 µs after the frame reception (see Section 2.4.8). The frequency synchronization features presented in Chapter 5 are *not* used.

Figure 69 shows the experimental setup with three Asus RT-AC86U devices placed next to each other at a distance of 1.5 m in LOS to a WARP SDR with WARPLab [250]. To generate a CCT with the three Asus RT-AC86U devices, the WARP SDR sends a trigger frame so that the three devices concurrently respond with an RTx of the frame. The WARP SDR simultaneously captures the triggered CCT and feeds the received IQ samples to a Matlab workspace so that the CCT can be processed by the TVZF receiver. The experiment is conducted with MCS 0, MCS 2 and MCS 4 (see Table 2) and with MSDU sizes from 128 to 1280 bytes with a step size of 128 bytes, respectively. For each parameter combination of the MCS and the MSDU size, 1000 CCTs are triggered to facilitate the calculation of the FRR and the average bit error rate (BER).



(a) FRR (b) BER

Figure 69: TVZF testbed setup with three unsyntonized Asus RT-AC86U wireless routers with a modified firmware: The Asus RT-AC86U devices generate a CCT with different CFOs by retransmitting a trigger frame from the WARP SDR by means of the ACK engine.

During the experiment, the three Asus RT-AC86U devices exhibit CFOs of roughly 5 kHz, −10 kHz, and −15 kHz with respect to the WARP SDR, with variations of ∼1 kHz. Further, the average TO of the CCTs from the three Asus RT-AC86U devices is 41.2 ns. This TO estimate is calculated as the difference of the measured CDS for CCTs with three Asus RT-AC86U devices and the measured CDS for a single Asus RT-AC86U transmitter, which are 112.4 ns and 71.2 ns, respectively. Due to this precise timing and due to the equal signal power levels of the concurrent transmitters, the received signal is effectively subject to a beating effect, periodically causing a deep fade over several or all subcarriers, as discussed in Section 6.4.1. This effect reduces the FRR for an increasing frame length, whereas the BER stays at about the same level, as shown in Figure 69. Nevertheless, the FRR could be increased for larger frames by artificially introducing small TOs between the three concurrent transmitters, as indicated by the experimental results in Figure 67. Additionally, more diverse signal power levels may also help to increase the FRR further, as shown subsequently in Section 6.6.

6.6 Impact of Relative Signal Power Levels

The testbed experiments in Section 6.4 and in Section 6.5 are conducted under the assumption that a CCT with an equal power level of the CCT signal components leads to strong interference, so that a receiver cannot exploit the capture effect (see Section 2.1.2). Hence, to showcase the enhanced reception performance of the TVZF receiver in relation to commodity devices, the testbed experiments are conducted with the same Tx power for each concurrent transmitter. However, an equal power level of all CCT signal components may not always be realistic in practice since different transmitters may have different distances from a receiver. Further, also the reception performance of the TVZF receiver may vary as a function of the relative power levels of different CCT signal components. Hence, this section quantifies the effect of the relative signal power levels of concurrent transmitters on the reception performance of the TVZF receiver through simulations. In addition, simulations are also conducted with a conventional receiver to identify the ranges of relative signal power levels for which the capture effect can be exploited. A comparison of these results with and without TVZF equalization also allows the identification of the ranges of relative signal power levels for which TVZF equalization enhances the reception performance beyond the capture effect.

6.6.1 *FRR Characteristics*

The CCT simulations presented in this section are conducted with combined parameter sweeps of the signal power levels of $S = 2$ concurrent transmitters. The TO Δt_2^{Tx} between both concurrent transmitters is fixed at 300 ns and the CFO Δf_2^{Tx} is fixed at 3 kHz. Figure 66a shows that the TVZF receiver achieves a good reception performance with this parameter combination at equal signal power levels, so that the TVZF equalization should take effect in the CCT simulations. The signal power level of each concurrent transmitter is adjusted through a gain factor that is swept from 0 to 1 in steps of 0.01. Each CCT signal component is processed through an AWGN channel that generates an SNR of 30 dB when the gain factor is 1, while the noise power is kept at a constant level when the gain factor is swept. Both noisy CCT signal components are added to emulate the effect of a superposition of a CCT. The resulting superposition of both signals is fed to the custom Matlab receiver that is also employed for the simulations in Chapter 3. The receiver processes this signal with and without TVZF equalization, respectively. When TVZF equalization is disabled, the receiver applies conventional symbol timing estimation, conventional pilot tracking and zero-forcing equalization (see Section 2.4). Any differences in the reception performance can be attributed to the TVZF equalization since the receiver is otherwise identical in both modes of operation.

Since the AWGN channels are flat-fading, the TVZF receiver obtains perfect channel estimates through interpolation and extrapolation of the pilot subcarrier support points if the latter are correctly estimated. Thus, the TVZF reception performance is not degraded by the interpolation-based channel estimation (see Section 6.3.3) and mainly depends on the SIC-based symbol timing estimation (see Section 6.3.1), the IIC-based detection and parameter estimation of CCT signal components (see Section 6.3.2) and the impact of events with destructive interference during a CCT on the remaining decoding steps, such as symbol demapping and convolutional decoding.

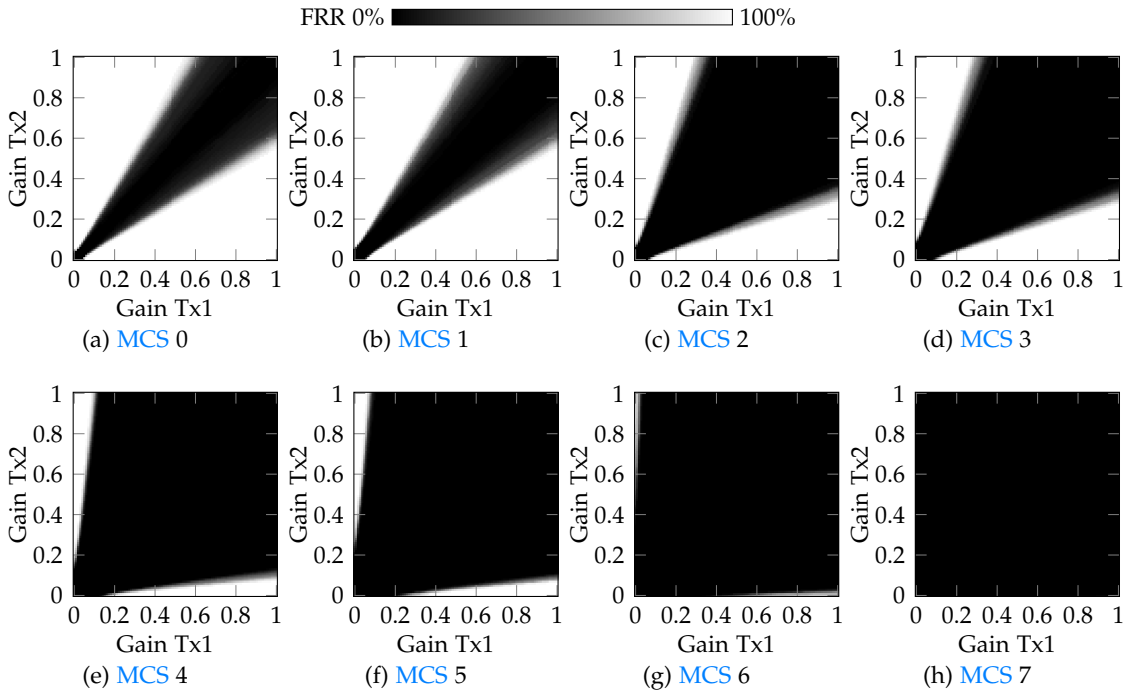


Figure 70: TVZF disabled: FRR of CCTs with AWGN channels with up to 30 dB SNR

Figure 70 shows the FRR characteristics of the receiver for all MCS indices (see Table 2) when TVZF equalization is disabled. When both gain factors are zero, the FRR is also zero as the receiver gets only noise. When the gain factor of only one transmitter increases, the SNR increases and the signal can be decoded. However, the capture effect allows also the signal power level of a secondary transmitter to rise to a certain level while the FRR stays at a high level. Still, an even increasing signal power level of a secondary transmitter leads to decoding errors due to interference, visible as dark regions in Figure 70.

The minimum ratio of signal power levels for which the capture effect occurs is also referred to as the capture threshold (see Section 2.1.2). Figure 70 indicates that the capture threshold increases with a higher MCS (see Table 2), i.e., the dark regions get wider. More precisely, the capture threshold depends on the modulation scheme rather than the coding rate. For instance, Figure 70a and Figure 70b have quite similar FRR characteristics while the respective MCS settings only differ in terms of their coding rates. In contrast, Figure 70c indicates a higher capture threshold than Figure 70a while the respective MCS settings only differ in terms of their modulation schemes. The reason is that the constellation points are denser and that the distance between decision boundaries is smaller for higher modulation schemes. When the EVM is large enough, symbol errors can occur since received data symbols may be displaced across decision boundaries. Due to the TO Δt_2^{Tx} in the simulations, the phases of the subcarriers of the secondary transmitter are fanned out into all angular directions (see Section 3.4.2), so that symbol errors occur on a certain fraction of data subcarriers of each OFDM symbol when the interference is strong enough. In this case, FEC only takes effect when the symbol error rate (SER) is low, which only holds for a narrow range of relative signal power levels of both transmitters. When the gain factor of the secondary transmitter increases further, the SER also increases, so that FEC cannot recover the CCT frames at a certain point.

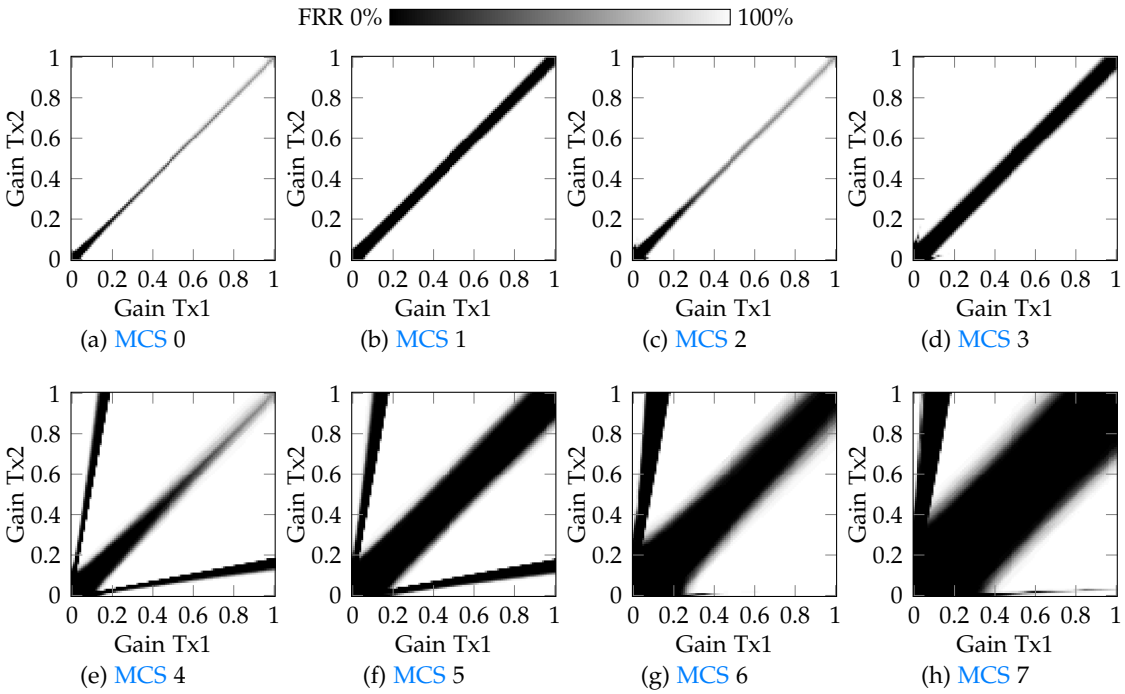
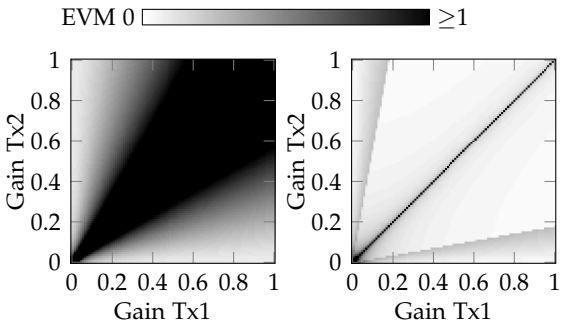


Figure 71: TVZF enabled: FRR of CCTs with AWGN channels with up to 30 dB SNR

Figure 71 shows the FRR characteristics of the receiver for all MCS indices (see Table 2) when TVZF equalization is enabled. The white regions in Figure 71 are apparently much wider than in the corresponding plots in Figure 70, confirming that TVZF equalization recovers CCT frames in parameter ranges in which the capture effect cannot be exploited. Particularly Figure 71a–d each show large white regions with an FRR of 100 %, except for a narrow diagonal region in which both concurrent transmitters have roughly the same power. This indicates that equal signal power levels of concurrent transmitters pose the most challenging conditions also for the TVZF receiver. The reason is that deep fades occur under these conditions on certain subcarriers within each OFDM symbol, which are caused by destructive interference during phase inversions of the CCT signal components. Still, when the signal power levels are slightly different, the superposition is not completely canceled when the phases of the CCT signal components become inverted, so that the TVZF receiver can decode the CCT frames. Furthermore, the dark regions in Figure 71a and in Figure 71c are thinner than in Figure 71b and in Figure 71d, which indicates that FEC mitigates the impact of symbol errors in these parameter ranges.

In Figure 71e–h, there are additional dark regions that are not visible in the plots with a lower MCS and that occur at parameter combinations with unequal signal power levels. Figure 71e, for instance, shows that the FRR drops from 100 % to 0 % when the gain of a primary transmitter is 1 and when the gain of a secondary transmitter falls below 0.18. Further, when the gain of a secondary transmitter falls below 0.11, the FRR rises to 100 %. These effects occur since the TVZF receiver works with detection thresholds, so that a weak CCT signal component may not be detected and cause detrimental interference. In the simulations, both the SIC-based symbol timing estimation (see Section 6.3.1) and the IIC-based CFO detection (see Section 6.3.2) employ a detection threshold of 0.18 with respect to the strongest CCT signal component. Since the constellation points



(a) TVZF disabled (b) TVZF enabled

Figure 72: EVM of CCTs with AWGN channels with up to 30 dB SNR

are denser at higher modulation schemes, the interference of an undetected secondary transmitter can be strong enough to cause symbol errors, i.e., to displace symbols to wrong constellation points. Further, the dark regions are not symmetric in Figure 71g–h since the TVZF receiver adapts the detection threshold of IIC dynamically as a function of the frame length, so that it detects the weak CCT signal component at higher MCS indices. This finding shows that the FRR characteristics of the TVZF receiver could be further optimized through fine-tuning of dynamic detection thresholds for both the SIC-based symbol timing estimation and the IIC-based detection of CCT signal components.

6.6.2 *EVM Characteristics*

Figure 72 provides a comparison of the EVM performance of the receiver when the TVZF equalization is disabled and enabled. Note that the EVM reflects the magnitude of the displacement of received data symbols in the IQ plane and that it is therefore independent of the employed MCS. Figure 72a shows that the EVM of a conventional receiver gradually increases with an increasing signal power of a secondary transmitter, reflecting its susceptibility to interference. In contrast, Figure 72b shows that the EVM performance follows more complex patterns when the TVZF equalization is enabled. When there is no interference by a secondary transmitter, the EVM gradually decreases with an increasing signal power of the primary transmitter, since the SNR increases. When interference of a secondary transmitter sets in from a signal power level of zero, the EVM gradually increases until the signal power level of the secondary transmitter reaches the detection threshold of the TVZF receiver. The EVM values in these border regions of Figure 72b correspond to the values of the conventional receiver in Figure 72a. When the signal power level of the secondary transmitter increases further, the TVZF receiver detects this CCT signal component, causing the EVM to jump to a lower value. When both CCT signal components are within the detection range of the TVZF receiver, the EVM only takes large values when both signal power levels are roughly equal. In this case, the EVM metric is dominated by the impact of destructive interference occurring on a subset of subcarriers within each OFDM symbol, i.e., the superposition of both signals is completely canceled on these subcarriers. Otherwise, a comparison of Figure 72a and Figure 72b clearly shows that TVZF equalization improves the reception performance for wide ranges of relative signal power levels of concurrent transmitters.

6.7 CONCLUSION

TVZF equalization enables a receiver to cope with non-stationary interference in a CCT when accurate frequency synchronization between concurrent transmitters fails or cannot be achieved. TVZF is designed to enable CCT-based communication schemes, such as the network flooding scenario presented in Section 1.2.1 and the extensive broadcasting scenario presented in Section 1.2.2. Particularly the latter scenario benefits from TVZF equalization when nodes that are distributed in an extended area cannot synchronize in frequency through a common frame reception.

The TVZF system model generalizes the system model of the CCT analysis in Chapter 3 for an arbitrary number of concurrent transmitters S. In particular, the TVZF system model describes a time-variant vector of channel coefficients $\mathbf{h}_\Sigma(q)$ as a function of the channel coefficients $\mathbf{h}_s$ of the concurrent transmitters with indices $s \in \{1, ..., S\}$ and their respective CFOs Δf_s^{Rx}, while the effects of the TOs Δt_s^{Rx} are implicitly covered by $\mathbf{h}_s$. Thus, to recover a received CCT frame, a TVZF receiver has to estimate these parameters to reconstruct the time-variant vector of channel coefficients $\hat{\mathbf{h}}_\Sigma(q)$ for TVZF equalization. Still, the procedure for parameter estimation is not prescribed by the TVZF system model, so custom solutions must be designed for specific PHYs.

This work presents a TVZF receiver design for the IEEE 802.11 Non-HT PHY that estimates all parameters for TVZF equalization by means of the received superposition of the signal components of a CCT. To this end, the TVZF receiver applies a SIC-based symbol timing estimation that ensures that all CCT signal components are covered by the GI of the OFDM PHY in order to mitigate ISI. Further, the TVZF receiver detects the CCT signal components with indices $s \in \{1, ..., \hat{S}\}$ by processing the pilot symbol traces with IIC. In this process, it estimates the channel coefficients $\hat{\mathbf{h}}_s^P$ of the four pilot subcarriers with the indices $P = \{-21, -7, 7, 21\}$ and the CFO $\Delta \hat{f}_s^{\mathrm{Rx}}$ for each $s \in \{1, ..., \hat{S}\}$. The TVZF receiver further estimates the complex-valued channel coefficients $\hat{\mathbf{h}}_s^\zeta$ for each $s \in \{1, ..., \hat{S}\}$ by interpolating the respective pilot subcarrier support points $\hat{\mathbf{h}}_s^P$ under the constraint of the superposition $\tilde{\mathbf{h}}_\Sigma^{\mathrm{LTF}}$ from the L-LTF. Optionally, the IIC-based parameter estimates $\Delta \hat{f}_s^{\mathrm{Rx}}$ and $\hat{\mathbf{h}}_s^P$ and the interpolation-based channel estimates $\hat{\mathbf{h}}_s^\zeta$ can be further optimized by means of the downhill simplex algorithm [122]. The TVZF receiver combines the parameter estimates of the $\hat{S}$ CCT signal components to compute a time-variant vector of channel coefficients $\hat{\mathbf{h}}_\Sigma(q)$ and finally equalizes the received CCT frame in the IEEE 802.11 Non-HT PPDU format.

The TVZF receiver is evaluated in testbed experiments with WARP SDRs [246] with combined parameter sweeps of the TO Δt_2^{Tx} and the CFO Δf_2^{Tx}. In case of more than two senders, the CFOs and the TOs are randomly drawn from uniform distributions such that $\Delta f_s^{\mathrm{Tx}} \in [-|\Delta f_2^{\mathrm{Tx}}|, |\Delta f_2^{\mathrm{Tx}}|] \wedge \Delta t_s^{\mathrm{Tx}} \in [0, \Delta t_2^{\mathrm{Tx}}] \; \forall s \in \{3, 4\}$. With an absolute TO of up to the duration of the GI and with a CFO within an interval of $[-5, 5]$ kHz, the average FRR is 97 %, 78 % and 49 % for $S = 2$, $S = 3$ and $S = 4$ concurrent transmitters, respectively. Moreover, the reception performance of the TVZF receiver is assessed with CCTs from three unsyntonized Asus RT-AC86U wireless routers that exhibit CFOs of about 5 kHz, -10 kHz and -15 kHz with respect to the WARP SDR. The Asus RT-AC86U devices are configured to generate CCTs through the ACK engine by means of a modified firmware. The TVZF receiver achieves an FRR of up to 92 % for $S = 3$ senders in this experiment.

While TVZF equalization enhances the reception performance of unsyntonized CCTs in comparison to conventional equalization and commodity devices, there are a few conditions under which the reception performance of the TVZF receiver is degraded. First, when the concurrent transmitters are accurately synchronized in time, i.e., with TOs of $\Delta t_s^{\mathrm{Tx}} \simeq 0\,\mathrm{s}$, a beating effect occurs, so that the superposition of a CCT fades simultaneously on several or all subcarriers, which may cause frame loss in case of a deep fade. Still, this effect could be mitigated through dithering, i.e., by randomizing the TOs Δt_s^{Tx} slightly. Second, if multiple concurrent transmitters have similar carrier frequencies, the TVZF receiver may detect only one CCT signal component and miss the others. In this case, the IIC-based estimation of the channel coefficients of the pilot subcarriers and the interpolation-based channel estimation may be inaccurate. Still, additional optimization enhances the reception performance in this parameter range. Third, when the power levels of the CCT signal components are roughly equal, phase inversions of the CCT signal components on certain subcarriers cause deep fades due to destructive interference, so that an increased number of symbol errors may occur. Still, the impact of this effect can be mitigated by using an MCS with a low coding rate. Fourth, the reception performance is degraded with an increasing number of concurrent transmitters. This effect is compounded by the design of the testbed experiments in which the CFOs Δf_3^{Tx} and Δf_4^{Tx} are randomly drawn from a uniform distribution in $[-|\Delta f_2^{\mathrm{Tx}}|, |\Delta f_2^{\mathrm{Tx}}|]$, enforcing similar carrier frequencies for low absolute values of Δf_2^{Tx}. Still, experiments with unsyntonized Asus RT-AC86U wireless routers show that the TVZF receiver can recover CCTs from three devices with an FRR of up to 92 %.

The TVZF receiver design for the IEEE 802.11 Non-HT PHY offers various advantages over SourceSync [186] (see Section 2.1.7.1), which supports a subset of the CCT-based communication schemes enabled by the TVZF receiver. In particular, SourceSync applies interleaving and periodic signaling with a custom OFDM PHY to separate training symbols from different transmitters so that the respective estimates can be obtained without interference, which creates additional overhead. In contrast, the TVZF receiver estimates the number of concurrent transmitters and their respective vectors of channel coefficients and CFOs only by means of the received superposition of the CCT signal components. With this, the TVZF receiver allows for a more lightweight operation in which nodes can opportunistically join a CCT without generating overhead for signaling. Moreover, while SourceSync deviates from the IEEE 802.11 standard in several ways to achieve synchronization in time and frequency, the TVZF receiver works with CCTs that comply with the IEEE 802.11 Non-HT PPDU format.

The TVZF receiver can optionally perform optimizations of the IIC-based parameter estimates and the interpolation-based channel coefficients through the downhill simplex algorithm [122], so that CCT parameter regions can be identified in which the TVZF receiver design can be further enhanced. In this process, the IIC-based parameter estimates are jointly optimized by means of $\hat{S} \cdot 9$ variables. Further, the interpolation-based channel coefficients are jointly optimized by means of $\hat{S} \cdot 5$ variables. Therefore, the processing of a CCT frame with optimization can take up to five seconds on a Core i5 desktop central processing unit (CPU). To speed up the processing of the datasets with combined parameter sweeps of the TO Δt_2^{Tx} and the CFO Δf_2^{Tx}, each of which contains 818 100 CCT frames, 80 000 core hours of the Lichtenberg high-performance computer of the Technische Universität Darmstadt are used for tests and evaluations.

When the additional optimization steps are disabled, the TVZF receiver runs much faster, so that data transfers become the main bottleneck. Still, the IIC-based detection of CCT signal components requires a certain amount of processing time due to multiple iterations of interference cancellation and due to computations of multiple FFT spectra. Further, the interpolation-based estimation of channel coefficients uses an exhaustive search algorithm to find the best combination of phase paths for the detected CCT signal components. While the TVZF receiver implementation in Matlab is primarily designed to assess the feasibility of the devised algorithms in testbed experiments, optimized designs with parallelized processing in hardware might work also in real-time. A modified TVZF receiver that executes the interpolation-based channel estimation by means of deep neural networks (DNNs) in two cascades shows a slightly degraded FRR in comparison to the receiver design presented in this thesis while speeding up the channel estimation by five orders of magnitude to 31.9 µs [195, 196]. Similarly, other processing steps of the TVZF receiver, such as the SIC-based symbol timing estimation and the IIC-based parameter estimation, may potentially also be carried out by means of DNNs to facilitate parallelized processing in hardware for real-time implementations. However, the design of proper system models, the generation of datasets that reflect the many-faceted characteristics of physical environments and the systematic development of DNN-based system components is a challenging endeavor [196].

Part IV

DISCUSSION AND CONCLUSION

Chapter 7 discusses the findings of this thesis and highlights the strengths and the limitations of the presented solutions. In addition, it gives an outlook on further advancements through an integration of CCT-based network flooding and extensive broadcasting with other mechanisms to widen the spectrum of dURLLC applications. Chapter 8 presents concluding remarks.

