Designing and Implementing Cognitive Medium Access Control Protocols, a Hierarchical Approach

Von der Fakultät für Elektrotechnik und Informationstechnik der
Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines Doktors der
Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Peng Wang, M.Sc.
aus Hunan, China

Berichter: Prof. Dr.-Ing. Marina Petrova
Prof. Ilenia Tinnirello

Tag der mündlichen Prüfung: 19. November 2018

Diese Dissertation ist auf den Internetseiten
der Universitätsbibliothek online verfügbar.

Abstract

With the explosive growth of mobile devices, industrial automation and the Internet of things, the need for high speed and reliable wireless communication systems become greater than ever. As a consequence, the issue of spectrum scarcity becomes a severe problem in conventional frequency bands. To handle the crowded radio environments and make full use of quickly emerged advanced technologies, Medium Access Control (MAC) protocols are also evolving rapidly. On one hand, MAC layer schemes inevitably turn to be smart to efficiently utilize the limited spectrum resources. On the other hand, many new communication systems are designed and deployed to perform in so far unconventional frequencies, for example on mm-Wave bands where a tremendous amount of spectrum exists. However, the signal propagation and channel conditions in the new environments are different to the conventional scenarios, which also requires distinguishing protocol design. Therefore, rethinking of the MAC designing philosophy for the new generation of communication systems becomes a necessity.

In this dissertation, we first focus on designing and implementing directional MAC protocols for mm-Wave communications. We analyze traditional MAC schemes such as CSMA/CA, and design new schemes to offer reliable and fast wireless services in directional networks. Cooperation among communicating nodes is considered in the new schemes to achieve network level improvement. Through both numerical and empirical analysis, we show that the newly proposed protocols can handle complex radio environments and achieve considerable improvements against conventional methods. We also design and implement cognitive MAC protocol targeted to decentralized multichannel environment which allows parallel transmission for secondary users while minimizing the interruption to the primary user. The proposed protocol is implemented using software defined radios. Our experimental performance evaluation indicates that the proposed scheme is able to carry out reliable parallel communications in multiple channels even in a severely interfered environment.

In addition to the newly designed MAC schemes, we propose a new MAC designing tool named Decomposite MAC Description Language (DMDL), which aims at providing a unified framework for easier MAC protocol design and implementation. In DMDL, a MAC protocol is conceptually modeled as a hierarchical finite state machine. MAC functionalities are decomposed by comprehensively reviewing existing MAC schemes and predicting future demands. DMDL achieves high decomposability and modularity, which enables it to be a pure graphical language. To show its capacity and performance, DMDL is completely implemented in GNU Radio and operated on USRPs. Several traditional wireless and cognitive MAC protocols are implemented and tested. The empirical results indicate that the performances of the implemented MAC protocols are in accordance with the theoretical expectations. Combing with its simple and straightforward working flow,

DMDL is proven to be an efficient MAC designing and implementing tool with high flexibility, reusability, and debuggability.

Kurzfassung

Mit der explosiven Zunahme an mobilen Gerten, der industriellen Automation und dem Internet der Dinge ist der Bedarf an schneller und verlsslicher kabelloser Kommunikation grer denn je. Als Folge dessen wird die Knappheit des konventionellen Frequenzspektrums zu einem groen Problem. Um die berfllte Radioumgebung zu kontrollieren und um neue, sich rasch entwickelnde, fortschrittliche Technologien besser nutzen zu knnen, findet bei den Medium Access Control (MAC) Protokollen ebenfalls eine schnelle Entwicklung statt. Auf der einen Seite werden MAC Protokolle unweigerlich intelligenter, um die limitierten Ressourcen effizienter nutzen zu knnen. Auf der anderen Seite werden viele neue Kommunikationssysteme entwickelt und aufgebaut, die in bislang ungenutzten Frequenzbndern, wie zum Beispiel dem mm-Wave Band, in dem eine groe Menge nicht genutzten Spektrums existiert, operieren. Die Signalausbreitung und der Kanal verhalten sich in diesen neuen Umgebungen jedoch anders als in den konventionellen Frequenzbndern, was ein angepasstes Protokolldesign erfordert. Deshalb ist es erforderlich, den Designprozess von MAC Protokollen fr die kommenden Generationen der mobilen Kommunikation zu berdenken.

In dieser Arbeit konzentrieren wir uns zunchst auf das Design von direktionalen MAC Protokollen fr den Einsatz in der mm-Wave Kommunikation. Wir analysieren traditionelle MAC Protokolle - wie CSMA/CA - und entwickeln neue Protokolle, um zuverlssige und schnelle Kommunikation in direktionalen Netzwerken zu gewhrleisten. Die Kooperation zwischen den Kommunikationspartnern wird in den neuen Protokollen bercksichtigt, um Verbesserungen auf der Netzwerkebene zu erreichen. Wir zeigen sowohl durch numerische als auch durch empirische Analysen, dass die vorgeschlagenen Protokolle komplexe Radioumgebungen handhaben knnen und eine erhebliche Leistungssteigerung gegenber den konventionellen Protokollen erreichen. Wir entwerfen weiterhin ein kognitives MAC Protokoll fr den Einsatz in dezentralisierten Mehrkanalumgebungen, welches parallele bertragungen fr die sekundren Nutzer ermglicht und gleichzeitig die Unterbrechungen des primren Nutzers minimiert. Das vorgeschlagene Protokoll wird mithilfe von Software Defined Radios implementiert. Unsere experimentelle Leistungsauswertung zeigt, dass das vorgeschlagene Verfahren zuverlssige parallele bertragungen in mehreren Kanlen ermglicht, selbst dann, wenn betrchtliche Interferenz vorhanden ist.