7 DISCUSSION

This thesis explores the technical feasibility of concurrent cooperative transmission (CCT) with orthogonal frequency-division multiplexing (OFDM) and presents practical solutions that comply with the OFDM-based IEEE 802.11 Non-HT and HT physical layers (PHYs). This chapter discusses the characteristics of these solutions as well as their mechanisms, design alternatives and further technological advancements to enable the envisioned decentralized URLLC (dURLLC) paradigm by means of CCT-based communications. Section 7.1 discusses the performance characteristics of CCT as well as the performance achievements of the developed prototypes. Section 7.2 discusses the core technologies and the key aspects of CCT and highlights their characteristics and performance implications. Section 7.3 discusses technology extensions that add enhanced functionalities to dURLLC applications and that optimize the performance of CCT-based communications.

7.1 Performance Characteristics and Achievements

The overarching goal of this thesis is to enable CCT for data dissemination in extensive areas with an unprecedented performance in terms of latency, reliability and throughput. Glossy [73] is the state-of-the-art technology for CCT-based network flooding and employs the IEEE 802.15.4 PHY, which is based on the direct-sequence spread spectrum (DSSS) modulation technique [107]. However, it has a PHY data rate of only 250 kbit/s [73, 107]. This thesis shows that CCT is feasible with broadband communications through OFDM, allowing for much higher data rates than with IEEE 802.15.4, so that a given amount of data can be transmitted in a shorter time, i.e., the latency per hop is reduced significantly. The prototypes developed in this work particularly enable CCT-based network flooding with coherent interference (CI) with the OFDM-based IEEE 802.11 Non-HT and HT PHYs in real-time and are assessed in various testbed experiments. The prototypes are based on the Wireless Open Access Research Platform (WARP) v3 [246] and on the Asus RT-AC86U wireless router, as presented in Chapter 4 and in Chapter 5, respectively.

This section discusses the performance characteristics of CCT-based network flooding in general and the performance achievements of the prototypes of this work in particular. Section 7.1.1 discusses the signal-to-interference-plus-noise ratio (SINR) characteristics of CCT with CI, which also have certain implications on other performance metrics. Section 7.1.2 discusses the characteristics of the communication range of CCT with CI. Section 7.1.3 discusses the occupancy of the medium by a CCT-based flooding process. Section 7.1.4 discusses the performance tradeoffs between the latency, the reliability and the throughput in CCT-based network flooding. Section 7.1.5 compares the performance of the prototypes of this work with Glossy [73]. Section 7.1.6 discusses the achievability of the requirements of Ultra-Reliable Low-Latency Communications (URLLC) through CCT-based network flooding with the prototypes of this work.

7.1.1 *SINR Characteristics of CCT with CI*

The CCT signal quality is assessed in testbed experiments with ping-pong transmissions between two groups of nodes in which the number of nodes per group is swept from one to nine and vice versa, as presented in Section 4.5.4 and in Section 5.4.3, respectively. Figure 46 and Figure 58 show that the SINR values of the CCTs typically have a much wider range than the transmissions from a single node and that the SINR characteristics are rather independent of the number of nodes in a group. While most of the SINR values of the CCTs are slightly higher or lower than the values of the transmissions from a single node, there are also some outliers with a very low SINR, indicating that destructive interference may occur over a number of OFDM subcarriers in a few CCTs.

In case of a CCT, multiple components of a signal interfere with each other at a receiver, so that the quality of the superimposed signal depends on the transmission powers and the path losses of the respective CCT signal components and on the time offsets (TOs), the phase offsets (POs) and the carrier frequency offsets (CFOs) between the senders as well as the elapsed time since the start of the frame, as analyzed for the IEEE 802.11 DSSS PHY in Section 3.3 and for the OFDM-based IEEE 802.11 Non-HT PHY in Section 3.4. The prototypes presented in Chapter 4 and Chapter 5 generate CCTs with CI, i.e., they make the mutual interference as invariant as possible over the duration of a frame, so that the impact of the parameter *time* is minimized. Furthermore, they accurately synchronize in time with respect to each other upon each common frame reception. When the TO Δt is very small, all OFDM subcarriers fade similarly in a CCT since only the second summand of the exponent in Equation 10 depends on the subcarrier index n. This relation may also hold in practice when the physical channel is rather flat over its entire bandwidth, which may particularly apply under line-of-sight (LOS) conditions. Furthermore, the PO $\Delta\theta$ can be assumed to be random in practice since the frequency oscillators of the distributed nodes are uncoupled. Thus, with time synchronization, i.e., with $\Delta t \sim 0\,\mathrm{s}$, Equation 10 shows that the interference of a CCT may be constructive or destructive by chance, due to the randomness of the PO $\Delta\theta$. However, the actual time synchronization accuracy of the prototypes presented in Chapter 4 and Chapter 5 may vary a little across different CCTs, as indicated by the characteristics of their measured interframe space (IFS) values shown in Figure 43b and in Figure 55b, respectively. Therefore, destructive interference over all subcarriers occurs only for a certain fraction of the CCTs generated by the prototype nodes.

In summary, the CCT signal quality in terms of the SINR takes a wider range of values than with transmissions from a single node, as shown in Figure 46 and in Figure 58. Due to the randomness of the POs of the concurrent transmitters in each CCT, a receiver typically gets different SINR values for successive CCTs and may by chance even get a very low SINR, which in turn may prohibit the correct decoding of the respective CCT. Furthermore, different receivers of the same CCT may typically get different SINR values since the CCT signal components superimpose differently in different locations. Still, most CCTs are received with high SINR values ranging at a level that is a little higher or a little lower than the SINR values of the transmissions from a single node. Due to the employment of multiple nodes and due to repetitive reception opportunities in CCT-based communication schemes, the impact of occasionally occurring low SINR values can usually be overcome, so that messages can be received reliably.

7.1.2 *Communication Range of CCT with CI*

The communication range of a single node generally depends on its transmission power, the path loss, the noise at a receiver, the used modulation scheme and the coding rate of the employed forward error correction (FEC). While the first three parameters affect the signal-to-noise ratio (SNR) at a receiver, the latter two parameters set a lower limit on the SNR required by a receiver to decode data below a certain bit error rate (BER).

Some related works on CCT-based relaying discussed in Section 2.1.7 highlight that a CCT may have an extended communication range in comparison to a transmission from a single node due to the accumulation of the signal power [68, 94, 114, 133]. While a CCT may also have a reduced gain due to destructive interference, the median gain is considered to increase with an increasing number of concurrent transmitters [68].

The results of this work, in contrast, indicate that the communication range of a CCT with CI and with an OFDM PHY may be but is not necessarily extended in comparison to a transmission from a single node, according to the aforementioned SINR characteristics. Further, the signal components of a CCT may interfere differently in different locations due to different propagation delays and channel conditions, so that neighboring nodes get different SINR values, i.e., they may or may not receive the CCT successfully by chance. Still, Figure 46 and Figure 58 show that most CCTs have a high SINR in a narrow range, which is quite independent of the number of transmitters. Therefore, in a dense network, most of the nodes at a certain distance from the previous hop can be expected to decode a flooding frame successfully and to forward it, while those active nodes may differ slightly in successive flooding processes. Consequently, the communication range of a CCT-based flooding process should be quite consistent across successive hops.

7.1.3 *Medium Occupancy of a CCT-Based Flooding Process*

In a CCT-based flooding process, a flooding frame is retransmitted by successive groups of nodes that opportunistically form virtual antenna arrays upon their respective receptions of the flooding frame, as described in Section 1.2.1. Sometimes a node may by chance get a very low SINR for a received CCT, according to the SINR characteristics discussed in Section 7.1.1, and fail to recover the flooding frame correctly. In this case, the node may get another chance to receive the flooding frame upon the next retransmission (RTx). If the considered node receives the flooding frame, it contributes to a CCT that is mainly generated by nodes that are further away than its direct neighbors. With this, the node occupies the medium locally for another transmit (Tx) slot after its direct neighbors.

Generally, a node may fail to receive a flooding frame a few times in succession if the conditions are bad and if there are other nodes in its neighborhood that continue to retransmit the flooding frame. The logical hop distance of a node from the initiator of a flooding process can thus be shifted by the respective number of failed receptions until the node receives the flooding frame correctly, i.e., the medium can be additionally occupied for a corresponding number of Tx slots. These Tx slots may occur in addition to the additional RTx repetitions of the low-level medium access control (MAC) protocol presented in Section 4.2.1.3. The additional occupancy of the medium must be considered to avoid collisions in flooding process pipelining, as further discussed in Section 7.1.4.3.

7.1.4 *Latency, Reliability and Throughput*

CCT allows data to be efficiently disseminated in an extensive area, compared to a MAC that is based on carrier sense multiple access with collision avoidance (CSMA/CA), as described in Section 1.2. In particular, CCT-based network flooding with OFDM can minimize the latency per hop in three ways. First, multiple nodes can start transmitting at the same time. Second, the waiting time for each medium access is kept at a minimum. Third, high data rates can reduce the transmission duration for a given amount of data.

This section discusses various tradeoffs between the key performance metrics of CCT-based network flooding, which are the *latency*, the *reliability* and the *throughput*. The low-level MAC protocol presented in Section 4.2.1 allows such performance tradeoffs to be tuned to some degree to meet specific requirements of various applications under the dURLLC paradigm, as introduced in Chapter 1. Since the low-level MAC protocol can be processed in real-time, as presented in Section 4.3, these tradeoffs can be tuned without additional signaling, while feedback from within the network may be considered.

7.1.4.1 *Latency*

The *latency* of a message delivery is the time span between the start of the transmission of a flooding frame by an initiator and the successful reception of the flooding frame by a particular node. Therefore, the *latency* is calculated as the *number of hops* required to reach the considered node multiplied by the *latency per hop*.

The *number of hops* depends on the *physical distance* of the considered node from the flooding initiator and on the *communication range* of the CCTs during the flooding process. The *physical distance* is typically rather limited in dURLLC applications since data are often only needed in the proximity of the nodes that generate them, as described in Chapter 1. The *communication range* of a CCT depends on the *modulation and coding scheme (MCS)* employed by the flooding initiator and on the *SINR* values at the individual receivers. The *MCS*, in turn, defines the PHY data rate and sets a limit for the minimum *SINR* required to decode a frame. The *SINR* values of the CCTs at the individual receivers, in turn, depend on the employed *transmission power* and on other factors and can take a wide range according to the discussion in Section 7.1.1. Still, the *communication range* of the CCTs of a flooding process in a dense wireless multi-hop network (WMN) can be expected to be quite consistent across successive hops, as discussed in Section 7.1.2.

The *latency per hop*, in turn, is calculated as the sum of the *transmission duration* of the *PHY protocol data unit (PPDU)* and the duration of the subsequent *idle period* before the following RTx, the latter of which is a short interframe space (SIFS) of 16 µs. The *PPDU transmission duration*, in turn, can be split into a fixed and a variable part. The fixed part of the *PPDU transmission duration* comprises the PHY preamble and the PHY header without the SERVICE field, i.e., the legacy short training field (L-STF), the legacy long training field (L-LTF) and the legacy SIGNAL field (L-SIG) with a total duration of 20 µs for an IEEE 802.11 Non-HT PPDU (see Section 2.4.2). The variable part of the *PPDU transmission duration* depends on the number of *DATA* OFDM symbols, which in turn depends on the length of the *PHY service data unit (PSDU)* and on the *MCS*. With this, the employed *MCS* has an influence on both the *number of hops* required to reach a particular node and on the *latency per hop*, in addition to its influence on the key performance metrics *reliability* and *throughput* that are discussed subsequently.

The *latency* of a CCT-based network flooding process can be tuned by exerting influence on the *number of hops* that is required to reach a particular node or on the *latency per hop*. While the *number of hops* can be reduced through a more robust *MCS*, this also increases the *PPDU transmission duration*, so the overall effect on the *latency* would be unclear and would have to be investigated further. Nevertheless, the *MCS* should generally be selected according to the node density and the overall channel conditions of a WMN. The *number of hops* can further be reduced through a higher *transmission power*.

The *latency per hop*, in turn, can be reduced by tuning the variable part of the *PPDU transmission duration*, which can be achieved by employing a higher *MCS* or by using a shorter *PSDU* in each flooding frame. A higher *MCS*, however, would also increase the *number of hops*, resulting in the unclear tradeoff described above. At the MAC layer, the *PSDU* corresponds to a *MAC protocol data unit (MPDU)*, which in turn comprises a MAC header with a fixed number of octets and a *MAC service data unit (MSDU)*. The number of octets inserted into the *MSDU* of a flooding frame, in turn, is under control of the initiator of a flooding process, so that it can tune the *latency per hop* by partitioning the data accordingly. However, while a fragmentation of data reduces the *latency* of disseminating each fragment, it also increases the overhead in terms of additional transmissions of PHY preambles, PHY headers and MAC headers as well as idle periods, so the key performance metric *throughput* is reduced by such measures.

The *PPDU transmission duration* can also be reduced through a wider *bandwidth* since more data can be transmitted in parallel, i.e., a given amount of data can be transmitted in a shorter time while using the same modulation scheme and the same coding rate. While the IEEE 802.11 Non-HT PHY allows for a 20 MHz wide channel [104, Section 17], the IEEE 802.11 HT PHY allows for 20 MHz and 40 MHz wide channels [104, Section 19] and the IEEE 802.11 VHT PHY allows for 20 MHz, 40 MHz, 80 MHz and 160 MHz wide channels [104, Section 21]. However, since the IEEE 802.11 HT and VHT PHYs employ larger PHY preambles and headers than the IEEE 802.11 Non-HT PHY, they can reduce the *latency per hop* only with a certain minimum size of the MSDU, depending on the MCS. Similar to employing a wider *bandwidth*, also *multiple input multiple output (MIMO) spatial streams* can be leveraged to attain more parallelism during data transmissions. The lowest *latency per hop* that can be achieved by the prototypes of this work is 44.2 µs with an IEEE 802.11 Non-HT PPDU at MCS 5 or higher, as illustrated in Figure 32.

7.1.4.2 *Reliability*

The *reliability* of a message delivery to a particular node is the probability of its success, which can be approximated empirically by means of the *frame reception rate (FRR)*. A frame is considered as received correctly if all its bits are recovered without errors. In CCT-based network flooding, a flooding process itself can be assumed to be persistent due to the redundancy of nodes at each hop in a dense WMN, as discussed in Section 7.1.6. Therefore, the *reliability* of a message delivery primarily depends on the ability of a node to recover the message by means of the respective CCTs generated by its neighbors. Note that a node typically has at least two opportunities to receive a flooding frame since different groups of neighbors may retransmit the frame successively. In addition, each node may retransmit a flooding frame multiple times, according to a setting of the low-level MAC protocol described in Section 4.2.1.3. Therefore, the *reliability* depends on the *probability of a correct frame reception* and on the *number of reception opportunities*.

In case of a CCT, the success of a frame reception depends on the interference occurring over time as a function of the TOs, the POs and the CFOs of the concurrent transmitters, as analyzed in Section 3.4. However, in case of a CCT with CI, the CFO between the concurrent transmitters is $\Delta f \simeq 0\,\text{Hz}$, so that the mutual interference of the transmitters becomes independent of the OFDM symbol index q, as can be seen in Equation 10. Thus, the impact of the mutual interference of the transmitters does not change over the duration of the frame, so that all bits are received under quite similar conditions. Consequently, the *probability of a correct frame reception* is the *bit reception probability* to the power of the *number of bits* of the frame, as with a single transmitter.

The *number of bits* of a flooding frame is a direct function of the number of bytes that the initiator of a flooding process inserts into the *MSDU*. Using a smaller *MSDU size* thus increases the *probability of a correct frame reception* during each CCT with CI, i.e., it increases the *reliability* of a message delivery through CCT-based network flooding. Besides, a smaller *MSDU size* reduces the key performance metrics *latency* and *throughput*. Note that the URLLC reliability requirement is defined for an *MSDU size* of 32 bytes [3].

The *bit reception probability* is complementary to the *BER* and is a function of the employed *modulation scheme*, the employed *coding rate* and the available *channel capacity*. Note that the selected *MCS* defines a combination of a *modulation scheme* and a *coding rate*, as listed in Table 2 and in Table 20 for the IEEE 802.11 Non-HT and HT PHYs, respectively. The *channel capacity* is defined as the maximum mutual information between an input and an output [158, Section 1.12], which can be increased by employing a *wider bandwidth* or by enhancing the *SNR* at the receiver. Thus, for a given bandwidth, the selected *MCS* sets a lower limit on the *SNR* or, in case of a CCT with CI, the *SINR* to attain a certain *BER*. With this, the *bit reception probability* can be increased by an *underutilization* of the available *channel capacity*, which can be achieved as follows. First, a more robust *MCS* can be used. Second, the *bandwidth* can be increased. Third, the *SINR* can be enhanced, e.g., by using a higher *transmission power*. In addition, the results presented in Section 5.4.4 indicate that *spatial diversity gains* can also be exploited. However, a received CCT with CI has a wider range of possible *SINR* values than a transmission from a single node and may by chance even have a very low *SINR* in a few cases due to the randomness of the POs between the concurrent transmitters, as described in Section 7.1.1. For successive CCTs of a particular flooding frame, the *bit reception probability* may vary accordingly.

The *number of reception opportunities* of a flooding frame at a particular node depends on three factors. First, the considered node may repeatedly receive the flooding frame from different groups of neighbors that retransmit the flooding frame for the first time in their respective areas during the respective flooding process. The communication ranges of these groups of neighbors overlap with each other, so that the flooding frame can propagate through the network. Hence, the considered node can receive the flooding frame from at least two groups of neighbors, unless it resides at the network border, where only one group may retransmit the flooding frame. Second, a few neighbors of the considered node may fail to decode the flooding frame successfully upon their first reception opportunities, so that they join the flooding process one or a few Tx slots later, as described in Section 7.1.3. Since these neighbors lag behind in the flooding process, their retransmissions of the flooding frame may create additional reception opportunities for the considered node. Third, each flooding frame can be retransmitted a specified number of times according to the low-level MAC protocol, as described in Section 4.2.1.3.

Overall, the *reliability* is the result of a certain *number of reception opportunities*, each of which has a certain *probability of a correct frame reception*. For a particular CCT with CI, this probability is typically similar at most nodes in a certain area, but there may also be nodes with a higher or with a lower probability according to the SINR characteristics described in Section 7.1.1. Similarly, the *probability of a correct frame reception* at a particular node may vary accordingly for successive CCTs with CI.

In addition to these considerations, sporadic interference from external devices may intermittently distort the reception of a CCT. Section 4.4.1 shows that additional RTx repetitions can help to overcome such interference in practice. Section 4.2.1.5 discusses several side effects of the tuning of the *reliability* through additional RTx repetitions in a network flooding scenario. The enhancement of the *reliability* through additional RTx repetitions is emulated by means of virtual RTx repetitions in the ping-pong experiments presented in Section 4.5 and in Section 5.4, respectively. These experiments also show that the Viterbi algorithm with soft-decision decoding [158, Section 12.3] provided by the Matlab WLAN toolbox [150] has superior performance.

Apart from CI with perfect frequency synchronization, the analysis in Section 3.4.2 shows that the FEC helps to decode CCTs with small CFOs between the concurrent transmitters, which also contributes to the *reliability* of a message delivery in practice. Moreover, Chapter 6 presents time-variant zero-forcing (TVZF), which can recover CCTs with incoherent interference that conventional receivers are not able to decode.

The prototypes of this work attempt to decode a flooding frame upon each reception opportunity and receive the flooding frame if they succeed in any of these attempts. A receiver could potentially enhance its reception performance further by combining the received CCT signals through maximum-ratio combining (MRC) [133, 160, 212].

7.1.4.3 *Throughput*

The *throughput* of a CCT-based flooding process is the *number of MSDU bytes* disseminated in a WMN in relation to the *time span* from the beginning of the respective flooding process until the point in time at which the next CCT-based flooding process is initiated. The *time span* is calculated as the *latency per hop* multiplied by a certain *number of Tx slots*. The *latency per hop* depends on the *number of MSDU bytes* and on the *PHY data rate*, i.e., on the selected *MCS*, as described in Section 7.1.4.1. The *number of Tx slots*, in turn, must be selected such that the CCTs generated by successive flooding processes do not collide with each other, i.e., all nodes must be able to receive the CCTs of a particular flooding process without interference from a previous or a subsequent flooding process. The *number of Tx slots* required between the initiations of successive flooding processes therefore depends on whether they are initiated by the same node or by different nodes.

If the same node initiates successive flooding processes, it can optimize the *throughput* through flooding process pipelining, allowing multiple flooding processes to be in transit. Each generation of an RTx of a flooding frame occupies two Tx slots, one for the reception and one for the transmission. The maximum number of CCTs that can be generated by a node is one plus the number of additional RTx repetitions, as presented in Section 4.2.1.3. In addition, a node may miss a few reception opportunities, as discussed in Section 7.1.3. In total, the *number of Tx slots* required by a node upon the arrival of a flooding process in its respective area is the number of possibly missed first reception opportunities plus twice the number of RTx that the node may generate plus one guard slot.

If different nodes in different locations of a WMN initiate successive flooding processes, each flooding process should finish before the respective next flooding process starts. Therefore, the total *number of Tx slots* is the sum of the number of Tx slots during which a flooding process propagates throughout the network, as given by Equation 42, and the number of Tx slots for additional RTx repetitions at the end of a flooding process, as given by Equation 43. In practice, the nodes may employ a high-level MAC protocol in order to coordinate the initiations of their respective flooding processes with each other, as further discussed in Section 7.3.2. Section 4.2.1.5 discusses how the `rtx_rep_per_node` and the `rtx_global_counter` fields of the low-level MAC protocol affect the efficiency of the channel utilization and thus the *throughput*. A variation of a high-level MAC protocol may define only the order in which the nodes initiate their respective flooding processes, but not the particular points in time at which their flooding processes should start. This would allow an optimization of the *throughput* through transmission pipelining, i.e., the node that is scheduled next could initiate its flooding process immediately when the flooding process of the respective previous node has passed by.

Overall, the *throughput* can be increased by employing a higher *PHY data rate* by means of a higher *MCS* since this reduces the *latency per hop*, as discussed in Section 7.1.4.1. In addition, the *throughput* can be increased by using a larger *number of MSDU bytes* in each flooding frame since this reduces the overhead of additional idle periods as well as of additional transmissions of PHY preambles, PHY headers and MAC headers. Further, the *throughput* can be increased by keeping the *number of Tx slots* of each flooding process as small as possible. While the *number of Tx slots* required by each flooding process depends on the respective scenario, as discussed above, it can be reduced by reducing the number of additional RTx repetitions. Note that an increase of the *throughput* by any of these three means goes at the expense of the *reliability*.

7.1.5 *Comparison with Glossy*

Glossy [73] is a remarkable technology that enables fast and reliable network flooding as well as global time synchronization throughout a wireless sensor network (WSN). Glossy is implemented for low-power Tmote Sky sensor nodes that are equipped with an MSP430 microcontroller and a CC2420 radio. An important design criterion of this underlying hardware platform is its energy efficiency, allowing a device to be powered via Universal Serial Bus (USB) or by a battery pack. Accordingly, the computational capacities of such a device in terms of processing power and available memory are rather limited. In contrast, the solutions devised in this work shall not be subject to such restrictions. Instead, the prototypes of this work leverage the capabilities of powerful processors, signal processing algorithms running on a field-programmable gate array (FPGA) and specialized IEEE 802.11 hardware, as presented in Chapter 4 and in Chapter 5.

Glossy employs the IEEE 802.15.4 offset QPSK (O-QPSK) PHY [107, Section 12], which is based on the DSSS modulation technique and which has a fixed PHY data rate of 250 kbit/s in the 2.4 GHz band. The prototypes of this work employ the OFDM-based IEEE 802.11 Non-HT and HT PHYs, offering different PHY data rates, as listed in Table 2 and in Table 20, respectively. Using a 40 MHz wide channel and 4 MIMO spatial streams, the PHY data rate of CCTs with CI can be as high as 324 Mbit/s at an FRR of 99.5 %, as shown in Table 21. This PHY data rate is 1296 times faster than that of Glossy.

The latency per hop of Glossy [73] is calculated as the processing delay of the radio at the beginning of a frame reception plus the software execution delay plus the PPDU transmission duration. The processing delay of the CC2420 radio is 2 µs [40, Section 6.3]. The software delay of Glossy amounts to ∼23.5 µs [73, Figure 9]. An IEEE 802.15.4 PPDU consists of a synchronization header, a PHY header and a PSDU [107, Section 12]. The synchronization header, in turn, comprises a PHY preamble consisting of 4 octets and a start-of-frame delimiter consisting of 1 octet. The PHY header consists of 1 octet. With this, an IEEE 802.15.4 PPDU contains 6 octets plus a PSDU of up to 128 octets. The PSDU, in turn, comprises a few protocol fields of Glossy, namely a header field that is used to identify Glossy frames as well as a footer with a relay counter field and fields with a received signal strength indicator (RSSI) and a cyclic redundancy check (CRC). These protocol fields of Glossy occupy 4 octets of the PSDU and roughly correspond to certain functions of the IEEE 802.11 MAC header in combination with the low-level MAC protocol presented in Section 4.2.1. Consequently, the latency per hop of an 8-byte MSDU is 2 µs + 23.5 µs + 576 µs = 601.5 µs. Similarly, the latency per hop of a 32-byte MSDU is 1369.5 µs, while the latency per hop of a 124-byte MSDU is 4313.5 µs.

With the WARP prototype presented in Chapter 4 using the IEEE 802.11 Non-HT PHY, the latency per hop ranges from 88.2 µs at MCS 0 to 44.2 µs at MCS 7 for an 8-byte MSDU, from 120.2 µs at MCS 0 to 48.2 µs at MCS 7 for a 32-byte MSDU and from 244.2 µs at MCS 0 to 60.2 µs at MCS 7 for a 124-byte MSDU. In comparison to Glossy, the latency per hop can thus be reduced by a factor of 6.8 to 13.6 for an 8-byte MSDU, by a factor of 11.4 to 28.4 for a 32-byte MSDU and by a factor of 17.7 to 71.7 for a 124-byte MSDU.

The reliability of Glossy is assessed in a testbed experiment with 92 nodes, while the initiators of the respective flooding processes are selected randomly [73, Section 7.2]. The maximum hop distance of any node in the network from the initiator is 5 hops and the PSDU contains 8 octets. The average FRR is 99.6 % with a single RTx per node, while it increases with additional RTx repetitions per node, reaching 99.999 % at five RTx per node [73, Figure 15]. The testbed experiments presented in Section 4.5.2, Section 5.4.1 and Section 5.4.4.2 show that the prototypes of this work can reach an FRR of 99.999 % already with a single RTx per node when a receiver has at least two opportunities to receive a flooding frame from different groups of nodes that forward it successively.

7.1.6 *Meeting the URLLC Requirement*

The 3rd Generation Partnership Project (3GPP) incorporates Ultra-Reliable Low-Latency Communications (URLLC) as one of three main service categories of fifth generation (5G) cellular networks [3]. This thesis introduces decentralized URLLC (dURLLC) as a similar paradigm in Chapter 1 and uses the URLLC requirement as a performance benchmark. The 3GPP states that "reliability can be evaluated by the success probability of transmitting X bytes within a certain delay, which is the time it takes to deliver a small data packet from the radio protocol layer 2/3 SDU ingress point to the radio protocol layer 2/3 SDU egress point of the radio interface, at a certain channel quality" and that "a general URLLC reliability requirement for one transmission of a packet is $1 - 10^{-5}$ for 32 bytes with a user plane latency of 1 ms" [3]. Accordingly, a CCT-based flooding process with a 32-byte MSDU must reach a destination node within 1 ms after its initiation and the destination node must receive the flooding frame with a probability of at least 99.999 %.

A CCT-based flooding process can be expected to be such persistent in a dense WMN that it does not become a limiting factor in meeting the URLLC reliability requirement, due to the following reasons. Most CCTs with CI have a high SINR in a narrow range, while there may be a few cases with a slightly higher or with a lower SINR by chance, as discussed in Section 7.1.1 and as shown in Figure 46 and in Figure 58, respectively. These SINR characteristics are rather independent of the number of transmitters of a CCT, so that the signal quality is quite independent of the node density of a WMN. Therefore, in a dense WMN, a high fraction of nodes at each hop can receive a particular CCT correctly and forward the flooding frame as a CCT, as discussed in Section 7.1.2. Further, the signal quality of CCTs generated by successive hops is quite consistent, as shown in Figure 44 and in Figure 56, respectively. Moreover, each network node may have multiple opportunities to receive a flooding frame, as discussed in Section 7.1.4.2. If the overall conditions of a CCT-based flooding process are such good that an arbitrary node has an FRR of more than 99.999 % according to the URLLC reliability requirement, all nodes that contribute to the flooding process effectively experience the same excellent conditions, so that the persistence of the flooding process itself is even much better. Consequently, the reliability of a CCT-based flooding process at a particular destination node depends on the ability of the destination node to receive the flooding frame by means of the CCTs generated by its respective neighbors, as discussed in Section 7.1.4.2. The number of CCT reception opportunities, in turn, can be tuned by letting each node generate additional RTx repetitions after its first RTx, as described in Section 4.2.1.3.

The testbed experiments presented in Section 4.5.2, in Section 5.4.1 and in Section 5.4.4.2 evaluate the achievability of the URLLC reliability requirement with the WARP prototype presented in Chapter 4 and with the Asus RT-AC86U prototype presented in Chapter 5, respectively. Each experiment is conducted with 100 runs of 10 000 successive CCTs generated as ping-pong transmissions between two groups of five nodes each.

The fact that the nodes can generate 10 000 successive CCTs confirms that CCT-based flooding processes are persistent. Note that the conditions of these flooding processes are worse than the conditions of a flooding process in a dense WMN, as the generation of each CCT is triggered through a single CCT reception opportunity at only five nodes. While the WARP prototype nodes may additionally generate a retryRTx upon a failed CCT reception to improve the persistence of a flooding process, as described in Section 4.2.1.4, the Asus RT-AC86U prototype does not have this feature.

Section 4.5.2 describes the methodology that is applied in the testbed experiments for the systematic evaluation of the achievability of the URLLC reliability requirement. In particular, the FRR is calculated with a variable number of virtual RTx repetitions for receptions from each of both groups and for receptions from any of both groups, the latter of which emulates the conditions of CCT-based network flooding in a WMN. For receptions from any of both groups in the experiment with WARP prototype nodes, Table 5 shows that a line-of-sight (LOS) receiver meets the URLLC reliability requirement without additional RTx repetitions at Non-HT MCS 0–4, while a non-line-of-sight (NLOS) receiver meets this requirement without additional RTx repetitions at Non-HT MCS 0–2. The experiments presented in Section 5.4.1 and in Section 5.4.4.2 show that the URLLC reliability requirement is also fulfilled with CCTs from Asus RT-AC86U prototype nodes and that CCT with MIMO spatial streams allows spatial diversity gains to be leveraged. Table 6 lists the latency per hop with a 32-byte MSDU for the Non-HT and the HT PHYs.

7.2 Core Technology

Enabling the CCT communication paradigm through prototypes and practical solutions involves various technologies and concepts. This section highlights the core technologies that enable CCT-based communications in practice and discusses various design decisions, potential enhancements and their respective performance implications.

Section 7.2.1 highlights the advantages of using the OFDM-based IEEE 802.11 Non-HT and HT PHYs in this work and discusses the feasibility of CCT with other OFDM PHYs. Section 7.2.2 highlights the differences between CCT with and without CI and shows that the devised solutions that enable these modes of operation are complementary. Section 7.2.3 discusses the time synchronization mechanisms employed in this work. Section 7.2.4 discusses the frequency synchronization mechanisms used in this work. Section 7.2.5 discusses the relevance of FEC for attaining a high reliability at a receiver. The low-level MAC protocol presented in Section 4.2.1 is discussed in Section 7.1.4.

7.2.1 *Physical Layer*

The prototypes and solutions of this work demonstrate the practical feasibility of CCT with OFDM by means of the IEEE 802.11 Non-HT and HT PHYs. On the one hand, retaining compatibility to the IEEE 802.11 PHYs has certain advantages with regard to the implementation of the developed prototypes. In particular, the WARP prototype presented in Chapter 4 is based on the IEEE 802.11 reference design for WARP v3 [247], whereas the Asus RT-AC86U prototype presented in Chapter 5 leverages the capabilities and some special functions of its integrated BCM4365E chip.

On the other hand, retaining compatibility to the IEEE 802.11 PHYs also has advantages for conducting experiments. First, commercial devices can be employed to characterize the synchronization requirements in testbed experiments, as presented in Section 3.3.3 and in Section 3.4.3, respectively. Second, the signal quality of CCTs that are generated by the prototypes of this work can be assessed by means of the FRRs of commercial devices, as presented in Section 4.5.1 and in Section 4.5.5, respectively. Third, since commercial devices can receive the CCTs generated by the prototypes of this work, they can also be employed as receivers in practical applications. In fact, multiple WARP prototype nodes can disseminate a video stream over multiple hops, while the video stream can be injected and received by commercial devices [166]. Fourth, the reception performance of the TVZF receiver can be compared to that of commercial devices, as shown in Section 6.4. Fifth, the TVZF receiver can be demonstrated to successfully decode the CCTs generated by unsyntonized commercial devices, as presented in Section 6.5.

However, retaining compatibility to the IEEE 802.11 Non-HT PHY brings challenges when decoding a CCT with incoherent interference since the received superimposed signal is subject to the mutual interference of its CCT signal components, which inhibits decoding with a conventional receiver. The TVZF receiver presented in Chapter 6 applies rather complicated processing steps to estimate the number of CCT signal components and their respective parameters so that it can reconstruct the interference patterns and equalize a received frame accordingly. While SourceSync [186] also addresses CCT in IEEE 802.11 networks, it modifies the IEEE 802.11 Non-HT PHY in order to synchronize the concurrent transmitters in time and frequency, as discussed in Section 2.1.7.1.

While the IEEE 802.11 Non-HT and HT PHYs are extensively employed in this work, the devised solutions are not limited to these PHYs and should also work with OFDM PHYs with a larger bandwidth. The IEEE 802.11 VHT PHY [104, Section 21], which is part of IEEE 802.11ac, and the IEEE 802.11 high efficiency (HE) PHY [105, Section 27], which is part of IEEE 802.11ax, both allow for a channel bandwidth of up to 160 MHz. While these PHYs offer higher data rates, their PHY preambles and PHY headers are longer than the ones of the IEEE 802.11 Non-HT PHY due to backwards compatibility. Therefore, the IEEE 802.11 VHT and HE PHYs can only offer performance gains when an MSDU has a certain minimum size, as discussed in Section 7.1.4.

The IEEE 802.11 HE PHY allows for multi-user MIMO (MU-MIMO) in the uplink (UL), i.e., multiple nodes can concurrently send to an access point (AP) [105, Section 27.3.3.2]. In this process, an AP coordinates an MU-MIMO UL transmission by broadcasting a trigger frame, which the respective nodes use for synchronization in time and frequency before they transmit their respective data by means of their assigned spectral resources. However, the IEEE 802.11 HE PHY does not facilitate CCT-based network flooding. Still, since mechanisms for synchronization in time and frequency must be present, commercial IEEE 802.11ax devices are a reasonable choice for developing a prototype for CCT-based network flooding through reverse engineering, like shown in Chapter 5. Regarding the accuracy of the frequency synchronization, the IEEE 802.11ax standard requires that at least 90 % of the absolute CFO values between response PPDUs and their respective trigger PPDUs should not exceed 350 Hz in case of HE PPDUs and 2 kHz in case of Non-HT PPDUs [105, Section 27.3.15]. The frequency synchronization mechanism of the WARP prototype exceeds this requirement, as shown in Figure 41. Further, the IEEE 802.11ax standard requires that the transmission of a response PPDU starts within 16 µs $\pm$ 0.4 µs after the respective trigger PPDU [105, Section 27.3.15]. The measured IFS values of both prototypes undercut the tolerable range of 0.8 µs, as visible in Figure 37 and in Figure 51, respectively.

A custom OFDM PHY could be designed to allow for an even lower latency per hop. Note that the PHY preamble, the PHY header and the SIFS duration of the IEEE 802.11 Non-HT PHY set a lower limit for the latency per hop, as discussed in Section 7.1.4.1. An OFDM PHY with a shorter PHY preamble, a shorter PHY header and a shorter IFS would overcome this limitation, while simultaneously enhancing the throughput and the reliability of CCT-based network flooding as a side effect, as discussed in Section 7.1.4. However, a prototype that can process the frames accordingly fast would also be required.

7.2.2 *Coherent Interference and Incoherent Interference*

This work investigates the feasibility of CCT with coherent interference (CI) and with incoherent interference, respectively. Figure 27 conceptually illustrates the goals of the two different approaches by means of the FRR characteristics of a receiver, shown as a function of the TO and the CFO between two concurrent transmitters.

CCT with CI essentially relies on frequency synchronization between concurrent transmitters, so that the phase relations between the respective transmitters are consistent throughout the duration of a transmitted frame. With perfect frequency synchronization, a CCT with CI resembles a transmission from a single node with multiple antennas, while the respective radio frequency (RF) chains are driven by a common clock source.

Figure 27a illustrates the goal stated in Section 1.3.2 for the IEEE 802.11 Non-HT PHY, showing that both the TO and the CFO between two concurrent transmitters must not exceed certain boundaries to obtain a superimposed signal that can be decoded by a conventional receiver. In particular, the TO must not exceed the guard interval (GI) duration to avoid intersymbol interference (ISI) between successive OFDM symbols, whereas the tolerable CFO generally depends on the frame length and other factors. In case of nearly perfect time synchronization between the concurrent transmitters, i.e., when the TO is smaller than 61.5 ns, as expressed by Equation 15, the tolerable CFO additionally depends on the PO, which is random in practice, as described in Section 3.4.1. In case of imperfect time synchronization between the concurrent transmitters, i.e., when the TO ranges between 61.5 ns and 800 ns, symbol errors may occur, which, however, can be corrected to a certain degree through FEC, as described in Section 3.4.2.

CCTs with incoherent interference usually cannot be decoded by conventional receivers. Figure 27b illustrates the goal stated in Section 1.3.3 for the IEEE 802.11 Non-HT PHY, showing that an enhanced receiver implementation shall allow for a larger CFO between concurrent transmitters than a conventional receiver, while the requirement for time synchronization better than the GI duration of the employed OFDM PHY may remain. Chapter 6 presents TVZF equalization and a corresponding receiver design as a solution. With this, CCTs from unsyntonized devices can be received, widening the spectrum of CCT-based applications towards scenarios in which the transmitters cannot be accurately synchronized in frequency. The extensive broadcasting scenario presented in Section 1.2.2 is an example falling into this category, unless specialized network equipment with wired frequency synchronization is employed, as further discussed in Section 7.3.1.

While the two modes of operation of CCT with and without CI are quite different, especially with regard to the techniques that enable them, they are complementary to each other rather than mutually exclusive. The FRRs of conventional receivers are only high for small ranges of low absolute CFO values between concurrent transmitters, as visible in Figure 25 and in Figure 26. Furthermore, the analysis in Section 3.4 shows that the tolerable absolute CFO values become smaller with an increasing frame length. In contrast, the TVZF receiver works best when the absolute CFO between the transmitters exceeds a certain threshold, as can be seen in Figure 66, in Figure 67 and in Figure 68, since the iterative interference cancellation (IIC) algorithm presented in Section 6.3.2 can detect CCT signal components with small CFOs only if the frame is sufficiently long. In conclusion, the operational ranges of both modes of operation with respect to the CFO extend and contract in a complementary fashion as a function of the frame length. Therefore, an enhanced receiver design could employ TVZF in addition to a conventional processing chain to improve the reception performance if required.

7.2.3 *Time Synchronization*

A receiver has to acquire an accurate symbol timing of a received frame in order to avoid ISI during the demodulation of OFDM symbols. The detection of an incoming frame in a stream of in-phase quadrature (IQ) samples and the symbol timing estimation is performed by means of the IEEE 802.11 PHY preamble, as described in Section 2.4.3. The prototypes of this work additionally utilize the acquired symbol timing of a received flooding frame to generate a reception-triggered RTx with accurate timing.

The WARP prototype employs a hardware timer that runs automatically after each frame reception in order to trigger the transmission of an RTx, as described in Section 4.3.2. In addition, it employs another hardware timer that runs automatically after each frame transmission to trigger the transmission of a retryRTx, as described in Section 4.3.3. Further, Section 4.4.2 describes the timing calibration so that a retryRTx is generated with the same timing as an RTx. The Asus RT-AC86U prototype generates an RTx of a received flooding frame by means of the acknowledgment (ACK) engine of its IEEE 802.11 chip, which is also driven by a hardware timer, as described in Section 5.2.3.

The WARP prototype and the Asus RT-AC86U prototype both achieve consistent time synchronization in experiments in which each frame is transmitted by a single node, as shown in Figure 37 and in Figure 51, respectively. The timing characteristics of both prototypes are also consistent in ping-pong experiments with CCTs between two groups of five nodes each, as shown in Figure 43b and in Figure 55b, respectively.

The TVZF receiver employs an enhanced symbol timing estimator that is based on successive interference cancellation (SIC) to detect weak CCT signal components that arrive before CCT signal components with more power, as described in Section 6.3.1. The TVZF receiver shifts its start of frame (SOF) sample position accordingly so that the weak CCT signal components are covered by the GI filled with a cyclic prefix (CP). With this, ISI between the different CCT signal components can be avoided, which is important to detect them and to estimate their respective CCT parameters accurately. Figure 60 illustrates the operation of the SIC-based symbol timing estimator.

The SIC-based symbol timing estimator is also integrated into the IEEE 802.11 decoder that is based on the Matlab WLAN toolbox [150]. The IEEE 802.11 decoder can be configured to process a received frame twice, once with the symbol timing estimator of the Matlab WLAN toolbox and once with the SIC-based symbol timing estimator, so that the effects of these algorithms on the reception performance can be compared. When processing the datasets of the ping-pong experiments with CCTs between two groups of five nodes each, which are presented in Section 4.5 for the WARP prototype and in Section 5.4 for the Asus RT-AC86U prototype, respectively, both decoder settings produce almost identical results. While the SIC-based symbol timing estimator usually results in a marginally better SINR, it provides no significant gains for CCT with CI.

7.2.4 *Frequency Synchronization*

The key feature of CCT with CI is the frequency synchronization of the transmitters. Since each node estimates its CFO with respect to a transmitter during a frame reception, multiple nodes can leverage a common frame reception as a reference to synchronize in frequency with each other so that they can generate a CCT with CI shortly afterwards. Section 3.6 presents an enhanced CFO estimation technique that leverages the common phase error (CPE) of the pilot subcarriers to estimate the residual CFO after applying a fast Fourier transform (FFT) on the received OFDM symbols. This enhanced CFO estimation technique is integrated into the WARP prototype, as presented in Section 4.1.1. Since the enhanced CFO estimator uses each OFDM symbol to reduce the impact of noise, the frequency synchronization mechanism of the WARP prototype is most accurate when the PPDU is sufficiently long, as can be seen in Figure 41a and in Figure 41b.

However, Figure 41c also shows that the phase error at the end of a PPDU increases with an increasing frame length. The reason is that the carrier frequencies of the two employed WARP v3 software-defined radios (SDRs) [246] may change slightly over time due to oscillator jitter, as described in Section 2.2.2. Therefore, the CFO between the two WARP v3 SDRs may differ slightly at the points in time when the WARP SDR with WARPLab [250] sends a trigger frame and when the device under test (DUT) generates an RTx. Since such variations of the CFO cannot be mitigated by the enhanced CFO estimator, the phase error at the end of a PPDU may be larger for longer frames. To reduce this effect, oscillators with a more stable carrier frequency would be required. While the WARP v3 has a temperature-compensated crystal oscillator (TCXO) with good characteristics in terms of frequency accuracy and stability, as presented in Section 3.5.1, oscillators with an even better short-term frequency stability may exist.

In the testbed experiments with ping-pong transmissions between two groups of WARP prototype nodes, each CCT serves as a frequency reference for the generation of the next CCT, so that small synchronization errors can make the CFO drift over time. The WARP prototype provides a CFO containment mechanism that pulls the carrier frequency of a CCT-based flooding process towards the specified center frequency of the selected channel, as presented in Section 4.2.4 and as validated in Section 4.4.3. The progression of the CFO during an experiment run with ping-pong transmissions between two groups of five WARP prototype nodes each is shown in Figure 43a.

The Asus RT-AC86U prototype provides a frequency synchronization procedure that can be executed once before conducting testbed experiments, as described in Section 5.2. The accuracy of the frequency synchronization depends on the number of tuning frames, as shown in Figure 53. Further, the carrier frequency is reasonably stable over more than one hour when the room temperature is also quite stable, as shown in Figure 50. The progression of the CFO during an experiment run with ping-pong transmissions between two groups of five Asus RT-AC86U prototype nodes each is shown in Figure 55a.

7.2.5 *Forward Error Correction*

While CCT with CI can be achieved by means of frequency synchronization mechanisms, the concurrent transmitters typically still have small CFOs with respect to each other. Under conditions of a small TO and a small CFO, symbol errors may successively start to occur on neighboring OFDM subcarriers, as shown by the analysis in Section 3.4.2. The occurrence of symbol errors on individual OFDM subcarriers is shown in Figure 24. Such symbol errors can be corrected to a certain degree through FEC.

In Section 4.5.1, the Viterbi algorithm with soft-decision decoding [158, Section 12.3] of the Matlab WLAN toolbox [150] has a superior reception performance in comparison to the commercial devices and the WARP prototype nodes. Accordingly, the reception performances of the WARP prototype and of the TVZF receiver could potentially be enhanced through an integration of the Viterbi algorithm with soft-decision decoding.

Low-density parity check (LDPC) codes are supported by the IEEE 802.11n/ac/ax standards and also improve the reception performance, as indicated by Co-BCast [184]. Further, LDPC codes are going to be used by IEEE 802.11bd in vehicular networks [262]. Accordingly, LDPC codes may also facilitate CCT-based communication schemes in mobile environments, in which CFOs may be introduced due to the Doppler effect [218].

7.3 Technology Extensions

This work provides solutions to enable CCT with the OFDM-based IEEE 802.11 PHYs. While the practical feasibility of CCT is shown through various testbed experiments, the WARP prototype presented in Chapter 4 also allows for CCT-based network flooding over multiple hops by means of the low-level MAC protocol presented in Section 4.2.1. Further, it provides a MAC header format that allows commercial devices to inject flooding frames into a network, in addition to the MAC header format that the WARP prototype nodes employ for CCT-based network flooding, as presented in Section 4.2.2. With this, the WARP prototype can enable various real-world dURLLC applications. However, the dURLLC paradigm may also comprise scenarios in which multiple nodes have to disseminate their messages in a coordinated fashion or in which pairs of nodes exchange data over several hops by means of unicast routes, as described in Chapter 1. This section discusses technology extensions that add support for more complex dURLLC applications and that aim to optimize the performance of CCT-based communications.

Section 7.3.1 discusses the extensive broadcasting scenario introduced in Section 1.2.2. Section 7.3.2 discusses a high-level MAC to coordinate CCT-based flooding processes. Section 7.3.3 discusses CCT-based multi-hop routing to support unicast traffic efficiently. Section 7.3.4 discusses the streaming of media through CCT-based network flooding. Section 7.3.5 discusses further performance optimizations of CCT-based network flooding. Section 7.3.6 discusses security extensions.

7.3.1 *Extensive Broadcasting with Wired APs*

The extensive broadcasting scenario presented in Section 1.2.2 comprises a centralized server that has wired connections to several wireless APs, e.g., by means of Ethernet [102]. In certain time intervals, the server obtains data that should be disseminated through wireless broadcasts in certain areas that exceed the communication range of a single AP. In this process, the APs should use the spectral resources of the wireless channel most efficiently, which can be achieved by letting multiple APs send the same data as CCTs, possibly with a limited number of repetitions to balance reliability and throughput.

To facilitate CCT with CI in this scenario, the APs must be synchronized in time and frequency with each other, which could be done by means of the Ethernet connections. Since the Ethernet broadcast frames from the centralized server typically arrive at quite the same times at the APs, they could serve as a reference for the time synchronization. Further, synchronous Ethernet (SyncE) could be used to synchronize the APs in frequency, as described in Section 2.2.6.3. However, SyncE requires specialized network equipment, so this solution might be considered to be expensive and inflexible.

To achieve CCT with CI through wireless synchronization, the APs could initially establish synchronization in time and frequency through a CCT-based flooding process. Afterwards, the APs would be split into two groups that alternatingly broadcast the data from the centralized server as CCTs. With this, the group that does not transmit can synchronize in frequency by means of the CCT from the other group, respectively. However, the scalability of this approach over an extensive area has to be explored. Alternatively, with receivers that integrate the TVZF equalizer presented in Chapter 6, this scenario could also be enabled through CCT with incoherent interference.

7.3.2 *High-Level MAC*

A CCT-based flooding process extends as a concentric circle around the flooding initiator and occupies the wireless medium in the respective area for a limited amount of time. Due to the persistence of repetitive CCTs, a CCT-based flooding process may in fact impede any other communications on the same frequency channel in its respective area. Thus, if conventional communications should take place between CCT-based flooding processes or if different nodes should be allowed to initiate CCT-based flooding processes, these nodes must coordinate with each other through a high-level MAC protocol.

A decentralized high-level MAC protocol could allow each node to reserve the wireless medium autonomously within a certain area and for a specified time interval, respectively. To this end, a node announces its reservations in the flooding frames that it disseminates throughout the WMN so that all other nodes know when the future CCT-based flooding processes will arrive, allowing them to suspend their regular communications temporarily. With this, a reservation resembles a network allocation vector (NAV) [104, Section 10]. To facilitate reservations of the wireless medium through a CCT-based flooding process, all nodes must acquire a common time reference with respect to the flooding initiator, which can be provided through the flooding frame itself, as described in Section 4.5.1. To establish fairness between the nodes in a WMN, certain policies can be implemented to make each node consider the traffic demands announced by the other nodes.

When making a reservation, a node must take all the reservations of the other nodes into account to avoid a collision with any other scheduled CCT-based flooding process. If the size of a WMN is limited and if each node announces its reservations globally, i.e., to all other nodes, collisions between CCT-based flooding processes can be avoided. However, if reservations are announced only throughout a limited number of hops, some nodes may have incomplete information of the reservations in the WMN, so that they may schedule CCT-based flooding processes that would collide with other ones. To avoid such impending collisions, all network nodes that forward a flooding frame with new reservations have to check the validity of these reservations immediately and possibly issue a collision warning that should quickly propagate back to the flooding initiator so that the new reservation can be discarded. If multiple nodes detect the same impending collision, they may synchronously send the same collision warning as a CCT. However, if neighboring nodes detect different impending collisions, their transmitted collision warnings themselves may collide. In this case, the particular nodes can detect their collision as their respective neighbors might not retransmit the collision warnings. Thus, a node could retry to send a collision warning after a short random backoff time. Alternatively, a node could piggyback a collision warning on an own flooding frame that is scheduled early enough to reach the node that tries to make the new reservation.