Zustzlich zu den hier vorgestellten neuen MAC Protokollen stellen wir auerdem ein neues Werkzeug zum Entwurf von MAC Protokollen vor, welches wir Decomposite MAC Description Language (DMDL) nennen. Das Ziel von DMDL ist es, ein vereinheitlichtes Rahmenwerk zum Entwurf und zur Implementierung von neuen MAC Protokollen zur Verfgen zu stellen. In DMDL wird ein MAC Protokoll durch eine hierarchische finite Zustandsmaschine dargestellt. MAC Funktionen werden durch eine genaue Analyse existierender Protokolle und durch die Abschtzung zuknftiger Anforderungen in ihre Grundbestandteile zerlegt. DMDL

erreicht einen hohen Grad an Fragmentierung und Modularitt, wodurch eine rein grafische Umsetzung mglich ist. Um die Effektivitt und Effizienz von DMDL zu demonstrieren, wurde es mithilfe von GNU Radio implementiert und unter Verwendung von USRPs getestet. Verschiedene traditionelle und kognitive MAC Protokolle werden implementiert und getestet. Die empirischen Resultate zeigen, dass die Leistung der implementierten Protokolle mit der theoretisch zu erwartenden Leistung bereinstimmt. Diese Ergebnisse, zusammen mit der einfachen Handhabung, zeigen, dass DMDL ein effektives Werkzeug zum Entwurf und zur Analyse von MAC Protokollen ist, welches einen hohen Grad an Flexibilitt, Wiederverwertbarkeit und Wartbarkeit aufweist.

Acknowledgments

The path towards Ph.D. is long and twisty. It is impossible to complete without the tremendous help and advice from people around me. For that, I would like to express the deepest appreciation to Professor Marina Petrova for the continuous support of my Ph.D. study and research, for her patience, motivation, and immense knowledge. She has guided me and encouraged me to carry on through these years and has contributed to this thesis with a major impact. Without her guidance and persistent help, the path would be a dead end.

I would like to express my sincere gratitude to Professor Petri Mähönen, who continually and convincingly conveyed a spirit of adventure in regards to research. His enthusiastic encouragement, useful critiques, and patient guidance have made the path a thoughtful and rewarding journey.

I would also like to thank professor Ilenia Tinnirello, for her willingness in been my second examiner. Our discussion and her feedback are very fruitful to my research and future career.

I am thankful to all my colleagues at INETS for the international, enjoyable and friendly working atmosphere. As the old Chinese saying goes, when three men meet together, one of them who is anxious to learn can always learn something of the other two. I am deeply grateful to Nikos Perpinias, Dr. Andrea Munari, Dr. Ljiljana Simić, Dr.-Ing. Janne Riihijärvi, Dr.-Ing. Andreas Achtzehn, Andra Voicu, and Fei Liu. I would also express special appreciation to Dr.-Ing. Junaid Ansari, who enlightened me at the beginning of my research and helped me throughout my Ph.D. study. The sleepless nights we were working together and the fun we have had in the institute and after work will be my best memories in the past seven years. In addition, I am deeply grateful to Gero Schmidt-Kärst, Lise-Lotte Birenius Schneider, and Barbara Hillebrand, who made me feel I have already gotten a home in the place that is thousands of kilometers far away from the place I was born.

A very special word of thanks goes for my parents, who have been great over the years and never raised an eyebrow when I told them I do not know how many years are needed to become a doctor. They supported me spiritually throughout the whole journey and my life in general.

Finally, I want to thank Dr. rer. nat. Cheng Wu, my beloved wife. She never lost faith in me, even when I was in doubt. She is a role model in hard working but also constantly showed me how to balance work and life. An independent and wise lady is a man's greatest treasure, not only in the way towards Ph.D. but also for the entire life.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Contribution 3

1.2.1 Directional MAC protocols 3

1.2.2 Multichannel Cognitive MAC protocol design and implementation 4

1.2.3 Decomposite MAC Description Language 4

1.3 Outline 5

2 Directional MAC protocols **6**

2.1 Related Work 6

2.1.1 Conventional Directional MAC protocols 6

2.1.2 mm-Wave Communication System 8

2.1.3 IEEE 802.11ad standard 8

2.1.4 Frame aggregation in IEEE 802.11ad standard 10

2.2 Network Topology and Performance 12

2.3 Cooperative Design in Uncoordinated Directional Networks 14

2.3.1 Carrier Sensing 14

2.3.2 Topology Analysis 18

2.3.3 Protocol Details 20

2.3.4 Mobility 22

2.3.5 Simulation Results 22

2.3.6 Summary 27

2.4 Spatial Reuse in Centralized Directional Networks 27

2.4.1 Spatial Buffering 28

2.4.2 DSTA Selection 31

2.4.3 Co-Channel Interference Measurements 31

2.4.4 NDSTA Grouping 32

2.4.5 Hardware Verification 33
2.4.6 Simulation Results 35
2.4.7 Summary 40
2.5 Directional Frame Aggregation 41
2.5.1 Interference analysis 42
2.5.2 CTMAC principle 45
2.5.3 Adaptive frame aggregation 46
2.5.4 Simulation Results 46
2.5.5 Summary 49

3 Multichannel Cognitive MAC protocol 50
3.1 Related Work 51
3.2 CogMAC+ Design 52
3.2.1 Multichannel Preamble Reservation 53
3.2.2 Multichannel Multiframe Transmission 56
3.2.3 Multichannel Virtual Carrier Sensing 60
3.2.4 Noise Floor Estimation and Adaptive Energy Detection Threshold 62
3.2.5 Channel Weighting and Dynamic Channel Selection 67
3.2.6 CCA Based Random Backoff Scheme 68
3.3 Protocol Implementation 69
3.4 Experimental Result 70
3.4.1 Baseline Comparison 70
3.4.2 CogMAC+ Enhancement 72
3.4.3 Performance Evaluation of Secondary Usage 75
3.5 Summary 78