As opposed to a decentralized high-level MAC, CCT-based flooding processes can also be coordinated by a centralized controller, like with Low-Power Wireless Bus (LWB) [72]. LWB treats the wireless medium like a shared bus that covers an entire WMN and that can be accessed by all nodes through CCT-based network flooding (see Section 2.3.1). Since LWB employs Glossy [73], it can leverage the capture effect during a contention slot in which the nodes may request new medium allocations. However, since OFDM is more susceptible to interference than DSSS (see Section 2.1.2), the procedure for requesting medium allocations would have to be redesigned for an OFDM-based PHY.

7.3.3 *Routing*

Some dURLLC applications may have to exchange data only between a pair of nodes rather than having to disseminate them throughout an entire area via network flooding. Section 2.3.2 discusses different variations of routing algorithms for WMNs and highlights OLA Routing On-Demand (OLAROAD) [47, 224] as a CCT-based solution that can overcome certain limitations of conventional routing algorithms. OLAROAD essentially constructs a corridor of nodes that forward the data between two nodes through CCTs. With this, the delivery of a message effectively works like a CCT-based flooding process in a geographically restricted area. The forwarding nodes are selected according to their respective distances in terms of the hop count from the two nodes that attempt to construct the route, so that the corridor width can be flexibly adjusted.

CCT-based routing could well be integrated with a decentralized higher-level MAC as proposed in Section 7.3.2. With this, a node can disseminate a route request (RREQ) and receive a route reply (RREP) from the other node while it has the medium reserved. Furthermore, the RREQ and RREP messages can instantaneously reserve the medium on the corridor between the two nodes during specified time intervals for data transfers. The nodes on the corridor check new medium reservations when processing the RREP. If a node already has a medium reservation for a time interval that is also requested for the route, it generates a collision warning, similarly as discussed in Section 7.3.2. However, in this case, a collision warning propagates into both directions so that both nodes that construct the respective route are informed about the impending collision. If no collision warning is generated, the route is successfully constructed and reserved, so that the two nodes can use it for data transfers afterwards. Further, the nodes can piggyback new medium reservations on the data frames while using the route.

In CCT-based routing, multiple routes can coexist since each route uses the medium only in a certain area at specific times. Thus, routes in different areas of a WMN can transfer data simultaneously without interfering with each other. Additionally, routes may overlap with each other as they are exclusively active during reserved time intervals. Moreover, CCT-based routing can also be extended to support multicast traffic.

With a larger WMN size and with more nodes, the routes may be longer and the traffic load may be higher. Since each node may have incomplete information of the medium reservations in the WMN, reserving the medium might become increasingly difficult. A better scalability can potentially be achieved by partitioning a WMN into clusters. With this, each CCT-based flooding process and each CCT-based message delivery is restricted to its respective cluster so that each medium reservation involves fewer nodes. If a node repeatedly fails to reserve the medium, another node can transfer one of its own medium reservations to that node in accordance with certain fairness policies. Alternatively, all medium reservations within a cluster can be announced globally through CCT-based network flooding so that all nodes have the same information.

A high-level routing algorithm can be employed to transfer data also between nodes in different clusters. In such a scheme, several nodes at the border between two clusters can receive a message from one cluster, buffer it and forward it into the next cluster. Any node of a cluster that has received a message from another cluster can piggyback a new medium reservation on one of its own flooding frames in order to schedule another CCT-based flooding process that shall transfer the buffered message through the cluster.

Alternatively, the node can also construct a CCT-based route into the area of the cluster in which the nodes reside that can forward the message into the respective next cluster. At the start of the newly reserved time interval, all nodes that have the message in their buffers retransmit it as a CCT so that the message propagates through the cluster. To ensure that each cluster can forward the message into the respective next cluster, the clusters must overlap and use different frequency channels so that their CCTs do not interfere with each other. The routing decisions of the clusters, in turn, could be based on the geographical cluster locations, which minimizes the routing overhead.

CCT-based routing can leverage the redundancy of nodes to improve the reliability of a message delivery not only through CCT-based communications within each cluster but also through the buffering and the forwarding of a message by groups of nodes between successive clusters in a WMN. With this, CCT-based routing has the potential to overcome scalability problems of conventional routing algorithms that are based on a CSMA/CA MAC, as described in Section 2.3.2. Still, there are various design options for a CCT-based routing protocol, which are the subject of further research.

7.3.4 *Media Streaming through CCT-Based Network Flooding*

CCT-based network flooding can facilitate the streaming of media like television and radio programs in an extensive area without a permanently installed infrastructure. With this, it can be employed to provide media streams in areas in which an installation of a television tower is not economic, like in remote villages or on secluded islands. Furthermore, it can also replace or supplement infrastructure-based media broadcasts in catastrophic scenarios to improve the resilience of information services.

The low-level MAC protocol of the WARP prototype allows an initiator to tune certain parameters of a CCT-based flooding process, such as the area coverage, the throughput and the reliability, without additional overhead for signaling, as described in Section 4.2.1. With this, certain parameters can be changed dynamically, while different media streams can also be operated with different parameters. Frames that are encoded in the flooding feed format, which is described in Section 4.2.2.1, may not only be processed by the WARP prototype nodes themselves but they can also be received by passive commercial devices. Furthermore, media streams can also be injected by a commercial device by means of the frame injection format, which is described in Section 4.2.2.2.

The feasibility of media streaming through CCT-based network flooding by means of the WARP prototype is explored in testbed experiments [166]. A commercial device with an Atheros AR928X wireless network adapter injects a video stream that is forwarded by nine WARP prototype nodes over multiple hops throughout an office environment [166]. To mitigate mutual interference between successive flooding processes, the source node injects them with a waiting time of 3 Tx slots, taking into account that some nodes may lag behind temporarily when missing reception opportunities of a flooding frame [166], as discussed in Section 7.1.3. Five receivers with an Atheros AR928X wireless network adapter are distributed at distances of one to five hops around the source node [166]. With 1380-byte MSDUs at Non-HT MCS 5 and with an `rtx_rep_per_node` value of 5, the FRRs of the receivers range from 99,85 % to 99,97 %, which is more than sufficient for video playback without noticeable artefacts [166].

7.3.5 *Performance Optimizations*

Flooding process pipelining is a technique that allows successive flooding processes to be simultaneously in transit in a WMN, as described in Section 7.1.4.3 and as applied in the experiments discussed in Section 7.3.4. However, nodes that miss their first reception opportunities of a flooding frame lag behind in the respective flooding process, so that the flooding initiator has to wait a certain number of Tx slots to ensure that successive flooding processes do not interfere with each other, as described in Section 7.1.3. Still, in a WMN with a high node density, a flooding process is persistent since a high fraction of nodes can receive a flooding frame successfully upon the first reception opportunity and forward it further, as discussed in Section 7.1.2 and in Section 7.1.6. Therefore, the flooding process pipelining could be optimized by letting each node attempt to detect missed reception opportunities so that it can update its internal state similarly as upon a successful flooding frame reception but without generating an RTx. A missed reception opportunity can be detected, for instance, through the reception of a frame with a valid MAC header according to the low-level MAC protocol but with a bad frame check sequence (FCS). To ensure that the internal state of each node is consistent when processing such frames, the MAC header would require a separate checksum. Alternatively, a node could also keep track of erroneous receptions and their respective times of occurrence. When the node successfully receives a flooding frame afterwards, it checks if the previous erroneous receptions appear to belong to the flooding process of the currently processed flooding frame so that it can decide whether to generate an RTx. In this way, the medium occupancy of each flooding process can be minimized, allowing the throughput to be increased by reducing waiting times in flooding process pipelining.

The SINR values of CCTs have a wider range than transmissions from a single node due to the randomness of the POs between the transmitters, as discussed in Section 7.1.1. Therefore, successive CCTs that are transmitted by the same nodes may also have different SINR values at a particular receiver. However, scenarios in which a single node initiates several flooding processes in close succession, as discussed in Section 7.3.4, are performed under otherwise largely identical conditions. The signal quality of the CCTs of successive flooding processes can possibly be optimized by letting the concurrently transmitting nodes adjust their phases such that they generate constructive interference at their respective next-hop neighbors. To this end, each node can use each CCT from its neighbors as feedback to adjust its own phase for the generation of the next CCT. However, the feasibility of such a mechanism is the subject of future research.

7.3.6 *Security*

A CCT-based flooding process can temporarily disrupt the communications of other nodes due to its persistent propagation throughout a WMN with a high node density. To prevent any misuse of CCT-based network flooding for jamming in an extensive area, countermeasures are required. The flooding initiator could, for instance, add a timestamp with a cryptographic signature so that the flooding frame is valid only for a short time. To decide quickly whether to retransmit a received flooding frame, the signature check could be executed by specialized hardware in real-time, similarly like the FCS check. However, security aspects are out of scope of this thesis and require further research.

8 CONCLUSION

This thesis proposes decentralized URLLC (dURLLC) as a new communication paradigm that allows for a quick and reliable exchange of data in the region of their origin without relying on a centralized infrastructure. To enable dURLLC applications and scenarios, this thesis explores the practical feasibility of concurrent cooperative transmission (CCT) with orthogonal frequency-division multiplexing (OFDM) as a means to disseminate data efficiently in an extensive area, exceeding the communication range of a single node by up to a few orders of magnitude. In contrast to a medium access control (MAC) mechanism that is based on carrier sense multiple access with collision avoidance (CSMA/CA), CCT allows spectral resources to be utilized more efficiently by leveraging interference instead of trying to avoid collisions. Further, CCT-based network flooding allows waiting times to be minimized through a reception-triggered low-level MAC.

While Glossy [73] showcases the practical feasibility of CCT-based network flooding, it works by means of the IEEE 802.15.4 physical layer (PHY) [107], which is based on direct-sequence spread spectrum (DSSS) technology, providing a data rate of 250 kbit/s. Broadband communications is the key enabler to enhance the reliability, the latency and the throughput of CCT-based network flooding, in comparison to the IEEE 802.15.4 PHY. A larger channel bandwidth essentially enhances the channel capacity, which in turn allows for a higher PHY data rate and for higher reliability when the channel capacity is not fully utilized. A higher PHY data rate, in turn, reduces the transmission duration for a given amount of data, allowing the latency per hop to be reduced.

This thesis explores CCT-based network flooding with the OFDM-based IEEE 802.11 Non-HT and HT PHYs, which are established standards that allow for interactions with commercial devices. The IEEE 802.11 Non-HT PHY is the most basic OFDM-based PHY of the IEEE 802.11 standard family, allowing for the lowest latency per hop with a short MAC service data unit (MSDU) due to its short PHY preamble and PHY header. Nevertheless, the methods and solutions of this thesis are also applicable with more recent and custom OFDM-based PHYs, so that the performance can be increased further through an even larger channel bandwidth or through more compact frame formats.

The analysis of CCT with the OFDM-based IEEE 802.11 Non-HT PHY considers the phase offset (PO), the carrier frequency offset (CFO) and the time offset (TO) between the transmitters, as well as the elapsed time, and draws a distinction between two cases. On the one hand, when the TO is less than 61.5 ns, there is a chance that no symbol errors occur during a CCT, depending on the initial PO, the CFO and the frame duration. The PO, in turn, is random as the frequency oscillators of the transmitters are uncoupled. Further, a CFO changes the phase relation between the CCT signal components over time, so that the interference of a CCT can potentially become destructive on a large portion of OFDM subcarriers, which cannot be recovered through forward error correction (FEC). On the other hand, when the TO ranges between 61.5 ns and the guard interval (GI) duration of 800 ns, symbol errors successively start to occur on certain OFDM subcarriers at the pace of the CFO, which can be corrected through FEC up to a certain amount.

CCT with coherent interference (CI) is the primary approach of this thesis that attempts to keep the phase relation between the CCT signal components as invariant as possible, which can be achieved through precise frequency synchronization of the transmitters. With this, the amount of symbol errors that start to occur after the beginning of a CCT can be reduced to a minimum. This work comprises two prototypes for CCT with CI, one of which is based on the Wireless Open Access Research Platform (WARP) v3 [246] and its IEEE 802.11 reference design [247], whereas the other one is created through firmware modifications of the Asus RT-AC86U wireless router. The WARP prototype achieves precise frequency synchronization through a single frame reception by means of an enhanced CFO estimator that is integrated into the PHY Rx core [9], taking advantage of the common phase error (CPE) of the four pilot subcarriers, and a frequency shifter that is integrated into the PHY Tx core [9]. The Asus RT-AC86U prototype synchronizes in frequency through multiple frame receptions, using a PHY register of the BCM4365E chip that exposes a coarse CFO estimate of each received frame, as well as a radio register that can be used to tune the voltage-controlled oscillator (VCO). Both prototypes generate a reception-triggered retransmission (RTx) of a flooding frame with accurate timing, similar to how they generate an acknowledgment (ACK) frame, respectively.

Both prototypes are employed in testbed experiments in which two groups of nodes generate successive CCTs in a ping-pong fashion, while each CCT triggers the generation of the respective next CCT. These experiments emulate the temporal performance of CCT-based network flooding for a large number of successive CCTs with a few nodes. The nodes consistently keep up the synchronization in time and frequency over time, allowing for experiments with 10 000 successive CCTs for various modulation and coding scheme (MCS) indices and MSDU sizes. Furthermore, commercial devices often have a better frame reception rate (FRR) for CCTs than for transmissions from a single node, which confirms that the CCTs keep up a good signal quality. The measured CFO and interframe space (IFS) values of both prototypes have a consistent quality.

The signal-to-interference-plus-noise ratio (SINR) values of CCTs have a wider range than transmissions from a single node, which is due to the randomness of the POs of the transmitters, so that the interference may be constructive or destructive by chance. While most CCTs typically have a SINR value that is at a similar level as a transmission from a single node, a few CCTs may have a higher SINR and a few CCTs may have a lower SINR at a particular receiver. However, a CCT-based flooding process in a dense wireless multi-hop network (WMN) is highly persistent due to the redundancy of nodes. In fact, the SINR characteristics of CCTs barely depend on the number of transmitters, which makes CCT-based network flooding particularly applicable for dense WMNs. Moreover, the SINR characteristics of CCTs are quite consistent over the RTx index, i.e., successive CCTs have a steady signal quality also after a large number of hops.

The requirement of Ultra-Reliable Low-Latency Communications (URLLC) is defined by the 3rd Generation Partnership Project (3GPP) as the delivery of a 32-byte MSDU with a reliability of $1 - 10^{-5}$ and with a latency of 1 ms [3]. This requirement is assessed with both prototypes in testbed experiments in which 1 000 000 CCTs are generated in a ping-pong fashion between two groups of five nodes each, respectively. The reliability is approximated by the FRR, which is calculated for receptions from each of both groups as well as for receptions from any of both groups, the latter of which emulates two different reception opportunities of a node during a CCT-based flooding process in a WMN.

With CCTs generated by WARP prototype nodes using the IEEE 802.11 Non-HT PHY, the URLLC reliability requirement is fulfilled without additional RTx repetitions with up to MCS 2 under non-line-of-sight (NLOS) conditions and with up to MCS 4 under line-of-sight (LOS) conditions (see Table 2). With CCTs generated by Asus RT-AC86U prototype nodes using the IEEE 802.11 Non-HT PHY, the URLLC reliability requirement is fulfilled without additional RTx repetitions up to Non-HT MCS 6 under LOS conditions and with Non-HT MCS 0 and 2 under NLOS conditions. Additional RTx repetitions allow the URLLC reliability requirement to be met also with other parameter combinations. Using a 32-byte MSDU with the IEEE 802.11 Non-HT PHY, the latency per hop ranges from 120.2 µs at MCS 0 over 60.2 µs at MCS 4 to 48.2 µs at MCS 7.

The Asus RT-AC86U prototype is further used to show that CCT with CI is also feasible with multiple input multiple output (MIMO) spatial streams. Testbed experiments are conducted on a 40 MHz wide channel in the 5 GHz band with the IEEE 802.11 HT PHY. With CCTs of a 32-byte MSDU at a PHY data rate of 324 Mbit/s through MCS 28, i.e., with four spatial streams, a LOS receiver achieves an FRR of 99.5 % with receptions from any of both groups, which is not achieved with less than four spatial streams. Further, with a fixed PHY data rate, the reliability typically increases with the number of spatial streams. For instance, at a PHY data rate of 108 Mbit/s and with receptions from any of both groups, the URLLC reliability requirement can be met without additional RTx repetitions when using four spatial streams, whereas additional RTx repetitions are required with less than four spatial streams.

When frequency synchronization between the transmitters of a CCT cannot be achieved, the interference changes over time, which causes frame loss with conventional receivers. This work proposes the equalization with time-variant zero-forcing (TVZF) for CCTs with incoherent interference and presents a TVZF receiver for the IEEE 802.11 Non-HT PHY. With an absolute TO of up to the duration of the GI and with a CFO within $[-5, 5]$ kHz, the average FRRs of CCTs with 2, 3 and 4 transmitters are 97 %, 78 % and 49 %, respectively. However, these parameter ranges also comprise small absolute CFO values, for which the TVZF receiver underperforms since it may not detect all CCT signal components. With CCTs from three unsyntonized Asus RT-AC86U wireless routers, the TVZF receiver achieves an FRR of up to 92 %. Since TVZF equalization complements the characteristics of conventional receivers, it can be employed as an additional receiver enhancement.

The WARP prototype comprises a low-level MAC protocol that does not only allow for ping-pong transmissions but also for CCT-based network flooding over multiple hops. As the WARP prototype processes the low-level MAC protocol in real-time, the behavior of the WMN nodes can be controlled by the flooding initiator through the flooding frame without additional signaling, allowing certain performance tradeoffs to be tuned to support dURLLC applications and scenarios. For instance, the WARP prototype can facilitate multi-hop video streaming, while commercial devices at distances of up to five hops from the video source can achieve FRRs from 99,85 % to 99,97 % [166].

A high-level MAC protocol may leverage the global time synchronization feature of CCT-based network flooding for medium reservations during specific time intervals. Further, CCT-based routing may draw on the redundancy of nodes to facilitate reliable communications also over large distances. CCT with a reservation-based high-level MAC in combination with a reception-triggered low-level MAC is the enabler for seamless interactions between entities in cyber-physical systems under the dURLLC paradigm.

Part V

APPENDIX

A

EVALUATION OF WARP PROTOTYPE: 1 VS. 5 TRANSMITTERS

This appendix lists the evaluation results of the experiment described in Section 4.5.1.

Group ID	Device ID	MCS 0	MCS 1	MCS 2	MCS 3	MCS 4	MCS 5	MCS 6	MCS 7
0	W3-a-00134	99.8	99.6	99.7	99.8	99.6	99.6	99.7	99.9
1	W3-a-00259	99.8	99.7	99.8	99.9	99.8	99.9	99.6	100

Table 12: FRR (%) of the WARP prototype nodes in 1x1 ping-pong experiment

Device ID	MCS 0			MCS 1			MCS 2			MCS 3		
	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any
Matlab	100	100	100	99.8	100	100	100	99.9	100	99.9	100	100
Acer-01	94.9	94.8	100	94.0	92.9	99.9	97.1	97.4	99.9	95.7	94.7	99.8
Acer-03	96.1	86.5	99.1	97.9	90.3	99.6	94.9	85.8	97.1	92.5	83.5	96.4
Acer-06	93.4	92.2	99.7	90.0	90.9	99.3	93.1	91.8	99.2	87.9	93.2	97.9
Acer-07	78.0	80.5	93.4	82.6	86.0	96.7	94.7	94.7	98.8	73.5	72.4	88.6
EeePC-01	97.1	96.5	99.9	97.2	96.5	99.9	96.2	96.8	99.2	96.4	96.6	98.8
EeePC-02	97.1	90.0	99.2	97.5	50.8	98.5	96.0	88.8	97.9	97.0	19.7	97.5
EeePC-08	95.6	96.2	99.1	96.5	93.3	99.4	96.3	96.5	99.0	96.6	96.4	98.8
EeePC-12	88.0	78.7	95.9	80.4	76.5	95.9	79.3	78.6	91.8	78.2	78.3	91.4

Device ID	MCS 4			MCS 5			MCS 6			MCS 7		
	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any
Matlab	99.9	100	100	100	100	100	99.9	100	100	100	100	100
Acer-01	87.9	87.9	96.0	93.9	93.0	98.5	88.5	88.6	95.7	90.4	89.8	98.2
Acer-03	98.7	96.5	99.5	99.1	98.4	99.7	93.6	87.4	96.6	97.5	94.4	98.9
Acer-06	79.5	79.9	93.7	85.9	83.3	95.6	76.6	75.9	90.3	69.8	85.4	94.2
Acer-07	82.0	78.9	92.8	80.0	78.8	89.7	81.1	79.4	91.8	75.2	60.7	86.5
EeePC-01	95.9	95.9	99.1	95.4	95.5	98.7	94.1	94.5	98.1	94.7	77.5	97.9
EeePC-02	93.9	0.7	94.0	95.0	0.0	95.0	97.9	0.2	97.9	97.5	0.0	97.5
EeePC-08	94.2	95.3	97.6	94.6	84.7	97.5	61.0	95.2	97.0	2.3	0.0	2.3
EeePC-12	74.5	71.1	83.7	69.4	64.6	82.3	82.1	79.3	92.0	71.8	70.9	87.4

Table 13: FRR (%) of monitor devices in 1x1 ping-pong experiment, separated by group ID (gID)

	Device ID	MCS 0			MCS 1			MCS 2			MCS 3		
		FRR	rRTx	FTR-G	FRR	rRTx	FTR-G	FRR	rRTx	FTR-G	FRR	rRTx	FTR-G
Group ID: 0	W3-a-00140	92.7	4.8	98.5	89.0	6.8	97.2	93.8	4.4	99.0	87.7	5.8	95.8
	W3-a-00537	89.1	4.3	96.2	88.1	4.8	96.1	91.3	3.9	97.3	83.8	4.5	93.1
	W3-a-00551	91.0	5.1	97.7	90.8	6.0	98.3	90.5	5.4	98.0	89.3	7.4	97.9
	W3-a-00134	92.1	7.1	99.8	91.4	7.7	99.8	93.2	6.2	99.9	91.9	7.2	99.9
	W3-a-00156	88.6	9.7	99.7	86.7	10.9	99.6	88.9	9.5	99.7	87.0	10.9	99.5
Group ID: 1	W3-a-00116	93.5	5.6	99.7	89.6	8.3	99.0	96.0	3.6	99.9	91.4	6.3	98.9
	W3-a-00130	91.5	6.2	99.2	87.6	7.3	98.4	88.2	6.1	98.4	88.4	7.2	97.6
	W3-a-00131	91.7	6.2	98.8	88.6	6.6	97.3	90.4	7.2	98.5	90.3	7.0	98.3
	W3-a-00259	95.7	4.0	100	95.7	4.0	100	95.6	4.1	100	95.7	4.0	100
	W3-a-00516	92.5	6.6	99.8	90.1	8.6	99.7	93.0	6.3	100	92.4	6.7	99.8

	Device ID	MCS 4			MCS 5			MCS 6			MCS 7		
		FRR	rRTx	FTR-G	FRR	rRTx	FTR-G	FRR	rRTx	FTR-G	FRR	rRTx	FTR-G
Group ID: 0	W3-a-00140	86.0	6.2	95.4	66.7	9.6	85.8	62.2	7.7	77.6	55.7	8.9	75.7
	W3-a-00537	65.9	4.7	84.2	71.7	7.0	89.7	58.9	6.3	69.2	61.6	7.9	72.9
	W3-a-00551	86.2	7.0	96.0	62.7	10.4	83.5	57.3	10.9	74.4	44.2	10.3	63.4
	W3-a-00134	88.3	9.8	99.3	80.0	14.3	98.0	67.0	18.2	92.3	59.0	20.5	91.9
	W3-a-00156	83.2	13.1	99.2	66.4	18.0	95.4	40.9	14.5	73.2	43.5	18.2	84.2
Group ID: 1	W3-a-00116	89.8	6.9	98.7	78.7	9.4	95.2	76.8	7.9	91.7	72.3	9.4	92.1
	W3-a-00130	89.7	7.0	98.8	79.3	7.3	94.6	80.3	7.1	93.0	66.6	7.0	82.9
	W3-a-00131	86.8	7.6	96.6	81.0	8.3	93.0	69.7	11.3	85.1	63.1	10.9	78.8
	W3-a-00259	94.7	5.0	100	85.9	9.9	99.9	87.6	8.1	99.7	77.9	13.5	99.1
	W3-a-00516	88.6	9.1	99.6	78.4	13.8	97.5	68.9	15.7	93.7	62.8	15.2	88.5

Table 14: Performance metrics (%) of the WARP prototype nodes in 5x5 ping-pong experiment: Section 4.2.5 gives the definitions of the FRR and the FTR-G metrics. As the experiment is conducted with two groups of nodes, the FTR is identical to the FRR and is not listed. The rRTx metric is defined as the fraction of retryRTx frames of a node over the total number of RTx frames of its group. Thus, the fraction of CCTs to which a particular node contributed by generating an RTx or a retryRTx is the sum of its FRR and its rRTx.