4 Decomposite MAC Description Language 80
4.1 Introduction 80
4.2 Related Work 81
4.2.1 MAC Design Tool 81
4.3 MAC Models 82

4.3.1 Reactive Systems and HFSM 83
4.3.2 Modeling MACs with HFSM 84
4.3.3 Requirements Analysis 86
4.3.4 MAC Representation 88
4.4 Architecture 90
4.4.1 *Command* 90
4.4.2 Blocks 91
4.5 Examples 104
4.5.1 Time limited Poisson Source 104
4.5.2 *p*-persistent CSMA 104
4.5.3 IEEE 802.11 DCF 105
4.5.4 Discussion on FSM and SDF 108
4.6 Summary 108

5 DMDL Implementation in GNU Radio 110
5.1 Blocks 110
5.1.1 Timing related blocks 110
5.1.2 Frame related blocks 112
5.1.3 Interface blocks 120
5.1.4 Auxiliary blocks 128
5.2 Signaling System 130
5.3 Performance Evaluation on Single Frame Transmission 132
5.3.1 Frame Waveform 133
5.3.2 Transmission and Reception Switch System 135
5.3.3 Timing in Single Frame Transmission 135
5.3.4 Timing in Handshake 140
5.4 Performance Evaluation on MAC Protocols implemented with GR-DMDL 145
5.4.1 Pure ALOHA 145
5.4.2 Slotted ALOHA 148
5.4.3 *p*-persistent CSMA 151
5.4.4 IEEE 802.11 DCF 154
5.4.5 CogMAC 174
5.5 Summary 177

6 Conclusions and Future Work **180**

6.1 Summary of the Key Results 180

6.2 Future Work 182

Acronyms **184**

List of Figures **189**

List of Tables **199**

Bibliography **201**

Eigene Verffentlichungen **211**

Journalverffentlichungen (mit peer-review) 211

Verffentlichungen in Proceedings internationaler Konferenzen (mit review) 211

Curriculum Vitae **213**

1 Introduction

This dissertation focuses on designing novel Medium Access Control (MAC) protocols targeted to offer reliable and efficient wireless services for various environments and application scenarios, and also providing a module based MAC designing and prototyping tool for MAC protocol development. Section 1.1 represents the motivation of this thesis work. In Section 1.2, the contributions are categorized and listed. In Section 1.3, the outline of this thesis is summarized.

1.1 Motivation

Resulting from the recent advances in analog and Radio Frequency (RF) Integrated Circuit (IC) design, sensors and processor technologies, developers gain massive resource to design unprecedentedly smart and complex communication systems. Besides, the constantly increasing demand for throughput and quality requires the new communication systems to either find new radio resources and be aware of the surrounding environments more than ever before. However, the newly deployed frequency bands have distinguished channel characteristics to the traditional ones which may need new MAC layer channel access methods. For example, the high attenuation of the mm-Wave propagation, in particular in the 60 GHz band, makes the directional transmission an essential part of the system, which is not addressed by the traditional protocols working on the conventional microwave (sub 6 GHz) bands. Moreover, the co-existence of multiple wireless networks on the limited frequency resources also outdates the traditional MAC paradigm due to its ignorance of the heterogeneous interference and its simplicity in environmental awareness, information sharing, and learning abilities [13]. To better utilize the rapid improvements in technologies and adapt to the new environments, both the MAC protocols and the way of designing MAC protocols should be rethought.

In order to perform high data rate and reliable communications, it is a new trend to develop communication systems which are targeted to operate at the so-far unconventional frequency bands, especially in mm-Wave bands where exist a huge amount of unlicensed frequency resources, e.g., 60 GHz bands. Moving from sub 6 GHz channels to mm-Wave bands where multiple gigabit unlicensed channel exist, has a great potential to upraise data rate and quality of wireless communications. Recently, the reduced costs of the production of mm-Wave communication devices makes transmission over such high frequency bands more applicable. However, large signal attenuation exists at 60 GHz band caused by oxygen absorption and short wavelength [46, 51, 58, 87, 91, 92, 106]. The significant differences of signal propagation characteristics between 60 GHz bands and the sub 6 GHz bands severely limits the usage of the conventional WI-FI principles for 60 GHz applications. To overcome the high pathloss, mm-Wave communications rely on

directional transmissions. However, deploying directional transmission technologies also introduces some additional problems. For example, the blind problem takes place if directional-only nodes are deployed in the network. It may happen that two nodes are in their transmission range but they cannot hear each other due to mismatched antenna beams. Receiving in omni-directional mode can partially avoid the blind problem, however, other problems such as asymmetric-in-gain problem and unheard Request To Send (RTS)/Clear To Send (CTS) problem may emerge, which make the performance of wireless directional networks even worse. Therefore, the principles and basic methods used in mm-Wave directional system need to be completely redesigned to match the new radio environments.