Device ID	MCS 0			MCS 1			MCS 2			MCS 3		
	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any
Matlab	100	99.9	100	100	99.9	100	100	100	100	100	99.7	100
Acer-01	95.0	94.9	99.9	92.1	93.1	99.8	97.1	97.6	99.8	95.7	95.9	99.6
Acer-03	100	100	100	99.4	99.7	99.9	99.9	99.9	100	100	99.8	100
Acer-06	97.1	96.7	100	94.2	94.1	99.7	95.5	94.9	99.6	96.1	96.4	98.8
Acer-07	93.5	94.8	98.6	95.1	95.5	99.7	97.4	97.8	99.6	97.0	97.6	99.5
EeePC-01	97.5	97.4	99.8	96.3	96.9	99.8	96.4	95.7	98.8	96.1	96.8	99.0
EeePC-02	98.4	98.2	99.8	97.6	97.6	99.8	94.5	95.2	98.7	95.9	96.8	99.2
EeePC-08	97.7	97.9	99.9	97.5	98.0	99.9	97.3	97.4	99.4	97.3	97.6	99.2
EeePC-12	99.1	99.6	99.9	94.9	96.3	99.1	98.0	98.8	99.6	85.7	84.0	94.9

Device ID	MCS 4			MCS 5			MCS 6			MCS 7		
	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any	gID 0	gID 1	any
Matlab	99.8	99.8	100	95.8	98.1	99.5	94.9	94.9	99.7	90.3	90.3	98.9
Acer-01	84.1	85.0	94.6	90.9	92.0	97.2	59.4	59.7	74.5	82.7	80.4	95.0
Acer-03	99.5	99.6	99.9	98.8	97.3	99.9	99.3	98.4	100	98.0	97.0	99.9
Acer-06	78.3	78.6	92.0	86.1	85.8	94.9	79.3	81.0	90.9	83.7	85.5	95.2
Acer-07	84.3	85.6	93.0	89.5	90.3	96.5	86.0	85.3	95.0	85.0	79.0	95.4
EeePC-01	96.0	96.3	98.9	93.0	92.0	97.0	93.0	92.0	98.2	91.3	93.6	98.4
EeePC-02	97.7	98.1	99.3	87.9	90.1	96.4	88.1	87.7	95.5	88.1	93.0	97.8
EeePC-08	96.7	97.2	99.3	92.7	95.1	97.8	94.8	95.7	98.9	91.9	91.4	98.1
EeePC-12	96.8	95.6	98.2	96.6	97.3	99.6	97.3	95.5	99.6	79.6	72.1	89.0

Table 15: FRR (%) of monitor devices in 5x5 ping-pong experiment, separated by group ID (gID)

B

EVALUATION OF WARP PROTOTYPE: URLLC REQUIREMENT

This appendix lists the evaluation results of the experiment described in Section 4.5.2.

Device ID	MCS 0			MCS 1		
	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	99.9842	99.9900	99.9998	99.9442	99.9790	99.9994
Matlab NLOS	99.9796	99.9942	100	99.9364	99.9654	99.9998
Device ID	MCS 2			MCS 3		
	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	99.9826	99.9940	100	99.9318	99.9436	99.9998
Matlab NLOS	99.9718	99.9894	99.9998	99.6818	99.9184	99.9958
Device ID	MCS 4			MCS 5		
	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	99.9688	99.9718	99.9996	99.7342	99.7378	99.9958
Matlab NLOS	99.3444	99.9068	99.9884	96.5686	98.1888	99.8856
Device ID	MCS 6			MCS 7		
	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	99.7716	99.3658	99.9868	99.5356	98.2762	99.9898
Matlab NLOS	62.6560	91.8522	96.9834	46.0936	80.9208	90.9330

Table 16: FRR (%) of 1 000 000 CCTs per MCS in 5x5 ping-pong experiment with a 32-byte MSDU, separated by group ID (gID), used to check the URLLC reliability requirement

C

EVALUATION OF WARP PROTOTYPE: CIRCULAR FORWARDING

This appendix lists the evaluation results of the experiment described in Section 4.5.5.

	Device ID	MCS 0			MCS 1			MCS 2			MCS 3		
		FTR	rRTx	FTR-G	FTR	rRTx	FTR-G	FTR	rRTx	FTR-G	FTR	rRTx	FTR-G
gID: 0	W3-a-00537	94.4	1.2	98.6	94.8	0.6	98.4	92.9	1.4	98.6	90.2	1.2	98.2
	W3-a-00551	94.8	1.0	98.4	96.8	0.5	98.0	96.7	0.9	98.6	95.8	0.9	98.0
	W3-a-00140	94.1	0.8	98.6	94.2	0.4	98.1	88.2	1.0	98.6	95.2	1.0	98.4
gID: 1	W3-a-00134	85.6	0.4	97.6	93.0	0.8	97.9	82.4	3.6	97.4	86.2	2.3	95.5
	W3-a-00156	94.0	0.3	97.8	94.0	0.2	97.7	89.8	0.3	97.7	89.4	0.3	97.4
	W3-a-00116	95.7	0.4	97.8	95.5	0.3	97.8	94.2	0.9	97.6	93.1	0.6	97.4
gID: 2	W3-a-00579	75.6	7.2	98.8	67.8	13.1	99.0	48.8	18.3	98.4	12.7	10.3	91.2
	W3-a-00130	93.7	0.6	98.2	95.2	0.7	98.4	94.0	1.0	98.4	93.8	1.5	98.0
	W3-a-00536	92.0	1.2	98.5	92.8	1.5	98.8	84.9	2.4	93.9	85.6	3.6	95.1
gID: 3	W3-a-00131	91.9	0.6	96.1	92.3	1.0	96.7	84.9	0.6	94.6	84.2	0.5	95.3
	W3-a-00516	91.1	0.9	98.2	94.2	0.9	98.4	87.9	0.8	97.2	83.2	1.5	97.1
	W3-a-00259	90.0	0.2	95.2	90.6	0.5	95.7	89.4	0.4	95.0	92.4	0.2	94.5

Table 17: Performance metrics (%) of 4 groups of WARP prototype nodes in an experiment with circular forwarding: Section 4.2.5 gives the definitions of the FTR and the FTR-G metrics. Note that the FTR is identical to the FRR of the frames from the previous hop of a group. The rRTx metric is defined as the fraction of retryRTx frames of a node over the total number of RTx frames of its group. The fraction of CCTs of a group to which a specific node contributed by generating an RTx or a retryRTx is the sum of its FTR and its rRTx.

Device ID	MCS 0					MCS 1				
	gID 0	gID 1	gID 2	gID 3	any	gID 0	gID 1	gID 2	gID 3	any
Matlab	98.6	97.9	0.0	0.0	100	98.3	97.5	0.0	0.0	100
Acer-01	93.8	0.0	91.2	90.8	99.9	91.0	0.0	86.8	92.4	99.8
Acer-03	0.0	97.1	97.7	94.2	99.1	0.0	94.4	97.7	94.5	99.1
Acer-06	95.4	96.9	91.9	0.0	99.2	92.1	94.7	90.5	0.0	99.4
Acer-07	91.6	96.3	0.0	97.8	100	90.9	94.8	0.0	94.8	100
EeePC-01	92.0	0.0	91.3	94.4	99.8	87.6	0.0	92.1	94.2	99.7
EeePC-02	0.0	92.4	89.0	13.4	96.1	0.0	93.6	88.6	0.2	96.9
EeePC-08	93.5	94.5	76.4	0.0	98.8	90.7	94.7	55.8	0.0	99.0
EeePC-12	97.5	94.2	0.0	95.9	100	97.6	94.6	0.0	86.4	99.7

Device ID	MCS 2					MCS 3				
	gID 0	gID 1	gID 2	gID 3	any	gID 0	gID 1	gID 2	gID 3	any
Matlab	96.3	96.7	0.0	0.0	99.8	97.4	100	0.0	0.0	100
Acer-01	80.4	0.0	83.1	92.1	99.7	82.1	0.0	89.1	89.0	99.6
Acer-03	0.0	94.8	93.6	83.8	99.0	0.0	96.2	92.4	91.7	99.3
Acer-06	87.0	94.0	66.2	0.0	98.9	87.4	91.5	92.3	0.0	98.7
Acer-07	91.4	92.0	0.0	94.8	100	89.5	89.0	0.0	95.6	100
EeePC-01	62.6	0.0	82.5	93.2	99.1	44.7	0.0	75.3	92.8	98.7
EeePC-02	0.0	89.1	82.4	0.0	95.2	0.0	84.2	83.1	0.0	94.6
EeePC-08	71.5	94.6	24.7	0.0	97.9	59.6	94.1	0.2	0.0	96.5
EeePC-12	97.4	86.5	0.0	39.4	98.6	96.9	86.5	0.0	37.7	98.2

Table 18: FRR (%) of monitor devices in an experiment with circular forwarding by 4 groups of WARP prototype nodes, separated by group ID (gID)

D

EVALUATION OF ASUS RT-AC86U PROTOTYPE: URLLC REQUIREMENT

This appendix lists the evaluation results of the experiment described in Section 5.4.1.

Device ID	MCS 0			MCS 1		
	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	99.9920	99.9968	100	99.8622	99.9942	100
Matlab NLOS	99.8428	99.9854	100	99.5130	99.2636	99.9976
Device ID	**MCS 2**			**MCS 3**		
	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	99.9994	99.9994	100	99.9996	99.9638	100
Matlab NLOS	99.9578	99.9576	99.9998	99.5668	98.8342	99.9954
Device ID	**MCS 4**			**MCS 5**		
	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	99.9948	99.9958	100	99.9930	99.9646	100
Matlab NLOS	98.3256	95.6282	99.9310	87.2150	72.8368	96.5120
Device ID	**MCS 6**			**MCS 7**		
	gID 0	gID 1	any	gID 0	gID 1	any
Matlab LOS	99.9698	99.7990	100	99.3154	97.4508	99.9804
Matlab NLOS	43.3970	28.5346	59.5114	25.3912	1.7816	26.5922

Table 19: FRR (%) of 1 000 000 CCTs per MCS in 5x5 ping-pong experiment with a 32-byte MSDU, separated by group ID (gID), used to check the URLLC reliability requirement

E

EVALUATION OF ASUS RT-AC86U PROTOTYPE: MIMO SPATIAL STREAMS

This appendix lists the evaluation results of the experiments described in Section 5.4.4.

Modulation	Coding rate	$N_{SS} = 1$	$N_{SS} = 2$	$N_{SS} = 3$	$N_{SS} = 4$
BPSK	1/2	MCS 0 13.5 Mbit/s	MCS 8 27.0 Mbit/s	MCS 16 40.5 Mbit/s	MCS 24 54.0 Mbit/s
QPSK	1/2	MCS 1 27.0 Mbit/s	MCS 9 54.0 Mbit/s	MCS 17 81.0 Mbit/s	MCS 25 108.0 Mbit/s
QPSK	3/4	MCS 2 40.5 Mbit/s	MCS 10 81.0 Mbit/s	MCS 18 121.5 Mbit/s	MCS 26 162.0 Mbit/s
16-QAM	1/2	MCS 3 54.0 Mbit/s	MCS 11 108.0 Mbit/s	MCS 19 162.0 Mbit/s	MCS 27 216.0 Mbit/s
16-QAM	3/4	MCS 4 81.0 Mbit/s	MCS 12 162.0 Mbit/s	MCS 20 243.0 Mbit/s	MCS 28 324.0 Mbit/s
64-QAM	2/3	MCS 5 108.0 Mbit/s	MCS 13 216.0 Mbit/s	MCS 21 324.0 Mbit/s	MCS 29 432.0 Mbit/s
64-QAM	3/4	MCS 6 121.5 Mbit/s	MCS 14 243.0 Mbit/s	MCS 22 364.5 Mbit/s	MCS 30 486.0 Mbit/s
64-QAM	5/6	MCS 7 135.0 Mbit/s	MCS 15 270.0 Mbit/s	MCS 23 405.0 Mbit/s	MCS 31 540.0 Mbit/s

Table 20: MCS parameters of the IEEE 802.11 HT PHY for 40 MHz channel spacing, 800 ns GI and all numbers of spatial streams (N_{SS}) [104, Section 19.5, Tables 19-31, 19-32, 19-33, 19-34]

MSDU: 32 bytes								
$N_{SS} = 1$	MCS 0	MCS 1	MCS 2	MCS 3	MCS 4	MCS 5	MCS 6	MCS 7
Group ID 0	100	99.70	99.48	98.86	96.46	96.60	96.52	94.80
Group ID 1	100	99.82	99.40	98.74	97.00	96.58	97.18	94.80
Any group	100	100	100	100	99.90	99.90	99.96	99.74
$N_{SS} = 2$	MCS 8	MCS 9	MCS 10	MCS 11	MCS 12	MCS 13	MCS 14	MCS 15
Group ID 0	99.96	99.42	99.02	98.72	97.52	94.06	88.76	69.54
Group ID 1	99.98	99.72	99.32	98.84	96.34	71.02	61.88	27.16
Any group	100	100	100	99.96	99.78	98.26	95.60	77.40
$N_{SS} = 3$	MCS 16	MCS 17	MCS 18	MCS 19	MCS 20	MCS 21	MCS 22	MCS 23
Group ID 0	100	99.86	99.72	99.58	98.46	82.48	62.42	33.28
Group ID 1	99.98	99.90	99.76	99.70	97.24	81.94	60.10	23.44
Any group	100	100	100	100	99.88	97.02	84.72	48.88
$N_{SS} = 4$	MCS 24	MCS 25	MCS 26	MCS 27	MCS 28	MCS 29	MCS 30	MCS 31
Group ID 0	99.94	99.82	99.54	99.72	94.38	39.68	22.64	1.48
Group ID 1	99.90	99.82	99.14	98.68	88.56	26.26	21.26	2.22
Any group	100	100	100	100	99.54	56.04	39.48	3.66

Table 21: FRR (%) of 10 000 CCTs through 100 runs of 100 CCTs each, with an MSDU of 32 bytes, a 40 MHz wide channel and a GI of 800 ns in the HT-MF PHY format (see Table 20)

MSDU: 128 bytes								
$N_{SS} = 1$	MCS 0	MCS 1	MCS 2	MCS 3	MCS 4	MCS 5	MCS 6	MCS 7
Group ID 0	100	100	100	99.80	99.12	98.42	98.68	98.48
Group ID 1	100	100	100	99.84	99.46	98.38	98.84	98.46
Any group	100	100	100	100	100	99.96	100	99.96
$N_{SS} = 2$	MCS 8	MCS 9	MCS 10	MCS 11	MCS 12	MCS 13	MCS 14	MCS 15
Group ID 0	100	99.98	99.68	99.32	97.70	89.06	85.82	43.98
Group ID 1	100	100	99.88	99.56	93.58	64.78	51.84	19.80
Any group	100	100	100	99.98	99.84	96.18	93.22	55.18
$N_{SS} = 3$	MCS 16	MCS 17	MCS 18	MCS 19	MCS 20	MCS 21	MCS 22	MCS 23
Group ID 0	99.98	99.96	99.32	99.80	95.44	82.98	54.16	27.54
Group ID 1	100	100	97.76	99.98	92.58	71.50	49.48	22.02
Any group	100	100	99.96	100	99.72	95.08	76.04	43.32
$N_{SS} = 4$	MCS 24	MCS 25	MCS 26	MCS 27	MCS 28	MCS 29	MCS 30	MCS 31
Group ID 0	100	99.98	99.70	99.52	79.28	28.62	5.02	0.08
Group ID 1	100	100	99.18	99.26	73.64	22.24	7.00	2.82
Any group	100	100	99.98	100	95.18	44.54	11.68	2.88

Table 22: FRR (%) of 10 000 CCTs through 100 runs of 100 CCTs each, with an MSDU of 128 bytes, a 40 MHz wide channel and a GI of 800 ns in the HT-MF PHY format (see Table 20)

MSDU: 512 bytes								
$N_{SS} = 1$	MCS 0	MCS 1	MCS 2	MCS 3	MCS 4	MCS 5	MCS 6	MCS 7
Group ID 0	93.82	99.54	98.22	99.74	98.18	98.02	95.60	91.60
Group ID 1	98.76	99.90	99.74	100	99.78	99.88	98.68	95.30
Any group	99.82	100	100	100	100	99.98	99.94	99.70
$N_{SS} = 2$	MCS 8	MCS 9	MCS 10	MCS 11	MCS 12	MCS 13	MCS 14	MCS 15
Group ID 0	96.06	99.38	94.74	98.74	87.74	53.08	39.38	15.14
Group ID 1	99.60	99.96	99.34	99.22	87.10	34.66	18.80	5.60
Any group	100	100	99.98	99.96	98.64	69.64	51.04	19.96
$N_{SS} = 3$	MCS 16	MCS 17	MCS 18	MCS 19	MCS 20	MCS 21	MCS 22	MCS 23
Group ID 0	99.70	99.84	96.72	99.68	78.52	35.34	18.64	7.48
Group ID 1	99.98	99.92	95.66	99.88	77.76	39.44	12.56	4.22
Any group	100	100	99.86	100	95.66	60.80	28.88	11.30
$N_{SS} = 4$	MCS 24	MCS 25	MCS 26	MCS 27	MCS 28	MCS 29	MCS 30	MCS 31
Group ID 0	99.78	99.94	97.76	98.42	47.00	4.56	N/A	N/A
Group ID 1	99.94	100	98.18	98.72	44.78	11.66	N/A	N/A
Any group	100	100	100	99.96	70.90	15.72	N/A	N/A

Table 23: FRR (%) of 10 000 CCTs through 100 runs of 100 CCTs each, with an MSDU of 512 bytes, a 40 MHz wide channel and a GI of 800 ns in the HT-MF PHY format (see Table 20)

	PHY data rate: 54 Mbit/s			PHY data rate: 108 Mbit/s		
	MCS 3 ($N_{SS} = 1$)	MCS 9 ($N_{SS} = 2$)	MCS 24 ($N_{SS} = 4$)	MCS 5 ($N_{SS} = 1$)	MCS 11 ($N_{SS} = 2$)	MCS 25 ($N_{SS} = 4$)
Group ID 0	98.7022	99.5932	99.9406	96.9796	98.9006	99.7868
Group ID 1	99.1318	99.6394	99.9352	97.7374	99.1626	99.7820
Any group	99.9892	99.9990	100	99.9260	99.9914	99.9992

Table 24: FRR (%) for $N_{SS} \in \{1, 2, 4\}$ spatial streams at PHY data rates of 54 Mbit/s and 108 Mbit/s, over 1 000 000 CCTs through 100 runs of 10 000 CCTs each, with an MSDU of 32 bytes, a 40 MHz wide channel and a GI of 800 ns in the HT-MF PHY format (see Table 20)

BIBLIOGRAPHY

[1] *5G; NR; Base Station (BS) Radio Transmission and Reception (3GPP TS 38.104 Version 17.5.0 Release 17)*. Technical Specification. 3GPP, Apr. 2022.

[2] *5G; NR; User Equipment (UE) Radio Transmission and Reception; Part 1: Range 1 Standalone (3GPP TS 38.101-1 Version 17.5.0 Release 17)*. Technical Specification. 3GPP, May 2022.

[3] *5G; Study on Scenarios and Requirements for Next Generation Access Technologies (3GPP TR 38.913 Version 17.0.0 Release 17)*. Technical Report. 3GPP, May 2022.

[4] Omid Abari, Hariharan Rahul, Dina Katabi, and Mondira Pant. "AirShare: Distributed Coherent Transmission Made Seamless." In: *34th IEEE International Conference on Computer Communications (INFOCOM)*. Hong Kong, China: IEEE, Apr. 2015, pp. 1742–1750. DOI: 10.1109/INFOCOM.2015.7218555.

[5] Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. "Mobile Edge Computing: A Survey." In: *IEEE Internet of Things Journal* 5.1 (Feb. 2018), pp. 450–465. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2750180.

[6] Norman Abramson. "The Aloha System: Another Alternative for Computer Communications." In: *AFIPS Fall Joint Computer Conference (FJCC)*. Houston, Texas, USA: ACM, Nov. 1970, pp. 281–285. DOI: 10.1145/1478462.1478502.

[7] Assad Alam, Bart Besselink, Valerio Turri, Jonas Mårtensson, and Karl H. Johansson. "Heavy-Duty Vehicle Platooning for Sustainable Freight Transportation: A Cooperative Method to Enhance Safety and Efficiency." In: *IEEE Control Systems Magazine* 35.6 (Dec. 2015), pp. 34–56. ISSN: 1941-000X. DOI: 10.1109/MCS.2015.2471046.

[8] Bastian Alt, Markus Weckesser, Christian Becker, Matthias Hollick, Sounak Kar, Anja Klein, Robin Klose, Roland Kluge, Heinz Koeppl, Boris Koldehofe, Wasiur R. Khudabukhsh, Manisha Luthra, Mahdi Mousavi, Max Mühlhäuser, Martin Pfannemüller, Amr Rizk, Andy Schürr, and Ralf Steinmetz. "Transitions: A Protocol-Independent View of the Future Internet." In: *Proceedings of the IEEE* 107.4 (Apr. 2019), pp. 835–846. ISSN: 1558-2256. DOI: 10.1109/JPROC.2019.2895964.

[9] Konrad Altenhofen. "Fluten mit geringer Latenz in IEEE-802.11g-Multi-Hop-Netzen durch gleichzeitige Übertragungen mit drahtloser Synchronisierung mit WARP Software-Defined Radios." BSc thesis. Technische Universität Darmstadt, Sept. 2020.

[10] Mohammad A. Amir Khojastepour, Karthikeyan Sundaresan, Sampath Rangarajan, and Mohammad Farajzadeh-Tehrani. "Scaling Wireless Full-Duplex in Multi-Cell Networks." In: *34th IEEE International Conference on Computer Communications (INFOCOM)*. Hong Kong, China: IEEE, Apr. 2015, pp. 1751–1759. DOI: 10.1109/INFOCOM.2015.7218556.

[11] Jeffrey G. Andrews. "Interference Cancellation for Cellular Systems: A Contemporary Overview." In: *IEEE Wireless Communications* 12.2 (Apr. 2005), pp. 19–29. ISSN: 1558-0687. DOI: 10.1109/MWC.2005.1421925.

[12] Alejandro Aragon-Zavala. "Indoor Wireless Technologies." In: *Indoor Wireless Communications: From Theory to Implementation*. Hoboken, New Jersey: John Wiley & Sons, Aug. 2017, pp. 7–44. ISBN: 9781119004547. DOI: 10.1002/9781119004547.ch2.

[13] Jean Armstrong. "Analysis of New and Existing Methods of Reducing Intercarrier Interference due to Carrier Frequency Offset in OFDM." In: *IEEE Transactions on Communications* 47.3 (Mar. 1999), pp. 365–369. ISSN: 0090-6778. DOI: 10.1109/26.752816.

[14] Sangkyu Baek, Donggun Kim, Milos Tesanovic, and Anil Agiwal. "3GPP New Radio Release 16: Evolution of 5G for Industrial Internet of Things." In: *IEEE Communications Magazine* 59.1 (Jan. 2021), pp. 41–47. ISSN: 1558-1896. DOI: 10.1109/MCOM.001.2000526.

[15] Horia Vlad Balan, Ryan Rogalin, Antonios Michaloliakos, Konstantinos Psounis, and Giuseppe Caire. "AirSync: Enabling Distributed Multiuser MIMO With Full Spatial Multiplexing." In: *IEEE/ACM Transactions on Networking* 21.6 (Dec. 2013), pp. 1681–1695. ISSN: 1063-6692. DOI: 10.1109/TNET.2012.2230449.

[16] Sanaz Barghi and Hamid Jafarkhani. "Exploiting Asynchronous Amplify-and-Forward Relays to Enhance the Performance of IEEE 802.11 Networks." In: *IEEE/ACM Transactions on Networking* 23.2 (Apr. 2015), pp. 479–490. ISSN: 1558-2566. DOI: 10.1109/TNET.2014.2300147.

[17] Gwen Barriac, Raghuraman Mudumbai, and Upamanyu Madhow. "Distributed Beamforming for Information Transfer in Sensor Networks." In: *3rd IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN)*. Berkeley, California, USA: IEEE, Apr. 2004, pp. 81–88. DOI: 10.1109/IPSN.2004.238786.

[18] Oscar Bejarano and Edward W. Knightly. "Virtual MISO Triggers in Wi-Fi-Like Networks." In: *32nd IEEE International Conference on Computer Communications (INFOCOM)*. Turin, Italy: IEEE, Apr. 2013, pp. 1582–1590. DOI: 10.1109/INFCOM.2013.6566954.

[19] Jacob Benesty, Israel Cohen, and Jingdong Chen. *Array Beamforming with Linear Difference Equations*. Springer Topics in Signal Processing. Cham, Switzerland: Springer, 2021. ISBN: 9783030682729.

[20] Nevio Benvenuto, Stefano Tomasin, and Daniele Veronesi. "Multiple Frequency Offsets Estimation and Compensation for Cooperative Networks." In: *IEEE Wireless Communications and Networking Conference (WCNC)*. Hong Kong, China: IEEE, Mar. 2007, pp. 891–895. DOI: 10.1109/WCNC.2007.169.

[21] Hugo Bergeron, Laura C. Sinclair, William C. Swann, Isaac Khader, Kevin C. Cossel, Michael Cermak, Jean-Daniel Deschênes, and Nathan R. Newbury. "Femtosecond Time Synchronization of Optical Clocks off of a Flying Quadcopter." In: *Nature Communications* 10.1 (Apr. 2019), pp. 1–7. DOI: 10.1038/s41467-019-09768-9.

[22] Dinesh Bharadia and Sachin Katti. "FastForward: Fast and Constructive Full Duplex Relays." In: *ACM Special Interest Group on Data Communication (SIGCOMM) Conference*. Chicago, Illinois, USA: ACM, Aug. 2014, pp. 199–210. DOI: 10.1145/2619239.2626327.

[23] Dinesh Bharadia and Sachin Katti. "Full Duplex MIMO Radios." In: *11th USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*. Seattle, Washington, USA: USENIX Association, Apr. 2014, pp. 359–372. URL: http://dl.acm.org/citation.cfm?id=2616448.2616482.

[24] Dinesh Bharadia, Emily McMilin, and Sachin Katti. "Full Duplex Radios." In: *ACM Special Interest Group on Data Communication (SIGCOMM) Conference*. Hong Kong, China: ACM, Aug. 2013, pp. 375–386. DOI: 10.1145/2486001.2486033.

[25] Suman Bhunia, Vahid Behzadan, Paulo Alexandre Regis, and Shamik Sengupta. "Adaptive Beam Nulling in Multihop Ad Hoc Networks Against a Jammer in Motion." In: *Computer Networks* 109 (Nov. 2016). Special issue on Recent Advances in Physical-Layer Security, pp. 50–66. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2016.06.030.

[26] Yuanguo Bi, Hangguan Shan, Xuemin Sherman Shen, Ning Wang, and Hai Zhao. "A Multi-Hop Broadcast Protocol for Emergency Message Dissemination in Urban Vehicular Ad Hoc Networks." In: *IEEE Transactions on Intelligent Transportation Systems* 17.3 (Mar. 2016), pp. 736–750. ISSN: 1558-0016. DOI: 10.1109/TITS.2015.2481486.

[27] Giuseppe Bianchi. "Performance Analysis of the IEEE 802.11 Distributed Coordination Function." In: *IEEE Journal on Selected Areas in Communications* 18.3 (Mar. 2000), pp. 535–547. ISSN: 0733-8716. DOI: 10.1109/49.840210.

[28] Nabhendra Bisnik and Alhussein A. Abouzeid. "Queuing Network Models for Delay Analysis of Multihop Wireless Ad Hoc Networks." In: *Ad Hoc Networks* 7.1 (Jan. 2009), pp. 79–97. ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2007.12.001.

[29] Emil Björnson, Jakob Hoydis, Marios Kountouris, and Mérouane Debbah. "Massive MIMO Systems With Non-Ideal Hardware: Energy Efficiency, Estimation, and Capacity Limits." In: *IEEE Transactions on Information Theory* 60.11 (Nov. 2014), pp. 7112–7139. ISSN: 1557-9654. DOI: 10.1109/TIT.2014.2354403.

[30] Emil Björnson, Erik G. Larsson, and Thomas L. Marzetta. "Massive MIMO: Ten Myths and One Critical Question." In: *IEEE Communications Magazine* 54.2 (Feb. 2016), pp. 114–123. ISSN: 1558-1896. DOI: 10.1109/MCOM.2016.7402270.

[31] Emil Björnson, Luca Sanguinetti, Jakob Hoydis, and Mérouane Debbah. "Optimal Design of Energy-Efficient Multi-User MIMO Systems: Is Massive MIMO the Answer?" In: *IEEE Transactions on Wireless Communications* 14.6 (June 2015), pp. 3059–3075. ISSN: 1558-2248. DOI: 10.1109/TWC.2015.2400437.

[32] Helmut Bölcskei, David Gesbert, and Arogyaswami J. Paulraj. "On the Capacity of OFDM-Based Spatial Multiplexing Systems." In: *IEEE Transactions on Communications* 50.2 (Feb. 2002), pp. 225–234. ISSN: 1558-0857. DOI: 10.1109/26.983319.

[33] D. Richard Brown and Radu David. "Receiver-Coordinated Distributed Transmit Nullforming with Local and Unified Tracking." In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Florence, Italy: IEEE, May 2014, pp. 1160–1164. DOI: 10.1109/ICASSP.2014.6853779.

[34] D. Richard Brown, Upamanyu Madhow, Patrick Bidigare, and Soura Dasgupta. "Receiver-Coordinated Distributed Transmit Nullforming with Channel State Uncertainty." In: *46th Annual Conference on Information Sciences and Systems (CISS)*. Princeton, New Jersey, USA: IEEE, Mar. 2012, pp. 1–6. DOI: 10.1109/CISS.2012.6310822.

[35] D. Richard Brown, Gregary B. Prince, and John A. McNeill. "A Method for Carrier Frequency and Phase Synchronization of two Autonomous Cooperative Transmitters." In: *6th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. New York City, New York, USA: IEEE, June 2005, pp. 260–264. DOI: 10.1109/SPAWC.2005.1505912.

[36] D. Richard Brown III and H. Vincent Poor. "Time-Slotted Round-Trip Carrier Synchronization for Distributed Beamforming." In: *IEEE Transactions on Signal Processing* 56.11 (Nov. 2008), pp. 5630–5643. ISSN: 1941-0476. DOI: 10.1109/TSP.2008.927073.

[37] Maxim Buevich, Niranjini Rajagopal, and Anthony Rowe. "Hardware Assisted Clock Synchronization for Real-Time Sensor Networks." In: *34th IEEE Real-Time Systems Symposium (RTSS)*. Vancouver, British Columbia, Canada: IEEE, Dec. 2013, pp. 268–277. DOI: 10.1109/RTSS.2013.34.

[38] James V. Candy. *Model-Based Signal Processing*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Hoboken, New Jersey: John Wiley & Sons, Oct. 2005. ISBN: 9780471732679. DOI: 10.1002/0471732672.

[39] Eduardo Castañeda, Adão Silva, Atílio Gameiro, and Marios Kountouris. "An Overview on Resource Allocation Techniques for Multi-User MIMO Systems." In: *IEEE Communications Surveys & Tutorials* 19.1 (Firstquarter 2017), pp. 239–284. ISSN: 1553-877X. DOI: 10.1109/COMST.2016.2618870.

[40] *CC2420 – 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. Revision 1.4. Chipcon Products from Texas Instruments. 2006.

[41] Ian D. Chakeres and Elizabeth M. Belding-Royer. "AODV Routing Protocol Implementation Design." In: *24th International Conference on Distributed Computing Systems Workshops (ICDCS Workshops)*. Tokyo, Japan: IEEE, Mar. 2004, pp. 698–703. DOI: 10.1109/ICDCSW.2004.1284108.

[42] Sajith Mohan Chakkedath and Mary Ann Weitnauer. "Cooperative Consensus Algorithm for Clock Synchronization in Wireless Sensor Networks." In: *87th IEEE Vehicular Technology Conference (VTC Spring)*. Porto, Portugal: IEEE, June 2018, pp. 1–5. DOI: 10.1109/VTCSpring.2018.8417557.

[43] Sajith Mohan Chakkedath and Mary Ann Weitnauer. "Network Time Synchronization Using OFDM-Based Cooperative Consensus (CoCo) Algorithm." In: *88th IEEE Vehicular Technology Conference (VTC Fall)*. Chicago, Illinois, USA: IEEE, Aug. 2018, pp. 1–5. DOI: 10.1109/VTCFall.2018.8690686.

[44] Yong Jun Chang. "Design of Concurrent Cooperative Transmission Systems on Software-Defined Radios." PhD thesis. Georgia Institute of Technology, Dec. 2013. URL: http://hdl.handle.net/1853/50360.

[45] Yong Jun Chang and Mary Ann Ingram. "Convergence Property of Transmit Time Pre-Synchronization for Concurrent Cooperative Communication." In: *IEEE Global Communications Conference (GLOBECOM)*. Miami, Florida, USA: IEEE, Dec. 2010, pp. 1–5. DOI: 10.1109/GLOCOM.2010.5684046.

[46] Yong Jun Chang, Haejoon Jung, Sunghwan Cho, and Mary Ann Weitnauer. "Network Time Synchronization for Large Multi-Hop Sensor Networks using the Cooperative Analog-and-Digital (CANDI) Protocol." In: *IEEE Wireless Communications and Networking Conference (WCNC)*. Istanbul, Turkey: IEEE, Apr. 2014, pp. 1950–1955. DOI: 10.1109/WCNC.2014.6952568.

[47] Yong Jun Chang, Haejoon Jung, and Mary Ann Ingram. "Demonstration of an OLA-Based Cooperative Routing Protocol in an Indoor Environment." In: *17th European Wireless Conference (EW) - Sustainable Wireless Technologies*. Vienna, Austria: IEEE, Apr. 2011, pp. 1–8. URL: https://ieeexplore.ieee.org/abstract/document/5898056.

[48] Yong Jun Chang, Qiongjie Lin, and Mary Ann Weitnauer. "Synchronization for Multi-Hop Distributed MIMO-OFDM." In: *IEEE International Conference on Communications (ICC)*. London, England, UK: IEEE, June 2015, pp. 1745–1750. DOI: 10.1109/ICC.2015.7248577.

[49] Robin Chataut and Robert Akl. "Massive MIMO Systems for 5G and Beyond Networks—Overview, Recent Trends, Challenges, and Future Research Direction." In: *Sensors* 20.10, 2753 (May 2020), pp. 1–35. DOI: 10.3390/s20102753.

[50] Bo Chen, Yue Qiao, Ouyang Zhang, and Kannan Srinivasan. "AirExpress: Enabling Seamless In-Band Wireless Multi-Hop Transmission." In: *21st Annual International Conference on Mobile Computing and Networking (MobiCom)*. Paris, France: ACM, Sept. 2015, pp. 566–577. DOI: 10.1145/2789168.2790114.

[51] Bo Chen, Gopi Krishna Tummala, Yue Qiao, and Kannan Srinivasan. "In-Band Wireless Cut-Through: Is It Possible?" In: *1st ACM Workshop on Hot Topics in Wireless (HotWireless)*. Maui, Hawaii, USA: ACM, Sept. 2014, pp. 1–6. DOI: 10.1145/2643614.2643623.

[52] Lu Chen, Fei Wu, Jiaqi Xu, Kannan Srinivasan, and Ness Shroff. "BiPass: Enabling End-to-End Full Duplex." In: *23rd Annual International Conference on Mobile Computing and Networking (MobiCom)*. Snowbird, Utah, USA: ACM, Oct. 2017, pp. 114–126. DOI: 10.1145/3117811.3117826.

[53] Ming Chen, YanJun Yang, and Jie Chen. "A Design of ±0.28 ppm Temperature-Compensated Crystal Oscillator in a 0.35 µm CMOS Process." In: *Analog Integrated Circuits and Signal Processing* 100.1 (July 2019), pp. 157–166. DOI: 10.1007/s10470-019-01442-w.

[54] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Phil Levis, and Sachin Katti. "Achieving Single Channel, Full Duplex Wireless Communication." In: *16th Annual International Conference on Mobile Computing and Networking (MobiCom)*. Chicago, Illinois, USA: ACM, Sept. 2010, pp. 1–12. DOI: 10.1145/1859995.1859997.

[55] Thomas Clausen, Philippe Jacquet, Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. *Optimized Link State Routing Protocol (OLSR)*. Whitepaper. INRIA, Mar. 2004, pp. 1–57. URL: https://hal.inria.fr/inria-00471712.

[56] Samir R. Das, Charles E. Perkins, and Elizabeth M. Belding-Royer. *Ad Hoc On-Demand Distance Vector (AODV) Routing*. RFC 3561. IETF, July 2003, pp. 1–37. DOI: 10.17487/RFC3561. URL: https://tools.ietf.org/html/rfc3561.

[57] Alexander A. Deev and Andrei A. Kalshchikov. "Subnanosecond Synchronization Method Based on the Synchronous Ethernet Network." In: *3rd International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*. Moscow, Russia: IEEE, Mar. 2021, pp. 1–6. DOI: 10.1109/REEPE51337.2021.9387995.

[58] Alper Demir, Amit Mehrotra, and Jaijeet Roychowdhury. "Phase Noise in Oscillators: A Unifying Theory and Numerical Methods for Characterization." In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 47.5 (May 2000), pp. 655–674. ISSN: 1558-1268. DOI: 10.1109/81.847872.

[59] Jean-Daniel Deschênes, Laura C. Sinclair, Fabrizio R. Giorgetta, William C. Swann, Esther Baumann, Hugo Bergeron, Michael Cermak, Ian Coddington, and Nathan R. Newbury. "Synchronization of Distant Optical Clocks at the Femtosecond Level." In: *Physical Review X* 6.2 (May 2016), pp. 021016 1–15. DOI: 10.1103/PhysRevX.6.021016.

[60] John Dilley, Bruce Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, and Bill Weihl. "Globally Distributed Content Delivery." In: *IEEE Internet Computing* 6.5 (Sept. 2002), pp. 50–58. ISSN: 1941-0131. DOI: 10.1109/MIC.2002.1036038.

[61] Derui Ding, Qing-Long Han, Zidong Wang, and Xiaohua Ge. "A Survey on Model-Based Distributed Control and Filtering for Industrial Cyber-Physical Systems." In: *IEEE Transactions on Industrial Informatics* 15.5 (May 2019), pp. 2483–2499. ISSN: 1941-0050. DOI: 10.1109/TII.2019.2905295.

[62] Manjunath Doddavenkatappa, Mun Choon Chan, and Ben Leong. "Splash: Fast Data Dissemination with Constructive Interference in Wireless Sensor Networks." In: *10th USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*. Lombard, Illinois, USA: USENIX Association, Apr. 2013, pp. 269–282. URL: https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/doddavenkatappa.

[63] Adwait Dongare, Patrick Lazik, Niranjini Rajagopal, and Anthony Rowe. "Pulsar: A Wireless Propagation-Aware Clock Synchronization Platform." In: *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. Pittsburgh, Pennsylvania, USA: IEEE, Apr. 2017, pp. 283–292. DOI: 10.1109/RTAS.2017.9.

[64] Melissa Duarte and Ashutosh Sabharwal. "Full-Duplex Wireless Communications Using Off-the-Shelf Radios: Feasibility and First Results." In: *44th Asilomar Conference on Signals, Systems and Computers (ACSSC)*. Pacific Grove, California, USA: IEEE, Nov. 2010, pp. 1558–1562. DOI: 10.1109/ACSSC.2010.5757799.

[65] Dmitrii Dugaev, Sergey Zinov, Eduard Siemens, and Viacheslav Shuvalov. "A Survey and Performance Evaluation of Ad-Hoc Multi-Hop Routing Protocols for Static Outdoor Networks." In: *International Siberian Conference on Control and Communications (SIBCON)*. Omsk, Russia: IEEE, May 2015, pp. 1–11. DOI: 10.1109/SIBCON.2015.7147048.

[66] Prabal Dutta, Razvan Musaloiu-E., Ion Stoica, and Andreas Terzis. "Wireless ACK Collisions Not Considered Harmful." In: *7th ACM Workshop on Hot Topics in Networks (HotNets-VII)*. Calgary, Alberta, Canada: ACM, Oct. 2008, pp. 1–6.

[67] Jeremy Elson, Lewis Girod, and Deborah Estrin. "Fine-Grained Network Time Synchronization Using Reference Broadcasts." In: *ACM SIGOPS Operating Systems Review* 36.SI (Dec. 2002), pp. 147–163. ISSN: 0163-5980. DOI: 10.1145/844128.844143.

[68] Gunnar Eriksson and Sara Linder. "Analysis of Cooperative Relaying in an Urban Scenario." In: *International Conference on Military Communications and Information Systems (ICMCIS)*. Brussels, Belgium: IEEE, May 2016, pp. 1–5. DOI: 10.1109/ICMCIS.2016.7496579.

[69] *ETSI EN 302 755 v1.4.1 – Digital Video Broadcasting (DVB); Frame Structure Channel Coding and Modulation for a Second Generation Digital Terrestrial Television Broadcasting System (DVB-T2)*. European Standard. ETSI, July 2015.

[70] Doron Ezri and Genadiy Tsodik. "The Impact of Synchronization on Receive Beamforming with Null Steering in OFDM MIMO Systems." In: *27th IEEE Convention of Electrical and Electronics Engineers in Israel*. Eilat, Israel: IEEE, Nov. 2012, pp. 1–4. DOI: 10.1109/EEEI.2012.6376930.

[71] Ye Feng, Chamil Jayasundara, Ampalavanapillai Nirmalathas, and Elaine Wong. "A Feasibility Study of IEEE 802.11 HCCA for Low-Latency Applications." In: *IEEE Transactions on Communications* 67.7 (July 2019), pp. 4928–4938. ISSN: 1558-0857. DOI: 10.1109/TCOMM.2019.2910055.

[72] Federico Ferrari, Marco Zimmerling, Luca Mottola, and Lothar Thiele. "Low-Power Wireless Bus." In: *10th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Toronto, Ontario, Canada: ACM, Nov. 2012, pp. 1–14. DOI: 10.1145/2426656.2426658.

[73] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. "Efficient Network Flooding and Time Synchronization with Glossy." In: *10th IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN)*. Chicago, Illinois, USA: IEEE, Apr. 2011, pp. 73–84. URL: http://ieeexplore.ieee.org/document/5779066.

[74] Gerhard P. Fettweis. "The Tactile Internet: Applications and Challenges." In: *IEEE Vehicular Technology Magazine* 9.1 (Mar. 2014), pp. 64–70. ISSN: 1556-6080. DOI: 10.1109/MVT.2013.2295069.

[75] Rodrigo Fonseca, Sylvia Ratnasamy, Jerry Zhao, Cheng Tien Ee, David Culler, Scott Shenker, and Ion Stoica. "Beacon Vector Routing: Scalable Point-to-Point Routing in Wireless Sensornets." In: *2nd USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*. Vol. 2. Boston, Massachusetts, USA: USENIX Association, May 2005, pp. 329–342.

[76] *Freifunk*. URL: https://freifunk.net.

[77] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava. "Timing-Sync Protocol for Sensor Networks." In: *1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Los Angeles, California, USA: ACM, Nov. 2003, pp. 138–149. DOI: 10.1145/958491.958508.

[78] Sachin Ganu, Kishore Ramachandran, Marco Gruteser, Ivan Seskar, and Jing Deng. "Methods for Restoring MAC Layer Fairness in IEEE 802.11 Networks with Physical Layer Capture." In: *2nd International Workshop on Multi-Hop Ad Hoc Networks: From Theory to Reality (REALMAN)*. Florence, Italy: ACM, May 2006, pp. 7–14. DOI: 10.1145/1132983.1132986.

[79] Zhen Gao, Yong Jun Chang, and Mary Ann Ingram. "Synchronization for Cascaded Distributed MIMO Communications." In: *29th IEEE Military Communications Conference (MILCOM)*. San Jose, California, USA: IEEE, Oct. 2010, pp. 387–392. DOI: 10.1109/MILCOM.2010.5680453.

[80] Paul Gardner-Stephen. "The Serval Project: Practical Wireless Ad-Hoc Mobile Telecommunications." In: *Flinders University, Adelaide, South Australia, Australia, Technical Report* (July 2011), pp. 1–29.

[81] Cengiz Gezer, Chiara Buratti, and Roberto Verdone. "Capture Effect in IEEE 802.15.4 Networks: Modelling and Experimentation." In: *5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*. Modena, Italy: IEEE, May 2010, pp. 204–209. DOI: 10.1109/ISWPC.2010.5483727.

[82] Francisco Girela-López, Jose López-Jiménez, Miguel Jiménez-López, Rafael Rodríguez, Eduardo Ros, and Javier Díaz. "IEEE 1588 High Accuracy Default Profile: Applications and Challenges." In: *IEEE Access* 8 (Mar. 2020), pp. 45211–45220. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2978337.

[83] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. "Collection Tree Protocol." In: *7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Berkeley, California, USA: ACM, Nov. 2009, pp. 1–14. DOI: 10.1145/1644038.1644040.

[84] *GNU Radio*. URL: https://www.gnuradio.org.

[85] Shyamnath Gollakota and Dina Katabi. "ZigZag Decoding: Combating Hidden Terminals in Wireless Networks." In: *ACM Special Interest Group on Data Communication (SIGCOMM) Conference*. Seattle, Washington, USA: ACM, Aug. 2008, pp. 159–170. DOI: 10.1145/1402958.1402977.

[86] Aditya Gudipati and Sachin Katti. "Strider: Automatic Rate Adaptation and Collision Handling." In: *ACM Special Interest Group on Data Communication (SIGCOMM) Conference*. Toronto, Ontario, Canada: ACM, Aug. 2011, pp. 158–169. DOI: 10.1145/2018436.2018455.

[87] Zoran Hadzi-Velkov and Boris Spasenovski. "Capture Effect in IEEE 802.11 Basic Service Area Under Influence of Rayleigh Fading and Near/Far Effect." In: *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Vol. 1. Lisbon, Portugal: IEEE, Sept. 2002, pp. 172–176. DOI: 10.1109/PIMRC.2002.1046683.

[88] Daniel Halperin, Thomas Anderson, and David Wetherall. "Taking the Sting out of Carrier Sense: Interference Cancellation for Wireless LANs." In: *14th Annual International Conference on Mobile Computing and Networking (MobiCom)*. San Francisco, California, USA: ACM, Sept. 2008, pp. 339–350. DOI: 10.1145/1409944.1409983.

[89] Ezzeldin Hamed, Hariharan Rahul, Mohammed A. Abdelghany, and Dina Katabi. "Real-Time Distributed MIMO Systems." In: *ACM Special Interest Group on Data Communication (SIGCOMM) Conference*. Florianopolis, Brazil: ACM, Aug. 2016, pp. 412–425. DOI: 10.1145/2934872.2934905.

[90] Fatemeh Hamidi-Sepehr, Masoud Sajadieh, Sergey Panteleev, Toufiqul Islam, Ingolf Karls, Debdeep Chatterjee, and Junaid Ansari. "5G URLLC: Evolution of High-Performance Wireless Networking for Industrial Automation." In: *IEEE Communications Standards Magazine* 5.2 (June 2021), pp. 132–140. ISSN: 2471-2833. DOI: 10.1109/MCOMSTD.001.2000035.

[91] Kenneth Hann, Sebastien Jobert, and Silvana Rodrigues. "Synchronous Ethernet to Transport Frequency and Phase/Time." In: *IEEE Communications Magazine* 50.8 (Aug. 2012), pp. 152–160. ISSN: 1558-1896. DOI: 10.1109/MCOM.2012.6257542.

[92] Fabian Hohmann and Anja Klein. "Cooperative Forwarding Using Distributed MISO in OFDMA Multihop Networks." In: *86th IEEE Vehicular Technology Conference (VTC Fall)*. Toronto, Ontario, Canada: IEEE, Sept. 2017, pp. 1–7. DOI: 10.1109/VTCFall.2017.8288248.

[93] Jack Kenneth Holmes. *Spread Spectrum Systems for GNSS and Wireless Communications*. GNSS Technology and Applications Series. Boston, Massachusetts, USA: Artech House, 2007. ISBN: 1596930837.

[94] Yao-Win Hong and Anna Scaglione. "Energy-Efficient Broadcasting with Cooperative Transmissions in Wireless Sensor Networks." In: *IEEE Transactions on Wireless Communications* 5.10 (Oct. 2006), pp. 2844–2855. ISSN: 1558-2248. DOI: 10.1109/TWC.2006.04608.

[95] Mark Horowitz, Chih-Kong Ken Yang, and Stefanos Sidiropoulos. "High-Speed Electrical Signaling: Overview and Limitations." In: *IEEE Micro* 18.1 (Jan. 1998), pp. 12–24. ISSN: 1937-4143. DOI: 10.1109/40.653013.

[96] Kai-Cheng Hsu, Kate Ching-Ju Lin, and Hung-Yu Wei. "Full-Duplex Delay-and-Forward Relaying." In: *17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. Paderborn, Germany: ACM, July 2016, pp. 221–230. DOI: 10.1145/2942358.2942370.

[97] Baogang Hu and Raymond G. Gosine. "A New Eigenstructure Method for Sinusoidal Signal Retrieval in White Noise: Estimation and Pattern Recognition." In: *IEEE Transactions on Signal Processing* 45.12 (Dec. 1997), pp. 3073–3083. ISSN: 1941-0476. DOI: 10.1109/78.650268.

[98] Yih-Chun Hu, Dave A. Maltz, and David B. Johnson. *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*. RFC 4728. IETF, Feb. 2007, pp. 1–107. DOI: 10.17487/RFC4728. URL: https://tools.ietf.org/html/rfc4728.

[99] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. *Mobile Edge Computing – A Key Technology Towards 5G*. Whitepaper. ETSI, Sept. 2015, pp. 1–16.

[100] Kaibin Huang, Robert W. Heath, and Jeffrey G. Andrews. "Limited Feedback Beamforming Over Temporally-Correlated Channels." In: *IEEE Transactions on Signal Processing* 57.5 (May 2009), pp. 1959–1975. ISSN: 1941-0476. DOI: 10.1109/TSP.2009.2014272.

[101] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*. IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008). June 2020, pp. 1–499. DOI: 10.1109/IEEESTD.2020.9120376.

[102] *IEEE Standard for Ethernet*. IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012). Mar. 2016, pp. 1–4017. DOI: 10.1109/IEEESTD.2016.7428776.

[103] *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, pp. 1–3534. DOI: 10.1109/IEEESTD.2016.7786995.

[104] *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 802.11-2020 (Revision of IEEE Std 802.11-2016). Feb. 2021, pp. 1–4379. DOI: 10.1109/IEEESTD.2021.9363693.

[105] *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 - Amendment 1: Enhancements for High-Efficiency WLAN*. IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020). May 2021, pp. 1–767. DOI: 10.1109/IEEESTD.2021.9442429.