Despite of using unconventional spectrum, the new emergencing communication technologies can also improve the system performance in conventional frequency bands. Due to the increasing number of applications and devices operating in the conventional frequency bands, in-band interference becomes one of the greatest problems that new MAC protocols have to address. Multichannel technologies can be used to avoid collision by allowing nodes to jump to empty channel if the current one is in use. This concept can be used in cognitive radio [11, 17, 42, 43] to enable secondary users to switch channels to dodge the primary users and enhance the performance by opportunistically access the spectrum hole. New problems have emerged in the new multichannel environments, e.g., the rendezvous problem that nodes may lose synchronization when their are jumping among channels uncoordinatedly. To solve this problem with affordable cost, CogMAC [48] was proposed to achieve reliable data communication by using the multichannel carrier sensing principle. It requires transmitting nodes to repetitively transmit the same frame back-to-back and idle nodes to continuously sense all channels. If the back-to-back transmissions last long enough, all idle nodes are able to receive that frame in any sensing order. CogMAC points out an potential way to solve the rendezvous problem for secondary usage in multichannel radio environment. However, because of the lack of detailed timing control, transmission scheduling, and cross layer optimization, there is still a huge room for improvement.

Due to the rapidly increasing demands on wireless communications and new emerged technologies, new protocols will be designed, tested, and standardized much faster than ever before. To enable fast MAC protocol design and also provide flexibility and reconfigurability, recently, module design has drawn significant attention [32, 47, 105, 110]. It follows the objective oriented principle which first decomposes MAC protocol into elementary autonomous components and then implements them. The module design principle has several benefits. First, the modularity decouples the whole MAC protocol into separated components, which simplifies the testing procedure and makes debugging easy. Second, it enables fast reconfiguration during implementation and optimization because the new protocols are built from well maintained components. Module design also enables visualization of MAC protocol because of its inherent graphical nature. A number of MAC designing tools [1,47,75,105] are aimed at providing graphical environment for communication system design, implementation, and even execution. In contrast to the traditional text based environments, the visualization makes the protocol design easier and

more straightforward. Even through the module based MAC design has widely used in many years, to the best of our knowledge, a comprehensive and detailed analysis on how to model a MAC protocol and how to optimize MAC decomposition is still missing. In addition, no deep study has been carried out to investigate how to visualize module designed based MAC protocols.

1.2 Contribution

The major contributions of this dissertation are as follows.

1. To enable efficient and reliable communication over mm-Wave channels, we proposed a series of directional MAC protocols to address the new features of mm-Wave technologies and wireless environments. Many traditional MAC schemes including the carrier sensing and handshake are revolutionarily redesigned to fit the new requirements. Due to the directivity, the importance of topology awareness is highlighted in the proposed protocols which are handled by the massively added information sharing and the topology recognition mechanisms. We built testbeds from various hardware platforms including Software Defined Radios (SDRs) for verification and measurements.

2. We propose a cognitive MAC protocol which allows the secondary users to perform seamless transmission in decentralized multichannel environment where the rendezvous problem exists. It also enables the spectrum coexistence of multiple secondary networks. We implement the protocol on a SDR platform and carry out many long term experiments for evaluation.

3. We propose a module based MAC design language which is able to carry the whole develop circle of MAC protocols, from design to execution. To show the practicality, we implement the language in GNU Radio and publicly release the source code. We implement a number of widely used MAC protocols using the language to show the accuracy and simplicity.

Most of the proposed protocols are implemented and tested on hardware-based platforms, including various SDRs such as USRPs. The empirical results show that the proposed MAC schemes and tools can effectively improve the network performance, stability and robustness. The detailed introduction of each work is listed below.

1.2.1 Directional MAC protocols

We proposed a series of directional MAC protocols in the dissertation. Due to the high pathloss and attenuation in mm-Wave bands, directional communication technologies are inherently adopted to enlarge communication range and guarantee the Signal to Interference and Noise Radio (SINR). In Cooperative-Directional

MAC (CoopDMAC) [84], the MAC challenges and opportunities of using directional transmission technology are discussed. Traditional technologies and algorithms in omni-directional networks, such as carrier sensing and handshaking are tailored to fit the new features. The potential of increasing overall network throughput by improving spatial reuse is discussed. By utilizing the advantages of parallel directional transmissions, we propose Spatial Buffering MAC (SBMAC) to enhance the network performance by allowing the nodes with low speed links to the center node to buffer their data to the nodes which have better links to the center node. The frame aggregation algorithm is also reviewed when directional transmission technology is involved. The proposed protocol, we propose Cross Talk MAC (CTMAC) [81] to further improve the single link performance by differentiate the size and transmission time of aggregated frames. Simulation results shows that all proposed MAC protocols outperform the conventional methods by more effectively utilizing the benefits of directional transmission technologies.

1.2.2 Multichannel Cognitive MAC protocol design and implementation

To address the rendezvous problem in the decentralized multichannel environment, we propose a cognitive MAC protocol named CogMAC+ [82, 83]. It is based on multichannel preamble reservation scheme and Carrier Sensing Multiple Access (CSMA) principle. CogMAC+ can let secondary users carry out parallel transmissions in different channels by enabling them to estimate the duration of channel occupancy of each other. In order to achieve the optimal sensing schedule, it allows secondary users to dynamically access channels according to the channel occupancy of primary users and other secondary networks. Moreover, CogMAC+ uses an adaptive energy detection scheme to dynamically set the energy detection threshold according to the carrier sensing status, the false positive detection ratio and the estimated noise level. CogMAC+ has been verified and evaluated empirically in a SDR testbed with realistic spectrum occupancies of licensed users. The empirical results indicates that CogMAC+ is able to carry out reliable parallel communication in multiple channels even in a severely interfered environment.

1.2.3 Decomposite MAC Description Language

Decomposite MAC Description Language (DMDL) is developed to be a unified MAC language for designing and implementing MAC protocols [85]. In DMDL, a MAC protocol is defined as a Synchronous Data Flow (SDF) which follows the Hierarchical Finite State Machine (HFSM) architecture. After analyzing the HFSM properties and common MAC functionalities in numerous types of MAC protocols and predicting future demands, many MAC components are decomposed and visualized, which are called blocks in DMDL. DMDL represents and implements a MAC protocol by building a directed graph, in which the blocks are the vertices that are connected by directed arcs. The control signal called *command* is defined to be passed among block via the directed arcs and convey both control information

and data. To verify the design of DMDL, it is implemented in GNU Radio with a complete PHY layer implementation and connected the real SDR hardware. An all-round calibration and based line tests are carried out to give a comprehensive understanding to the language implementation. A number of MAC protocols, from the simplest ALOHA to the CogMAC+ which is a cognitive MAC protocol with high complexity, are implemented and tested using the new language. The empirical results show that the protocols implemented by the proposed language agree well with the theory.

1.3 Outline

The dissertation is organized as follows. In Chapter 2, existing directional MAC protocols and standards are first reviewed and discussed. To provide a comprehensive background introduction to the proposed MAC schemes, the targeted application scenarios, i.e., mm-Wave channel and communication technologies are also briefly introduced. In the remaining of the chapter, CoopDMAC is first given to generally discuss the methodology of designing reliable and efficient cooperative MAC protocol in distributed directional wireless networks. After illustrating the general solution, SBMAC is proposed to show how to improve the Institute of Electrical and Electronics Engineers (IEEE) 802.11ad standard by utilizing the principles discussed in CoopDMAC. CTMAC is introduced in the end of this chapter to show how to improve the performance of frame aggregation in directional wireless networks. Chapter 3 presents CogMAC+, as well as the comparison with its precedent CogMAC. First we discuss the new features and updates in CogMAC+ which help the protocol to outperform the former, and then show the SDR implementation and experimental results. CogMAC+ is implemented in LabVIEW but CogMAC is implemented in Wireless Open Access Research Platform (WARP). Therefore a baseline comparison is given to verify the new implementation. Empirical data from the measurement campaign is shown in the end of this chapter. Chapter 4 describes the idea of DMDL and elaborate the rules of protocol decomposition in detail. Chapter 5 gives an example of DMDL implementation on GNU Radio and shows many implementations of classic protocols using DMDL. Empirical data are shown in the end of this chapter. The dissertation is concluded in Chapter 6.

2 Directional MAC protocols

Directional antennas in wireless networks can provide substantial performance gain and opportunities. In the case of high-frequency systems, such as emerging 60 GHz networks, they are inherent feature of the system design. In contrast to the traditional omni-directional antennas, directional antennas are capable of reducing the interference level, increasing the communication range, and improving spatial reuse. However, new challenges emerge especially at the MAC layer with the deployment of the directional transmission technologies, such as the deafness problem and the asymmetric-in-gain problem. In this chapter, existing MAC protocols and industrial standards using directional transmission technologies are first reviewed. By revisiting the conventional directional MAC protocols and standards, the MAC challenges and opportunities of using directional transmission technology are concluded and further analyzed in Section 2.3. We propose Cooperative Directional MAC (Coop-DMAC) to improve the network performance in terms of throughput and latency. The potential of using directional transmission technologies to increase overall network throughput by improving spatial reuse is discussed in Section 2.4.1. By utilizing the advantages of parallel directional transmissions, We propose Spatial Buffering MAC (SBMAC) to enhance the network performance by allowing the nodes have only low speed links to buffer their data to the nodes which have high speed links. In Section 2.5, We propose Cross Talk MAC (CTMAC) [81] to further enhance the single link throughput when frame aggregation is used.

2.1 Related Work

Directional MAC protocols have been studied for a long time in context of sub 6 GHz bands. With the emerging mm-Wave technologies, directional MAC protocol design has recently drawn high attention again. In this section, we review both the conventional MAC protocol design and the new mm-Wave MACs. Due to the totally different spectrum and propagation characteristics, the MAC protocol designs also show apparent differences. In the end of the section, the IEEE 802.11ad standard is investigated to show the current progress on standardization of directional MACs.

2.1.1 Conventional Directional MAC protocols

A majority of directional MAC protocols [23, 39, 77, 88, 89, 114] rely on Directional Virtual Carrier Sensing (DVCS) and Directional Network Allocation Vector (DNAV) [68] schemes to avoid collision. DVCS and DNAV allow a node to sense the channel in a certain direction and block it until the ongoing transmission completes. These

schemes are based on the Angle of Arrival (AoA) measurements and no extra positioning equipment, such Global Positioning System (GPS), is required.

The deafness problem emerges when directional antennas are applied [94]. Deafness problem does not exist in omni-directional MAC protocols because onmi-directional nodes inherently hear the transmission in its vicinity. In order to alleviate the deafness problem, the majority of directional MAC protocols assume that their antennas can also work in the omni-directional mode, which ensures reception in all directions. Choudhury et al. propose Basic DMAC [89], which is often considered as the benchmark of directional MAC protocols. Basic DMAC assumes that all directional nodes know the profiles of their neighboring nodes, such as their directions, from an upper layer. In this way, no neighbor discovery or localization scheme are needed in Basic DMAC. Idle nodes are supposed to listen to the channel omni-directionally and assumed to be capable of finding the AoA of the received signal. To achieve the communication between Directional-Directional (DD) neighbors, Basic DMAC uses their common neighbors to forward the RTS frame.