[106] *IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 2: Logical Link Control*. ISO 8802-2 IEEE Std 802.2, 1st edition (Revision of IEEE Std 802.2-1985). Dec. 1989, pp. 1–114. DOI: 10.1109/IEEESTD.1989.5776621.

[107] *IEEE Standard for Low-Rate Wireless Networks*. IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015). July 2020, pp. 1–800. DOI: 10.1109/IEEESTD.2020.9144691.

[108] Mayank Jain, Jung Il Choi, Taemin Kim, Dinesh Bharadia, Siddharth Seth, Kannan Srinivasan, Philip Levis, Sachin Katti, and Prasun Sinha. "Practical, Real-Time, Full Duplex Wireless." In: *17th Annual International Conference on Mobile Computing and Networking (MobiCom)*. Las Vegas, Nevada, USA: ACM, Sept. 2011, pp. 301–312. DOI: 10.1145/2030613.2030647.

[109] Gentian Jakllari, Srikanth V. Krishnamurthy, Michalis Faloutsos, Prashant V. Krishnamurthy, and Ozgur Ercetin. "A Cross-Layer Framework for Exploiting Virtual MISO Links in Mobile Ad Hoc Networks." In: *IEEE Transactions on Mobile Computing* 6.6 (June 2007), pp. 579–594. ISSN: 1558-0660. DOI: 10.1109/TMC.2007.1068.

[110] Suhanya Jayaprakasam, Sharul Kamal Abdul Rahim, and Chee Yen Leow. "Distributed and Collaborative Beamforming in Wireless Sensor Networks: Classifications, Trends, and Research Directions." In: *IEEE Communications Surveys & Tutorials* 19.4 (Fourthquarter 2017), pp. 2092–2116. ISSN: 1553-877X. DOI: 10.1109/COMST.2017.2720690.

[111] Yi Jiang, Babak Daneshrad, and Gregory J. Pottie. "A Practical Approach to Joint Timing, Frequency Synchronization and Channel Estimation for Concurrent Transmissions in a MANET." In: *IEEE Transactions on Wireless Communications* 16.6 (June 2017), pp. 3461–3475. ISSN: 1558-2248. DOI: 10.1109/TWC.2017.2682840.

[112] Chris Johnson. *5G New Radio in Bullets*. Independently Published, 2019. ISBN: 9781077484351.

[113] Jingon Joung, Yeow Khiang Chia, and Sumei Sun. "Energy-Efficient, Large-Scale Distributed-Antenna System (L-DAS) for Multiple Users." In: *IEEE Journal of Selected Topics in Signal Processing* 8.5 (Oct. 2014), pp. 954–965. ISSN: 1941-0484. DOI: 10.1109/JSTSP.2014.2309942.

[114] Haejoon Jung, Yong Jun Chang, and Mary Ann Ingram. "Experimental Range Extension of Concurrent Cooperative Transmission in Indoor Environments at 2.4GHz." In: *29th IEEE Military Communications Conference (MILCOM)*. San Jose, California, USA: IEEE, Oct. 2010, pp. 148–153. DOI: 10.1109/MILCOM.2010.5680162.

[115] Haejoon Jung and Mary Ann Weitnauer. "Multi-Packet Interference in Opportunistic Large Array Broadcasts over Disk Networks." In: *IEEE Transactions on Wireless Communications* 12.11 (Nov. 2013), pp. 5631–5645. ISSN: 1558-2248. DOI: 10.1109/TWC.2013.093013.121912.

[116] Gaurav Khanna and Sanjay K. Chaturvedi. "A Comprehensive Survey on Multi-Hop Wireless Networks: Milestones, Changing Trends and Concomitant Challenges." In: *Wireless Personal Communications* 101.2 (July 2018), pp. 677–722. DOI: 10.1007/s11277-018-5711-8.

[117] Andrzej Kochut, Arunchandar Vasan, A. Udaya Shankar, and Ashok Agrawala. "Sniffing Out the Correct Physical Layer Capture Model in 802.11b." In: *12th IEEE International Conference on Network Protocols (ICNP)*. Berlin, Germany: IEEE, Oct. 2004, pp. 252–261. DOI: 10.1109/ICNP.2004.1348115.

[118] Linghe Kong and Xue Liu. "mZig: Enabling Multi-Packet Reception in ZigBee." In: *21st Annual International Conference on Mobile Computing and Networking (MobiCom)*. Paris, France: ACM, Sept. 2015, pp. 552–565. DOI: 10.1145/2789168.2790104.

[119] Rakhesh Singh Kshetrimayum. *Fundamentals of MIMO Wireless Communications*. Cambridge, England, UK: Cambridge University Press, 2017. ISBN: 9781108415699.

[120] Amy Kumar, Soura Dasgupta, and Raghuraman Mudumbai. "Distributed Null-forming Without Prior Frequency Synchronization." In: *Australian Control Conference (AUCC)*. Perth, Western Australia, Australia: IEEE, Nov. 2013, pp. 207–211. DOI: 10.1109/AUCC.2013.6697274.

[121] Mathias Kurth, Anatolij Zubow, and Jens-Peter Redlich. "Cooperative Opportunistic Routing Using Transmit Diversity in Wireless Mesh Networks." In: *27th IEEE International Conference on Computer Communications (INFOCOM)*. Phoenix, Arizona, USA: IEEE, Apr. 2008, pp. 1310–1318. DOI: 10.1109/INFOCOM.2008.188.

[122] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E. Wright. "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions." In: *SIAM Journal on Optimization* 9.1 (Dec. 1998), pp. 112–147. DOI: 10.1137/S1052623496303470.

[123] Sriram Lakshmanan and Raghupathy Sivakumar. "Diversity Routing for Multi-Hop Wireless Networks with Cooperative Transmissions." In: *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. Rome, Italy: IEEE, June 2009, pp. 1–9. DOI: 10.1109/SAHCN.2009.5168928.

[124] Olaf Landsiedel, Federico Ferrari, and Marco Zimmerling. "Chaos: Versatile and Efficient All-to-All Data Sharing and In-Network Processing at Scale." In: *11th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Rome, Italy: ACM, Nov. 2013, pp. 1–14. DOI: 10.1145/2517351.2517358.

[125] Andres Laya, Charalampos Kalalas, Francisco Vazquez-Gallego, Luis Alonso, and Jesus Alonso-Zarate. "Goodbye, ALOHA!" In: *IEEE Access* 4 (May 2016), pp. 2029–2044. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2557758.

[126] Trung-Kien Le, Umer Salim, and Florian Kaltenberger. "An Overview of Physical Layer Design for Ultra-Reliable Low-Latency Communications in 3GPP Releases 15, 16, and 17." In: *IEEE Access* 9 (Dec. 2020), pp. 433–444. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3046773.

[127] Jeongkeun Lee, Wonho Kim, Sung-Ju Lee, Daehyung Jo, Jiho Ryu, Taekyoung Kwon, and Yanghee Choi. "An Experimental Study on the Capture Effect in 802.11a Networks." In: *2nd ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WinTECH)*. Montréal, Québec, Canada: ACM, Sept. 2007, pp. 19–26. DOI: 10.1145/1287767.1287772.

[128] Christoph Lenzen, Philipp Sommer, and Roger Wattenhofer. "Optimal Clock Synchronization in Networks." In: *7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Berkeley, California, USA: ACM, Nov. 2009, pp. 225–238. DOI: 10.1145/1644038.1644061.

[129] Xiaohua Li, Fan Ng, and Taewoo Han. "Carrier Frequency Offset Mitigation in Asynchronous Cooperative OFDM Transmissions." In: *IEEE Transactions on Signal Processing* 56.2 (Feb. 2008), pp. 675–685. ISSN: 1941-0476. DOI: 10.1109/TSP.2007.907820.

[130] Zan Li, Torsten Braun, and Desislava C. Dimitrova. "Methodology for GPS Synchronization Evaluation with High Accuracy." In: *81st IEEE Vehicular Technology Conference (VTC Spring)*. Glasgow, Scotland, UK: IEEE, May 2015, pp. 1–6. DOI: 10.1109/VTCSpring.2015.7145929.

[131] Patrick Lieser, Flor Alvarez, Paul Gardner-Stephen, Matthias Hollick, and Doreen Boehnstedt. "Architecture for Responsive Emergency Communications Networks." In: *IEEE Global Humanitarian Technology Conference (GHTC)*. San Jose, California, USA: IEEE, Oct. 2017, pp. 1–9. DOI: 10.1109/GHTC.2017.8239239.

[132] Hyojun Lim and Chongkwon Kim. "Flooding in Wireless Ad Hoc Networks." In: *Computer Communications* 24.3 (Feb. 2001), pp. 353–363. ISSN: 0140-3664. DOI: 10.1016/S0140-3664(00)00233-4.

[133] Jian Lin, Haejoon Jung, Yong Jun Chang, Jin Woo Jung, and Mary Ann Weitnauer. "On Cooperative Transmission Range Extension in Multi-Hop Wireless Ad-Hoc and Sensor Networks: A Review." In: *Ad Hoc Networks* 29 (June 2015), pp. 117–134. ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2015.01.018.

[134] Kaisen Lin and Philip Levis. "Data Discovery and Dissemination with DIP." In: *7th IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN)*. St. Louis, Missouri, USA: IEEE, Apr. 2008, pp. 433–444. DOI: 10.1109/IPSN.2008.17.

[135] Qiongjie Lin, Yong Jun Chang, Feng Wang, and Mary Ann Weitnauer. "Implementation and Analysis of Multi-User MIMO with Amplify-and-Forward Relaying." In: *IEEE International Conference on Communications Workshops (ICC Workshops)*. Sydney, New South Wales, Australia: IEEE, June 2014, pp. 32–37. DOI: 10.1109/ICCW.2014.6881168.

[136] Chunmei Liu, Chen Shen, Jack Chuang, Richard A. Rouil, and Hyeong-Ah Choi. "Evaluating Unicast and MBSFN in Public Safety Networks." In: *31st IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Virtual Conference: IEEE, Aug. 2020, pp. 1–7. DOI: 10.1109/PIMRC48278.2020.9217346.

[137] *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); LTE Physical Layer; General Description (3GPP TS 36.201 Version 15.1.0 Release 15)*. Technical Specification. 3GPP, July 2018.

[138] Jiakang Lu and Kamin Whitehouse. "Flash Flooding: Exploiting the Capture Effect for Rapid Flooding in Wireless Sensor Networks." In: *28th IEEE International Conference on Computer Communications (INFOCOM)*. Rio de Janeiro, Brazil: IEEE, Apr. 2009, pp. 2491–2499. DOI: 10.1109/INFCOM.2009.5062177.

[139] Anh Luong, Peter Hillyard, Alemayehu Abrar, Charissa Che, Anthony Rowe, Thomas Schmid, and Neal Patwari. "A Stitch in Time and Frequency Synchronization Saves Bandwidth." In: *17th IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN)*. Porto, Portugal: IEEE, Apr. 2018, pp. 96–107. DOI: 10.1109/IPSN.2018.00016.

[140] Ming Ma and Yuanyuan Yang. "A Novel Contention-Based MAC Protocol with Channel Reservation for Wireless LANs." In: *IEEE Transactions on Wireless Communications* 7.10 (Oct. 2008), pp. 3748–3758. ISSN: 1558-2248. DOI: 10.1109/T-WC.2008.08885.

[141] Zachary MacHardy, Ashiq Khan, Kazuaki Obana, and Shigeru Iwashina. "V2X Access Technologies: Regulation, Research, and Remaining Challenges." In: *IEEE Communications Surveys & Tutorials* 20.3 (Thirdquarter 2018), pp. 1858–1877. ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2808444.

[142] Anthony Magee. "Synchronization in Next-Generation Mobile Backhaul Networks." In: *IEEE Communications Magazine* 48.10 (Oct. 2010), pp. 110–116. ISSN: 1558-1896. DOI: 10.1109/MCOM.2010.5594685.

[143] David A. Maltz, Josh Broch, and David B. Johnson. "Lessons from a Full-Scale Multihop Wireless Ad Hoc Network Testbed." In: *IEEE Personal Communications* 8.1 (Feb. 2001), pp. 8–15. ISSN: 1558-0652. DOI: 10.1109/98.904894.

[144] *Man page of `pcap_inject()`, part of libpcap*. Mar. 2022. URL: https://www.tcpdump.org/manpages/pcap_inject.3pcap.html.

[145] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B. Letaief. "A Survey on Mobile Edge Computing: The Communication Perspective." In: *IEEE Communications Surveys & Tutorials* 19.4 (Fourthquarter 2017), pp. 2322–2358. ISSN: 1553-877X. DOI: 10.1109/COMST.2017.2745201.

[146] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. "The Flooding Time Synchronization Protocol." In: *2nd ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Baltimore, Maryland, USA: ACM, Nov. 2004, pp. 39–49. DOI: 10.1145/1031495.1031501.

[147] Patrick Marsch and Gerhard P. Fettweis. *Coordinated Multi-Point in Mobile Communications: From Theory to Practice*. Cambridge, England, UK: Cambridge University Press, July 2011. ISBN: 9781107004115.

[148] Patrick Marsch and Gerhard P. Fettweis. "Static Clustering for Cooperative Multi-Point (CoMP) in Mobile Communications." In: *IEEE International Conference on Communications (ICC)*. Kyoto, Japan: IEEE, June 2011, pp. 1–6. DOI: 10.1109/icc.2011.5963458.

[149] *Matlab `boxplot`*. URL: https://www.mathworks.com/help/stats/boxplot.html.

[150] *Matlab WLAN Toolbox*. URL: https://www.mathworks.com/products/wlan.html.

[151] Michalis Mavrovouniotis, Changhe Li, and Shengxiang Yang. "A Survey of Swarm Intelligence for Dynamic Optimization: Algorithms and Applications." In: *Swarm and Evolutionary Computation* 33 (Apr. 2017), pp. 1–17. ISSN: 2210-6502. DOI: 10.1016/j.swevo.2016.12.005.

[152] Adrian Meier. *Freifunk: freie Funknetze als Chance für den ländlichen Raum*. München, Deutschland: AVM, 2010. ISBN: 9783899757811.

[153] Serge R. Mghabghab and Jeffrey A. Nanzer. "Impact of VCO and PLL Phase Noise on Distributed Beamforming Arrays With Periodic Synchronization." In: *IEEE Access* 9 (Apr. 2021), pp. 56578–56588. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3071637.

[154] Serge R. Mghabghab and Jeffrey A. Nanzer. "Open-Loop Distributed Beamforming Using Wireless Frequency Synchronization." In: *IEEE Transactions on Microwave Theory and Techniques* 69.1 (Jan. 2021), pp. 896–905. ISSN: 1557-9670. DOI: 10.1109/TMTT.2020.3022385.

[155] David L. Mills. *Computer Network Time Synchronization: The Network Time Protocol on Earth and in Space*. 2nd ed. Boca Raton, Florida, USA: CRC Press, 2017. ISBN: 9781439814642.

[156] Andreas F. Molisch, Vishnu V. Ratnam, Shengqian Han, Zheda Li, Sinh Le Hong Nguyen, Linsheng Li, and Katsuyuki Haneda. "Hybrid Beamforming for Massive MIMO: A Survey." In: *IEEE Communications Magazine* 55.9 (Sept. 2017), pp. 134–141. ISSN: 1558-1896. DOI: 10.1109/MCOM.2017.1600400.

[157] Bishwarup Mondal, Victor Sergeev, Avik Sengupta, Gregory Ermolaev, Alexei Davydov, Eddy Kwon, Seunghee Han, and Apostolos Papathanassiou. "MU-MIMO and CSI Feedback Performance of NR/LTE." In: *53rd Annual Conference on Information Sciences and Systems (CISS)*. Baltimore, Maryland, USA: IEEE, Mar. 2019, pp. 1–6. DOI: 10.1109/CISS.2019.8692922.

[158] Todd K. Moon. *Error Correction Coding: Mathematical Methods and Algorithms*. Hoboken, New Jersey: John Wiley & Sons, 2005. ISBN: 9780471739210. DOI: 10.1002/0471739219.

[159] Paul H. Moose. "A Technique for Orthogonal Frequency Division Multiplexing Frequency Offset Correction." In: *IEEE Transactions on Communications* 42.10 (Oct. 1994), pp. 2908–2914. ISSN: 0090-6778. DOI: 10.1109/26.328961.

[160] Mahdi Mousavi and Anja Klein. "Energy and Social Cost Minimization for Data Dissemination in Wireless Networks: Centralized and Decentralized Approaches." In: *IEEE Transactions on Vehicular Technology* 69.5 (May 2020), pp. 5521–5534. ISSN: 1939-9359. DOI: 10.1109/TVT.2020.2981894.

[161] Mahdi Mousavi, Hussein Al-Shatri, and Anja Klein. "Cost Sharing Games for Energy-Efficient Multi-Hop Broadcast in Wireless Networks." In: *IEEE Transactions on Wireless Communications* 19.1 (Jan. 2020), pp. 310–324. ISSN: 1558-2248. DOI: 10.1109/TWC.2019.2944374.

[162] Raghuraman Mudumbai, Gwen Barriac, and Upamanyu Madhow. "On the Feasibility of Distributed Beamforming in Wireless Networks." In: *IEEE Transactions on Wireless Communications* 6.5 (May 2007), pp. 1754–1763. ISSN: 1558-2248. DOI: 10.1109/TWC.2007.360377.

[163] Raghuraman Mudumbai, D. Richard Brown III, Upamanyu Madhow, and H. Vincent Poor. "Distributed Transmit Beamforming: Challenges and Recent Progress." In: *IEEE Communications Magazine* 47.2 (Feb. 2009), pp. 102–110. ISSN: 1558-1896. DOI: 10.1109/MCOM.2009.4785387.

[164] Raghuraman Mudumbai, João P. Hespanha, Upamanyu Madhow, and Gwen Barriac. "Scalable Feedback Control for Distributed Beamforming in Sensor Networks." In: *International Symposium on Information Theory (ISIT)*. Adelaide, South Australia, Australia: IEEE, Sept. 2005, pp. 137–141. DOI: 10.1109/ISIT.2005.1523309.

[165] Raghuraman Mudumbai, João P. Hespanha, Upamanyu Madhow, and Gwen Barriac. "Distributed Transmit Beamforming Using Feedback Control." In: *IEEE Transactions on Information Theory* 56.1 (Jan. 2010), pp. 411–426. ISSN: 1557-9654. DOI: 10.1109/TIT.2009.2034786.

[166] Niklas Müller. "Video Broadcasting in IEEE 802.11 Multihop Networks." BSc thesis. Technische Universität Darmstadt, Aug. 2021.

[167] Patrick Murphy. "Design, Implementation and Characterization of a Cooperative Communications System." PhD thesis. Rice University Department of Electrical and Computer Engineering, Dec. 2010. URL: http://warp.rice.edu/w/MurphyPhDThesis.

[168] Patrick Murphy and Ashutosh Sabharwal. "Design, Implementation, and Characterization of a Cooperative Communications System." In: *IEEE Transactions on Vehicular Technology* 60.6 (July 2011), pp. 2534–2544. ISSN: 1939-9359. DOI: 10.1109/TVT.2011.2158461.

[169] Farhan Nawaz, Hemant Kumar, Syed Ali Hassan, and Haejoon Jung. "Opportunistic Large Array Propagation Models: A Comprehensive Survey." In: *Sensors* 21.12 (June 2021), pp. 1–25. ISSN: 1424-8220. DOI: 10.3390/s21124206.

[170] Diep N. Nguyen and Marwan Krunz. "Be Responsible: A Novel Communications Scheme for Full-Duplex MIMO Radios." In: *34th IEEE International Conference on Computer Communications (INFOCOM)*. Hong Kong, China: IEEE, Apr. 2015, pp. 1733–1741. DOI: 10.1109/INFOCOM.2015.7218554.

[171] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. "The Broadcast Storm Problem in a Mobile Ad Hoc Network." In: *5th Annual International Conference on Mobile Computing and Networking (MobiCom)*. Seattle, Washington, USA: ACM, Aug. 1999, pp. 151–162. DOI: 10.1145/313451.313525.

[172] Tsutomu Niiho, Mariko Nakaso, Koichi Masuda, Hiroyuki Sasai, Kuniaki Utsumi, and Masaru Fuse. "Multi-Channel Wireless LAN Distributed Antenna System Based on Radio-over-Fiber Techniques." In: *17th Annual Meeting of the IEEE Lasers and Electro-Optics Society (LEOS)*. Vol. 1. Río Grande, Puerto Rico, USA: IEEE, Nov. 2004, pp. 57–58. DOI: 10.1109/LEOS.2004.1363108.

[173] Jens Ohm and Hans Dieter Lüke. *Signalübertragung: Grundlagen der digitalen und analogen Nachrichtenübertragungssysteme*. 10th ed. Springer-Lehrbuch. Berlin, Heidelberg: Springer, 2007. ISBN: 9783540692584. DOI: 10.1007/978-3-540-69258-4.

[174] Jongjun Park, Jongsoo Jeong, Hoon Jeong, Chieh-Jan Mike Liang, and JeongGil Ko. "Improving the Packet Delivery Performance for Concurrent Packet Transmissions in WSNs." In: *IEEE Communications Letters* 18.1 (Jan. 2014), pp. 58–61. ISSN: 1558-2558. DOI: 10.1109/LCOMM.2013.112013.131974.

[175] Bradford W. Parkinson and Stephen W. Gilbert. "NAVSTAR: Global Positioning System—Ten Years Later." In: *Proceedings of the IEEE* 71.10 (Oct. 1983), pp. 1177–1186. ISSN: 1558-2256. DOI: 10.1109/PROC.1983.12745.

[176] Pulin Patel and Jack Holtzman. "Analysis of a Simple Successive Interference Cancellation Scheme in a DS/CDMA System." In: *IEEE Journal on Selected Areas in Communications* 12.5 (June 1994), pp. 796–807. ISSN: 1558-0008. DOI: 10.1109/49.298053.

[177] Pulin Patel and Jack Holtzman. "Performance Comparison of a DS/CDMA System Using a Successive Interference Cancellation (IC) Scheme and a Parallel IC Scheme Under Fading." In: *IEEE International Conference on Communications (ICC) / SUPERCOMM*. Vol. 1. New Orleans, Louisiana, USA: IEEE, May 1994, pp. 510–514. DOI: 10.1109/ICC.1994.368852.

[178] Ben Peiffer, Raghu Mudumbai, Sairam Goguri, Anton Kruger, and Soura Dasgupta. "Experimental Demonstration of Nullforming From a Fully Wireless Distributed Array." In: *35th IEEE Military Communications Conference (MILCOM)*. Baltimore, Maryland, USA: IEEE, Nov. 2016, pp. 442–447. DOI: 10.1109/MILCOM.2016.7795367.

[179] Ben Peiffer, Raghu Mudumbai, Sairam Goguri, Anton Kruger, and Soura Dasgupta. "Experimental Demonstration of Retrodirective Beamforming From a Fully Wireless Distributed Array." In: *35th IEEE Military Communications Conference (MILCOM)*. Baltimore, Maryland, USA: IEEE, Nov. 2016, pp. 442–447. DOI: 10.1109/MILCOM.2016.7795367.

[180] Ben Peiffer, Raghu Mudumbai, Anton Kruger, Amy Kumar, and Soura Dasgupta. "Experimental Demonstration of a Distributed Antenna Array Pre-synchronized for Retrodirective Transmission." In: *50th Annual Conference on Information Sciences and Systems (CISS)*. Princeton, New Jersey, USA: IEEE, Mar. 2016, pp. 460–465. DOI: 10.1109/CISS.2016.7460546.

[181] Eldad Perahia and Robert Stacey. *Next Generation Wireless LANs: 802.11n and 802.11ac*. 2nd ed. Cambridge, England, UK: Cambridge University Press, 2013. ISBN: 9781107016767.

[182] Gregers Petersen. *Subversion, Conversion, Development: Cross-Cultural Knowledge Exchange and the Politics of Design*. Cambridge, Massachusetts, USA: MIT Press, 2014. Chap. 3 (Freifunk: When technology and politics assemble into subversion), pp. 39–56. ISBN: 9780262322508.

[183] Thierry Pollet, Mark van Bladel, and Marc Moeneclaey. "BER Sensitivity of OFDM Systems to Carrier Frequency Offset and Wiener Phase Noise." In: *IEEE Transactions on Communications* 43.2/3/4 (Feb. 1995), pp. 191–193. ISSN: 0090-6778. DOI: 10.1109/26.380034.

[184] Hang Qiu, Konstantinos Psounis, Giuseppe Caire, Keith M. Chugg, and Kaidong Wang. "High-Rate WiFi Broadcasting in Crowded Scenarios via Lightweight Coordination of Multiple Access Points." In: *17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. Paderborn, Germany: ACM, July 2016, pp. 301–310. DOI: 10.1145/2942358.2942372.

[185] François Quitin, Muhammad Mahboob Ur Rahman, Raghuraman Mudumbai, and Upamanyu Madhow. "Distributed Beamforming with Software-Defined Radios: Frequency Synchronization and Digital Feedback." In: *IEEE Global Communications Conference (GLOBECOM)*. Anaheim, California, USA: IEEE, Dec. 2012, pp. 4787–4792. DOI: 10.1109/GLOCOM.2012.6503876.

[186] Hariharan Rahul, Haitham Hassanieh, and Dina Katabi. "SourceSync: A Distributed Wireless Architecture for Exploiting Sender Diversity." In: *ACM Special Interest Group on Data Communication (SIGCOMM) Conference*. New Delhi, India: ACM, Aug. 2010, pp. 171–182. DOI: 10.1145/1851182.1851204.