Shihab et al. propose DtDMAC [23] that only relies on the directional mode. Furthermore, DtDMAC needs no equipment for synchronization because it works in a complete asynchronous manner. In DtDMAC, communications between DD neighbors is enabled without any intermediate nodes. It continuously rotates its receiver beam sector by sector to achieve omni-directional receiving. The transmitter repeatedly transmits Directional RTS (DRTS) frames according to the number of sectors to ensure the reception of the RTS at the intended receiver. If the location of the intended receiver is known, the transmitter needs to transmit $2M$ DRTS in the direction of its intended receiver, where M denotes the number of sectors. Otherwise, the transmitter needs to transmit up to $2M^2$ DRTS, i.e., $2M$ DRTS in each of M directions, with a random backoff duration in front of each DRTS to guarantee the reception. Obviously, the control overhead severely limits the system performance when M is large.[1] Moreover, neither the transmitter nor the receiver informs its neighbors after the DRTS and Directional CTS (DCTS) exchange which may cause deafness and collisions.

In directional MAC protocols, the directions of neighbors gain in importance because nodes need them to carry out directional communications and avoid collisions. Instead of simply assuming availability, the majority of the directional MAC protocols try to address it. In a number of different proposals, extra devices or mechanisms are used to provide such information, e.g., [114] uses the GPS and [89] relies on an unspecified upper layer mechanism. However, as stated in [68], directions obtained from AoAs ares more reliable than from the extra devices because the AoA represents the real path of the wireless signal.

To the best of our knowledge, most of the state-of-the-arts on directional MAC protocols considers only very limited and simplified network topologies. Most of the protocols are simulated with a fixed and stationary topology. For instance, a

[1] We note that the number of sectors can be large especially in the case of future mm-Wave systems where beam width generated by phased antenna arrays can be very narrow

linear distribution topology is used in [38], and a grid topology is used for simulation in [69].

2.1.2 mm-Wave Communication System

A large amount of currently unused spectrum exists at mm-Wave frequency band. In particular, at least 6 GHz continuous unlicensed band are available around 60 GHz carrier frequency worldwide, which has a great potential for high speed and short range indoor communications [73, 91]. Using mm-Wave technologies proposes new challenges due to the different propagation characteristics compared to the sub 6 GHz wireless networks. In order to exploit the large amount of unoccupied bandwidth, a comprehensive understanding of the 60 GHz propagation is essential [87, 91]. In order to reveal the channel characteristics in different real deployments, several channel sounder based measuring campaigns have been recently reported [46, 51, 58, 106]. General properties of the mm-Wave wireless channel, i.e., propagation time delay spread and pathloss in outdoor environments were measured and analyzed in [92, 106]. The typical short range indoor propagation was characterized in [58]. The reflection, scattering, diffraction, and penetration of 60 GHz transmissions in an office building environment were measured in [51]. Moreover, empirical study of the bit error rate (BER) of realistic modulated 60 GHz links is enabled by means of the new emerging 60 GHz hardware. For instance, Ansari et al. carried out a bit error ratio based measurement [46] by using Commercial-Of-The-Shelf (COTS) such as SiversIMA FC1005V/00 and HXI Gigalink 6651. Compared to the results from sounder based systems, the results of BER based measurements are more practical and straightforward for upper layer applications, e.g., designing and verifying the IEEE 802.11ad rate adaptation scheme and investigating the potential of spatial reuse.

2.1.3 IEEE 802.11ad standard

In order to provide high speed and reliable communications for new emerged wireless applications, the IEEE 802.11ad task group has worked on the IEEE 802.11ad standard [44]. It achieves high data rates up to multi-gigabit level in the 60 GHz mm-Wave band, where a lot of unlicensed spectrums exist worldwide. The directional antenna is used to compensate the pathloss. The traditional 802.11 architecture is no longer suitable for 60 GHz wireless ad hoc networks. A new architecture named Personal Basic Service Set (PBSS) is used in 802.11ad which performs in an ad hoc manner. In a PBSS, one station (STA) is chosen as the PBSS Central Point (PCP) / Access Point (AP) which synchronizes the BPSS and schedules the transmission in Data Transmission Time (DTT). The aforementioned DVCS and DNAV schemes are adopted by 802.11ad standard. However, due to the fact that 60 GHz channel is heavily location depended and the link can be efficiently blocked for example by human bodies, it is risky to use a specific central STA for network synchronization and scheduling in mobile environments.

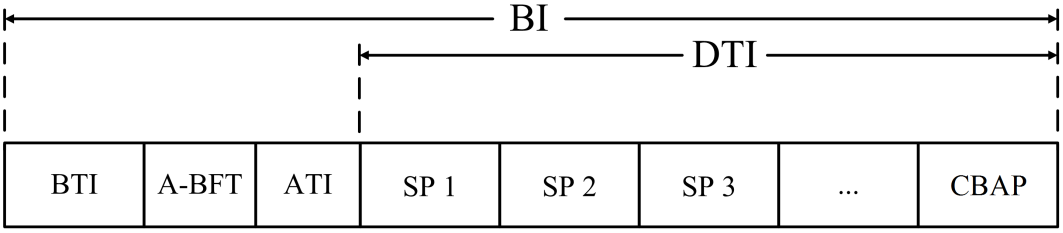


Figure 2.1: A general PBSS of IEEE 802.11ad standard and the Beacon Interval.

In the IEEE 802.11ad standard, the basic time unit of the Directional Multi-Gigabit (DMG) channel access is the Beacon Interval (BI) (Figure 2.1). A typical BI consists of two kinds of access periods, namely the Beacon Header Interval (BHI) and the Date Transfer Interval (DTI). BHI is a contiguous period, which starts as the BI and ends before the starting time of the DTI. BHI can be further subdivided into Beacon Transmission Interval (BTI), Association Beamforming Training (A-BFT) period, and Announcement Transmission Interval (ATI). PCP/AP transmits beacon frames for scheduling transmission in BTI, and the beamforming training is conducted during A-BFT by exchanging beamforming training frames between DMG STAs. The PCP/AP transmits DMG announcement frames during ATI. Data transmissions are carried out during DTI.