[187] Hariharan Rahul, Swarun Kumar, and Dina Katabi. "JMB: Scaling Wireless Capacity with User Demands." In: *ACM Special Interest Group on Data Communication (SIGCOMM) Conference*. Helsinki, Finland: ACM, Aug. 2012, pp. 235–246. DOI: 10.1145/2342356.2342401.

[188] Gonzalo M. Ramírez-Ávila, Jürgen Kurths, and Jean Louis Deneubourg. "Fireflies: A Paradigm in Synchronization." In: *Chaotic, Fractional, and Complex Dynamics: New Insights and Perspectives*. Ed. by Mark Edelman, Elbert E. N. Macau, and Miguel A. F. Sanjuan. Cham, Switzerland: Springer, 2018, pp. 35–64. ISBN: 9783319681092. DOI: 10.1007/978-3-319-68109-2_3.

[189] *Recommendation ITU-T G.8261/Y.1361 – Timing and Synchronization Aspects in Packet Networks*. Telecommunication Standardization Sector of ITU (ITU-T), Aug. 2019.

[190] *Recommendation ITU-T G.8262/Y.1362 – Timing Characteristics of Synchronous Equipment Slave Clock*. Telecommunication Standardization Sector of ITU (ITU-T), Nov. 2018.

[191] *Recommendation ITU-T G.8264/Y.1364 – Distribution of Timing Information Through Packet Networks*. Telecommunication Standardization Sector of ITU (ITU-T), Aug. 2017.

[192] Mark C. Reed, Christian B. Schlegel, Paul D. Alexander, and John A. Asenstorfer. "Iterative Multiuser Detection for CDMA with FEC: Near-Single-User Performance." In: *IEEE Transactions on Communications* 46.12 (Dec. 1998), pp. 1693–1699. ISSN: 1558-0857. DOI: 10.1109/26.737408.

[193] Mattia Rizzi, Stefano Rinaldi, Paolo Ferrari, Alessandra Flammini, Daniele Fontanelli, and David Macii. "Enhancing Accuracy and Robustness of Frequency Transfer Using Synchronous Ethernet and Multiple Network Paths." In: *IEEE Transactions on Instrumentation and Measurement* 65.8 (Aug. 2016), pp. 1926–1936. ISSN: 0018-9456. DOI: 10.1109/TIM.2016.2559950.

[194] Lawrence G. Roberts. "ALOHA Packet System with and without Slots and Capture." In: *ACM SIGCOMM Computer Communication Review* 5.2 (Apr. 1975), pp. 28–42. ISSN: 0146-4833. DOI: 10.1145/1024916.1024920.

[195] Vladimir Roskin. "Separated Channel Estimation in Asynchronous Multi-User Wi-Fi Transmissions via Interpolation with Deep Neural Networks." BSc thesis. Technische Universität Darmstadt, Nov. 2018.

[196] Vladimir Roskin. "DNN-based Enhancements for Time-Variant Zero-Forcing." MSc thesis. Technische Universität Darmstadt, May 2021.

[197] Harshavardhan Sabbineni and Krishnendu Chakrabarty. "Location-Aided Flooding: An Energy-Efficient Data Dissemination Protocol for Wireless Sensor Networks." In: *IEEE Transactions on Computers* 54.1 (Jan. 2005), pp. 36–46. ISSN: 1557-9956. DOI: 10.1109/TC.2005.8.

[198] Hemant Saggar, Yi Jiang, Babak Daneshrad, and Greg Pottie. "A Concurrent CSMA MAC Protocol for Mobile Ad Hoc Networks Using Beamnulling." In: *34th IEEE Military Communications Conference (MILCOM)*. Tampa, Florida, USA: IEEE, Oct. 2015, pp. 97–102. DOI: 10.1109/MILCOM.2015.7357425.

[199] Achaleshwar Sahai, Gaurav Patel, and Ashutosh Sabharwal. "Pushing the Limits of Full-Duplex: Design and Real-Time Implementation." In: *CoRR* arXiv:1107.0607 [cs.NI] (July 2011), pp. 1–12. URL: http://arxiv.org/abs/1107.0607.

[200] Adel A. M. Saleh and Reinaldo A. Valenzuela. "A Statistical Model for Indoor Multipath Propagation." In: *IEEE Journal on Selected Areas in Communications* 5.2 (Feb. 1987), pp. 128–137. ISSN: 0733-8716. DOI: 10.1109/JSAC.1987.1146527.

[201] Naveen Santhapuri, Justin Manweiler, Souvik Sen, Romit Roy Choudhury, Srihari Nelakuduti, and Kamesh Munagala. "Message in Message (MIM): A Case for Reordering Transmissions in Wireless Networks." In: *7th ACM Workshop on Hot Topics in Networks (HotNets-VII)*. Calgary, Alberta, Canada: ACM, Oct. 2008, pp. 25–30.

[202] Anna Scaglione and Yao-Win Hong. "Opportunistic Large Arrays: Cooperative Transmission in Wireless Multihop Ad Hoc Networks to Reach Far Distances." In: *IEEE Transactions on Signal Processing* 51.8 (Aug. 2003), pp. 2082–2092. ISSN: 1053-587X. DOI: 10.1109/TSP.2003.814519.

[203] Timothy M. Schmidl and Donald C. Cox. "Robust Frequency and Timing Synchronization for OFDM." In: *IEEE Transactions on Communications* 45.12 (Dec. 1997), pp. 1613–1621. ISSN: 0090-6778. DOI: 10.1109/26.650240.

[204] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. *Nexmon: The C-Based Firmware Patching Framework*. 2017. URL: https://nexmon.org.

[205] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. "CSMA/CN: Carrier Sense Multiple Access With Collision Notification." In: *IEEE/ACM Transactions on Networking* 20.2 (Apr. 2012), pp. 544–556. ISSN: 1558-2566. DOI: 10.1109/TNET.2011.2174461.

[206] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. "Successive Interference Cancellation: Carving Out MAC Layer Opportunities." In: *IEEE Transactions on Mobile Computing* 12.2 (Feb. 2013), pp. 346–357. ISSN: 1558-0660. DOI: 10.1109/TMC.2012.17.

[207] *Serval Project*. URL: http://servalproject.org.

[208] Farrukh Salim Shaikh and Roland Wismüller. "Routing in Multi-Hop Cellular Device-to-Device (D2D) Networks: A Survey." In: *IEEE Communications Surveys & Tutorials* 20.4 (Fourthquarter 2018), pp. 2622–2657. ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2848108.

[209] Sajid M. Sheikh, Riaan Wolhuter, and Herman A. Engelbrecht. "A Survey of Cross-Layer Protocols for IEEE 802.11 Wireless Multi-Hop Mesh Networks." In: *International Journal of Communication Systems* 30.6, 3129 (Apr. 2017), pp. 1–32. DOI: 10.1002/dac.3129.

[210] Clayton Shepard, Rahman Doost-Mohammady, Jian Ding, Ryan E. Guerra, and Lin Zhong. "ArgosNet: A Multi-Cell Many-Antenna MU-MIMO Platform." In: *52nd Asilomar Conference on Signals, Systems and Computers (ACSSC)*. Pacific Grove, California, USA: IEEE, Oct. 2018, pp. 2237–2241. DOI: 10.1109/ACSSC.2018.8645415.

[211] Clayton Shepard, Hang Yu, Narendra Anand, Erran Li, Thomas Marzetta, Richard Yang, and Lin Zhong. "Argos: Practical Many-Antenna Base Stations." In: *18th Annual International Conference on Mobile Computing and Networking (MobiCom)*. Istanbul, Turkey: ACM, Aug. 2012, pp. 53–64. DOI: 10.1145/2348543.2348553.

[212] Birsen Sirkeci-Mergen, Anna Scaglione, and Gökhan Mergen. "Asymptotic Analysis of Multistage Cooperative Broadcast in Wireless Networks." In: *IEEE Transactions on Information Theory* 52.6 (June 2006), pp. 2531–2550. ISSN: 1557-9654. DOI: 10.1109/TIT.2006.874514.

[213] George Sklivanitis, Konstantinos Alexandris, and Aggelos Bletsas. "Testbed for Non-Coherent Zero-Feedback Distributed Beamforming." In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vancouver, British Columbia, Canada: IEEE, May 2013, pp. 2563–2567. DOI: 10.1109/ICASSP.2013.6638118.

[214] George Sklivanitis and Aggelos Bletsas. "Testing Zero-Feedback Distributed Beamforming with a Low-Cost SDR Testbed." In: *45th Asilomar Conference on Signals, Systems and Computers (ACSSC)*. Pacific Grove, California, USA: IEEE, Nov. 2011, pp. 104–108. DOI: 10.1109/ACSSC.2011.6189964.

[215] Maja Sliskovic. "Carrier and Sampling Frequency Offset Estimation and Correction in Multicarrier Systems." In: *IEEE Global Telecommunications Conference (GLOBECOM)*. Vol. 1. San Antonio, Texas, USA: IEEE, Nov. 2001, pp. 285–289. DOI: 10.1109/GLOCOM.2001.965124.

[216] Dongjin Son, Bhaskar Krishnamachari, and John Heidemann. "Experimental Study of Concurrent Transmission in Wireless Sensor Networks." In: *4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Boulder, Colorado, USA: ACM, Oct. 2006, pp. 237–250. DOI: 10.1145/1182807.1182831.

[217] Elvino S. Sousa and John A. Silvester. "Optimum Transmission Ranges in a Direct-Sequence Spread-Spectrum Multihop Packet Radio Network." In: *IEEE Journal on Selected Areas in Communications* 8.5 (June 1990), pp. 762–771. ISSN: 1558-0008. DOI: 10.1109/49.56383.

[218] Joachim Speidel. *Introduction to Digital Communications*. 1st ed. Signals and Communication Technology. Cham, Switzerland: Springer, 2019. ISBN: 9783030005481. DOI: 10.1007/978-3-030-00548-1.

[219] Quentin H. Spencer, Christian B. Peel, A. Lee Swindlehurst, and Martin Haardt. "An Introduction to the Multi-User MIMO Downlink." In: *IEEE Communications Magazine* 42.10 (Oct. 2004), pp. 60–67. ISSN: 1558-1896. DOI: 10.1109/MCOM.2004.1341262.

[220] Kun Tan, He Liu, Ji Fang, Wei Wang, Jiansong Zhang, Mi Chen, and Geoffrey M. Voelker. "SAM: Enabling Practical Spatial Multiple Access in Wireless LAN." In: *15th Annual International Conference on Mobile Computing and Networking (MobiCom)*. Beijing, China: ACM, Sept. 2009, pp. 49–60. DOI: 10.1145/1614320.1614327.

[221] Aimin Tang and Xudong Wang. "Medium Access Control for a Wireless LAN with a Full Duplex AP and Half Duplex Stations." In: *IEEE Global Communications Conference (GLOBECOM)*. Austin, Texas, USA: IEEE, Dec. 2014, pp. 4732–4737. DOI: 10.1109/GLOCOM.2014.7037555.

[222] Arash Saber Tehrani, Alexandros G. Dimakis, and Michael J. Neely. "SigSag: Iterative Detection Through Soft Message-Passing." In: *IEEE Journal of Selected Topics in Signal Processing* 5.8 (Dec. 2011), pp. 1512–1523. ISSN: 1941-0484. DOI: 10.1109/JSTSP.2011.2169042.

[223] Federico Terraneo, Alberto Leva, Silvano Seva, Martina Maggio, and Alessandro Vittorio Papadopoulos. "Reverse Flooding: Exploiting Radio Interference for Efficient Propagation Delay Compensation in WSN Clock Synchronization." In: *36th IEEE Real-Time Systems Symposium (RTSS)*. San Antonio, Texas, USA: IEEE, Dec. 2015, pp. 175–184. DOI: 10.1109/RTSS.2015.24.

[224] Lakshmi Thanayankizil and Mary Ann Ingram. "Reactive Robust Routing with Opportunistic Large Arrays." In: *IEEE International Conference on Communications Workshops (ICC Workshops)*. Dresden, Germany: IEEE, June 2009, pp. 1–5. DOI: 10.1109/ICCW.2009.5208049.

[225] Lakshmi V. Thanayankizil, Aravind Kailas, and Mary Ann Ingram. "Opportunistic Large Array Concentric Routing Algorithm (OLACRA) for Upstream Routing in Wireless Sensor Networks." In: *Ad Hoc Networks* 9.7 (Sept. 2011), pp. 1140–1153. ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2010.12.004.

[226] Ozan K. Tonguz, Nawaporn Wisitpongphan, Jayendra S. Parikh, Fan Bai, Priyantha Mudalige, and Varsha K. Sadekar. "On the Broadcast Storm Problem in Ad Hoc Wireless Networks." In: *3rd International Conference on Broadband Communications, Networks and Systems (BROADNETS)*. San Jose, California, USA: IEEE, Oct. 2006, pp. 1–11. DOI: 10.1109/BROADNETS.2006.4374403.

[227] Yung-Szu Tu and Gregory J. Pottie. "Coherent Cooperative Transmission From Multiple Adjacent Antennas To a Distant Stationary Antenna Through AWGN Channels." In: *55th IEEE Vehicular Technology Conference (VTC Spring)*. Vol. 1. Birmingham, Alabama, USA: IEEE, May 2002, pp. 130–134. DOI: 10.1109/VTC.2002.1002678.

[228] Donald W. Tufts and Ramdas Kumaresan. "Estimation of Frequencies of Multiple Sinusoids: Making Linear Prediction Perform Like Maximum Likelihood." In: *Proceedings of the IEEE* 70.9 (Sept. 1982), pp. 975–989. ISSN: 1558-2256. DOI: 10.1109/PROC.1982.12428.

[229] *Universal Software Radio Peripheral*. URL: https://www.ettus.com/products.

[230] Athena Vakali and George Pallis. "Content Delivery Networks: Status and Trends." In: *IEEE Internet Computing* 7.6 (Nov. 2003), pp. 68–74. ISSN: 1941-0131. DOI: 10.1109/MIC.2003.1250586.

[231] Mahesh K. Varanasi and Behnaam Aazhang. "Multistage Detection in Asynchronous Code-Division Multiple-Access Communications." In: *IEEE Transactions on Communications* 38.4 (Apr. 1990), pp. 509–519. ISSN: 1558-0857. DOI: 10.1109/26.52662.

[232] Daniele Veronesi and Dennis L. Goeckel. "Multiple Frequency Offset Compensation in Cooperative Wireless Systems." In: *IEEE Global Telecommunications Conference (GLOBECOM)*. San Francisco, California, USA: IEEE, Nov. 2006, pp. 1–5. DOI: 10.1109/GLOCOM.2006.120.

[233] Andrew J. Viterbi. "Very Low Rate Convolutional Codes for Maximum Theoretical Performance of Spread-Spectrum Multiple-Access Channels." In: *IEEE Journal on Selected Areas in Communications* 8.4 (May 1990), pp. 641–649. ISSN: 1558-0008. DOI: 10.1109/49.54460.

[234] Feng Wang and Mary Ann Ingram. "A Practical Equalizer for Cooperative Delay Diversity with Multiple Carrier Frequency Offsets." In: *IEEE International Conference on Communications (ICC)*. Ottawa, Ontario, Canada: IEEE, June 2012, pp. 4100–4104. DOI: 10.1109/ICC.2012.6364069.

[235] Xiaodong Wang and H. Vincent Poor. "Iterative (Turbo) Soft Interference Cancellation and Decoding for Coded CDMA." In: *IEEE Transactions on Communications* 47.7 (July 1999), pp. 1046–1061. ISSN: 1558-0857. DOI: 10.1109/26.774855.

[236] Yin Wang, Yuan He, Dapeng Cheng, Yunhao Liu, and Xiang-yang Li. "TriggerCast: Enabling Wireless Constructive Collisions." In: *32nd IEEE International Conference on Computer Communications (INFOCOM)*. Turin, Italy: IEEE, Apr. 2013, pp. 480–484. DOI: 10.1109/INFCOM.2013.6566819.

[237] Yin Wang, Yuan He, Xufei Mao, Yunhao Liu, and Xiang-yang Li. "Exploiting Constructive Interference for Scalable Flooding in Wireless Networks." In: *IEEE/ACM Transactions on Networking* 21.6 (Dec. 2013), pp. 1880–1889. ISSN: 1063-6692. DOI: 10.1109/TNET.2013.2238951.

[238] Christopher Ware, Joe Chicharo, and Tadeusz Wysocki. "Simulation of Capture Behaviour in IEEE 802.11 Radio Modems." In: *54th IEEE Vehicular Technology Conference (VTC Fall)*. Vol. 3. Atlantic City, New Jersey, USA: IEEE, Oct. 2001, pp. 1393–1397. DOI: 10.1109/VTC.2001.956425.

[239] Christopher Ware, John Judge, Joe Chicharo, and Eryk Dutkiewicz. "Unfairness and Capture Behaviour in 802.11 Adhoc Networks." In: *IEEE International Conference on Communications (ICC)*. Vol. 1. New Orleans, Louisiana, USA: IEEE, June 2000, pp. 159–163. DOI: 10.1109/ICC.2000.853084.

[240] Christopher Ware, Tadeusz Wysocki, and Joe Chicharo. "Hidden Terminal Jamming Problems in IEEE 802.11 Mobile Ad Hoc Networks." In: *IEEE International Conference on Communications (ICC)*. Vol. 1. Helsinki, Finland: IEEE, June 2001, pp. 261–265. DOI: 10.1109/ICC.2001.936314.

[241] *Website of tcpdump, a powerful command-line packet analyzer, and libpcap, a portable C/C++ library for network traffic capture*. Mar. 2022. URL: https://www.tcpdump.org.

[242] *Website of Wireshark*. Mar. 2022. URL: https://www.wireshark.org.

[243] Stephen B. Weinstein. "The History of Orthogonal Frequency-Division Multiplexing [History of Communications]." In: *IEEE Communications Magazine* 47.11 (Nov. 2009), pp. 26–35. ISSN: 1558-1896. DOI: 10.1109/MCOM.2009.5307460.

[244] Geoffrey Werner-Allen, Geetika Tewari, Ankit Patel, Matt Welsh, and Radhika Nagpal. "Firefly-Inspired Sensor Network Synchronicity with Realistic Radio Effects." In: *3rd ACM Conference on Embedded Networked Sensor Systems (SenSys)*. San Diego, California, USA: ACM, Nov. 2005, pp. 142–153. DOI: 10.1145/1098918.1098934.

[245] Kamin Whitehouse, Alec Woo, Fred Jiang, Joseph Polastre, and David Culler. "Exploiting the Capture Effect for Collision Detection and Recovery." In: *2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*. Sydney, New South Wales, Australia: IEEE, May 2005, pp. 45–52. DOI: 10.1109/EMNETS.2005.1469098.

[246] *Wireless Open Access Research Platform*. URL: http://warpproject.org.

[247] *Wireless Open Access Research Platform: 802.11 Reference Design for WARP v3*. URL: https://warpproject.org/trac/wiki/802.11.

[248] *Wireless Open Access Research Platform: Clocking*. URL: https://warpproject.org/trac/wiki/HardwareUsersGuides/WARPv3/Clocking.

[249] *Wireless Open Access Research Platform: MMCX Clock Module*. URL: https://warpproject.org/trac/wiki/HardwareUsersGuides/CM-MMCX.

[250] *Wireless Open Access Research Platform: WARPLab*. URL: https://warpproject.org/trac/wiki/WARPLab.

[251] Wenxin Xu and Graham Campbell. "A Near Perfect Stable Random Access Protocol for a Broadcast Channel." In: *IEEE International Conference on Communications (ICC) / SUPERCOMM – Discovering a New World of Communications*. Vol. 1. Chicago, Illinois, USA: IEEE, June 1992, pp. 370–374. DOI: 10.1109/ICC.1992.268230.

[252] Qing Yang, Xiaoxiao Li, Hongyi Yao, Ji Fang, Kun Tan, Wenjun Hu, Jiansong Zhang, and Yongguang Zhang. "BigStation: Enabling Scalable Real-Time Signal Processing in Large MU-MIMO Systems." In: *ACM SIGCOMM Computer Communication Review* 43.4 (Aug. 2013), pp. 399–410. ISSN: 0146-4833. DOI: 10.1145/2534169.2486016.

[253] Yang Yang and Ness B. Shroff. "Scheduling in Wireless Networks with Full-Duplex Cut-Through Transmission." In: *34th IEEE International Conference on Computer Communications (INFOCOM)*. Hong Kong, China: IEEE, Apr. 2015, pp. 2164–2172. DOI: 10.1109/INFOCOM.2015.7218602.

[254] Vivek Yenamandra and Kannan Srinivasan. "Vidyut: Exploiting Power Line Infrastructure for Enterprise Wireless Networks." In: *ACM Special Interest Group on Data Communication (SIGCOMM) Conference*. Chicago, Illinois, USA: ACM, Aug. 2014, pp. 595–606. DOI: 10.1145/2619239.2626329.

[255] Kasim Sinan Yildirim and Aylin Kantarci. "Time Synchronization Based on Slow-Flooding in Wireless Sensor Networks." In: *IEEE Transactions on Parallel and Distributed Systems* 25.1 (Jan. 2014), pp. 244–253. ISSN: 1558-2183. DOI: 10.1109/TPDS.2013.40.

[256] Ali Özgür Yılmaz. "Cooperative Diversity in Carrier Frequency Offset." In: *IEEE Communications Letters* 11.4 (Apr. 2007), pp. 307–309. ISSN: 1558-2558. DOI: 10.1109/LCOM.2007.348283.

[257] Dingwen Yuan. "Improving Quality of Service in Wireless Sensor Networks for Industrial Automation." PhD thesis. Technische Universität Darmstadt, June 2016. URL: https://tuprints.ulb.tu-darmstadt.de/5508.

[258] Dingwen Yuan and Matthias Hollick. "Let's Talk Together: Understanding Concurrent Transmission in Wireless Sensor Networks." In: *38th IEEE Conference on Local Computer Networks (LCN)*. Sydney, New South Wales, Australia: IEEE, Oct. 2013, pp. 219–227. DOI: 10.1109/LCN.2013.6761237.

[259] Dingwen Yuan and Matthias Hollick. "Ripple: High-Throughput, Reliable and Energy-Efficient Network Flooding in Wireless Sensor Networks." In: *16th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. Boston, Massachusetts, USA: IEEE, June 2015, pp. 1–9. DOI: 10.1109/WoWMoM.2015.7158133.

[260] Dingwen Yuan, Michael Riecker, and Matthias Hollick. "Making 'Glossy' Networks Sparkle: Exploiting Concurrent Transmissions for Energy Efficient, Reliable, Ultra-Low Latency Communication in Wireless Control Networks." In: *11th European Conference on Wireless Sensor Networks (EWSN)*. Ed. by Bhaskar Krishnamachari, Amy L. Murphy, and Niki Trigoni. Oxford, England, UK: Springer, Feb. 2014, pp. 133–149. DOI: 10.1007/978-3-319-04651-8_9.

[261] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. "A Survey of Autonomous Driving: Common Practices and Emerging Technologies." In: *IEEE Access* 8 (Apr. 2020), pp. 58443–58469. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2983149.

[262] Sherali Zeadally, Muhammad Awais Javed, and Elyes Ben Hamida. "Vehicular Communications for ITS: Standardization and Challenges." In: *IEEE Communications Standards Magazine* 4.1 (Mar. 2020), pp. 11–17. ISSN: 2471-2833. DOI: 10.1109/MCOMSTD.001.1900044.

[263] Allert van Zelst and Tim C. W. Schenk. "Implementation of a MIMO OFDM-Based Wireless LAN System." In: *IEEE Transactions on Signal Processing* 52.2 (Feb. 2004), pp. 483–494. ISSN: 1941-0476. DOI: 10.1109/TSP.2003.820989.

[264] Huacheng Zeng, Hongxiang Li, and Qiben Yan. "Uplink MU-MIMO in Asynchronous Wireless LANs." In: *19th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. Los Angeles, California, USA: ACM, June 2018, pp. 21–30. DOI: 10.1145/3209582.3209585.

[265] Eustathia Ziouva and Theodore Antonakopoulos. "CSMA/CA Performance Under High Traffic Conditions: Throughput and Delay Analysis." In: *Computer Communications* 25.3 (Feb. 2002), pp. 313–321. ISSN: 0140-3664. DOI: 10.1016/S0140-3664(01)00369-3.

[266] Rui Zou, Zishan Liu, Lin Zhang, and Muhammad Kamil. "A Near Collision Free Reservation Based MAC Protocol for VANETs." In: *IEEE Wireless Communications and Networking Conference (WCNC)*. Istanbul, Turkey: IEEE, Apr. 2014, pp. 1538–1543. DOI: 10.1109/WCNC.2014.6952438.

ERKLÄRUNG ZUR DISSERTATIONSSCHRIFT

gemäß § 9 der Allgemeinen Bestimmungen der Promotionsordnung der Technischen Universität Darmstadt vom 12. Januar 1990 (ABI. 1990, S. 658) in der Fassung der 8. Novelle vom 1. März 2018

Hiermit versichere ich, Robin Klose, die vorliegende Dissertationsschrift ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht. Eigenzitate aus vorausgehenden wissenschaftlichen Veröffentlichungen sowie die Urheberschaften der einzelnen Beiträge sind in Anlehnung an die Hinweise des Promotionsausschusses des Fachbereichs Informatik zum Thema „Kumulative Dissertation und Eigenzitate in Dissertationen“ (CR; 01.12.2022) im Kapitel „*Collaborations and My Contributions*“ auf den Seiten xxv bis xxvi angegeben. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. In der abgegebenen Dissertationsschrift stimmen die schriftliche und die elektronische Fassung überein.

Darmstadt, 19. Dezember 2022

Robin Klose