A number of works suggest improvements to IEEE 802.11ad by addressing the directional MAC challenges. For example, Park et al. use conventional 2.4 GHz to perform the coordination and neighbor discovery for 60 GHz directional devices [40]. A relay based mechanism for IEEE 802.11ad Contention Based Access Period (CBAP) enhancement, namely Directional Cooperative MAC (D-CoopMAC) is proposed in [88]. D-CoopMAC is based on the typical PBSS structure of IEEE 802.11ad standard. The key idea of this work is to enhance the performance of low data rate STAs by allowing relay STAs to forward their packets. D-CoopMAC uses a cooperative weight vector to determine whether the relay functionality should be carried out. The vector is defined by the ratio of the reciprocal of the data rate of the direct link and the sum of the reciprocals of the relay links:

$$W_{\text{relay}} = \frac{R^{-1}_{\text{source, PCP/AP}}}{R^{-1}_{\text{source, relay}} + R^{-1}_{\text{relay, PCP/AP}}}. \tag{2.1}$$

If the vector in (2.1) is less than 1, then the source STA should communicate with the PCP / AP directly. Otherwise, using relay STA offers better overall throughput. The work in [88] also considers the influence of different Modulation and Coding Scheme (MCSs) of IEEE 802.11ad which is essential for the weight vector definition. However, [88] only considers an ideal directional antenna model which has no sidelobes or backlobe, i.e., the overall interference level is underestimated. Furthermore, its Ready-To-Send (RTS) - DMG Clear-To-Send (CTS) - DMG Help-To-Send (HTS) handshake mechanism cannot alleviate the hidden terminal problem, which may lead to low efficiency.

It is worth noting that the conventional four way handshake or similar mechanisms are no longer efficient for contention-based directional-only networks, e.g., IEEE 802.11ad CBAP, because it cannot alleviate the deafness problem while generating unnecessary Co-Channel Interference (CCI) [39, 77]. By contrast, the scheduling of SPs in IEEE 802.11ad standard that is performed at PCP/AP eliminates these problems due to its centralized nature. To the best of our knowledge, the detailed analysis and improvements of the IEEE 802.11ad MAC, especially its SPs are quite limited. The performances of both CBAP and SP of IEEE 802.11ad are analyzed in [18]. The authors use a Markov chain based model to analyze the CBAP performance, and a queueing system to analyze the performance during SP. However, the directivity of the antenna, which is one of the most significant characteristics of IEEE 802.11ad standard, is not well addressed in the study. Moreover, the influence of the network topology on the SINRs and the corresponding MCS levels is not addressed when analyzing throughput. Lastly, only the delay is considered which provides a limited evaluation of SPs.

In order to carry out concurrent reliable transmission with directional antennas, the CCI should be accurately measured, especially when practical antennas with sidelobes are considered. A number of studies investigate the feasibility of conducting parallel transmissions and its potential benefits [71, 84]. Xu et al. purpose a link scheduling strategy for concurrent transmissions in mm-Wave band [79]. It uses SINR based criterion to judge whether concurrent transmissions are capable of being carried out. The sidelobes and backlobe effects are also addressed in their study, which make it more applicable. Their study illustrates that SINR based scheduling can achieve a considerable improvement. However, the benefit of jointly considering different MCS levels and concurrent transmissions are not presented in this study. Actually, in response to the lowered SINR caused by CCI, parallel transmissions may be still beneficial with lower MCSs which adapt to the new SINRs.

The aforementioned MAC layer enhancements to IEEE 802.11ad standard are only evaluated by simulations, which are less convincing due to their ideal and inaccurate estimation of 60 GHz propagation characteristics. To the best of our knowledge, there is no prior IEEE 802.11ad related study that jointly addresses the Physical (PHY) layer measurement and MAC layer protocol design.

2.1.4 Frame aggregation in IEEE 802.11ad standard

The structure of an IEEE 802.11ad frame which is named as PHY layer Protocol Data Unit (PPDU) in the standard is shown in Figure 2.2(a). It consists of a PHY Layer Convergence Procedure (PLCP) preamble, a PLCP header, a MAC header, a MAC Service Data Unit (MSDU) which is the MAC layer payload, and a checksum sequence. In order to show the transmission efficiency, we use the darker shaded blocks to represent the useful payload, and lighter blocks to mark the control overhead. As shown in the Figure 2.2(a), the overhead of transmitting one MSDU without aggregation is high. Frame aggregation is proposed to reduce the average overhead for each MSDU by grouping them together and then transmitting in one

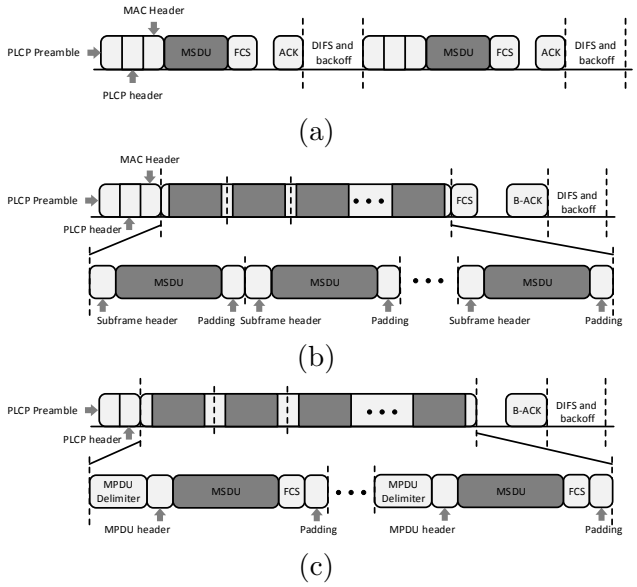


Figure 2.2: Frame format of IEEE 802.11ad standard. (a) frame format of normal MPDU; (b) frame format of AMSDU; (c) frame format of AMPDU.

channel access. There are two types of frame aggregation in IEEE 802.11 standards, namely Aggregated MAC Protocol Data Unit (AMPDU) and Aggregated MSDU (AMSDU). AMPDU is done by first adding a MAC header and generating the checksum sequence to each MSDU to create an MPDU which is the payload of PHY layer, and then transmits multiple MPDUs in a single PPDU. Unlike the AMPDU, AMSDU allows the transmitter to send multiple MSDUs to the same receiver in a single MPDU, i.e., the aggregated MSDUs are transmitted by using only one PHY layer header, one MAC layer header and being proved by one MAC layer checksum sequence.

The most significant difference between AMSDU and AMPDU is the time of performing the MAC layer encapsulation [115]. For AMSDU, the MAC layer encapsulation is done after the aggregation, therefore all encapsulated MSDUs has the same destination address and one checksum sequence. On one hand, its compact structure has a high channel utilization ratio. On the other hand, it has low resistance to the bit error rate because the packet error results in retransmit all MSDUs in the failed AMSDU. In contrast, due to the separately encapsulated MPDU, AMPDU only need to retransmit the collapse MPDU after checking the MPDU individually. Therefore, it is proved in a number of literatures that AMPDU performance better than AMSDU in error-prone wireless channels. [12, 19]. The comparison of AMSDU and AMPDU is out of the scope of this study, so we only analyze the frame aggregation scheme by using AMPDU in the rest of this study.

2.2 Network Topology and Performance

In recent years, stochastic geometry models have been used for investigating the performance of large scale wireless networks [37]. However, the typical scenario of using directional antennas is quite different from the omni-directional ones. First, the interference generated by directional antennas are not identical in all directions. Therefore, compared to the distribution of locations, the specific network topology plays a more important role in network performance. Second, many promising directional antenna techniques, such as 802.11ad standard, are operating in the mm-Wave band whose high pathloss severely limits the network scale. Third, in order to have successful transmission, directional nodes are more likely to know the direction of its neighbors. In this way, directional nodes have the potential to avoid interference by selecting directions intelligently. Therefore, we investigate the relation between network topologies and the performance in this study. To simplify the proof, we assume that the beamwidth of all nodes are identical, and only line of sight links are considered in this study. Therefore, a N node network can own at most $\binom{N}{2}$ undirected links. It is worth noting that the network topology in this study includes both the locations of all networked nodes and their beamwidth.

Besides the destined transmissions, interference is also generated via links. However, the level of interference is not the same for different links. Under a given network topology, some links may be interfered with other transmissions that are carried out in other links. For instance, in Figure 2.5, link DA may be interfered with the transmission via link BA. In order to represent the interference level of links, we have the following definition.

Definition 1 *Non-interferable link is the link which cannot be interfered with any other transmission in the network.*

The number of non-interferable links represents the potential of a network to carry out parallel transmissions. Thus we define the non-interferable factor to represent the potential of the spatial reuse of a network.

Definition 2 *Non-interferable factor of a network is defined as the ratio of the number of non-interferable links to the number of all links. Non-interferable network is the network where all links are non-interferable links. The Non-interferable factor of a non-interferable network is 1.*

The larger non-interferable factor means that more simultaneous transmissions can be carried out. Generally, the highest performance takes place when all links are non-interferable links. In order to find the highest performance with respect to the network topology, we first consider node N_1 whose beamwidth approaches to θ. As shown in Figure 2.3, we define the direction of its first neighboring node N_2 as ray l_2 whose initial point is N_1. Therefore, the transmission from N_1 to N_2 can interfere the area between ray l_1 and ray l_3, where the angle between l_1 to l_2 and the angle

between l_2 to l_3 approach $\frac{\theta}{2}$. To avoid unnecessary interval between two allocations, we allocate N_3 in the ray l_3. If we repeat the allocation for the whole circle, up to $\lfloor \frac{4\pi}{\theta} \rfloor$ neighboring nodes are able to be allocated around N_1. However, considering the area of interference in Figure 2.4, no node is allowed to be allocated in the opposite direction of a pre-allocated node. Therefore, at least half of the nodes should be removed to avoid interference. In order to simplify, we assume that all nodes are allocated one by one in the upper half plane.

We consider node N_n, the $n-1$th neighboring node of node N_1 which is allocated in the corresponding rays l_n as $(n-1)\frac{\theta}{2} < \pi < n\frac{\theta}{2}$. We assume that all links including N_1N_2, N_1N_n and N_2N_n are non-interferable. Obviously, node N_n should be placed above the ray N_2A, which is one side of the beam of node N_2 when it is facing towards node N_1. We extend ray l_1 to an auxiliary point B and draw an auxiliary line N_nC which is parallel to it. Let $\angle BN_1N_n$ be λ, thus $\angle CN_nN_1$ is equal to λ.

As we assumed before, λ is less than $\frac{\theta}{2}$. We add an auxiliary point D which lies on the upper side of node N_n's beam. As shown in the figure, $\angle DN_nC$ is equal to $\frac{\theta}{2} - \lambda$ which is greater than 0. Consider that node N_n is located above line N_2A, so node N_2 must be in the range of node N_n's beam, i.e., link N_2N_n is not interferable, a contradiction. Therefore, λ must be greater than or equal to $\frac{\theta}{2}$. That is, if we want to allocate $n-1$ nodes around node N_1, the beamwidth of all nodes should be less than $\frac{2\pi}{n}$. Therefore we have derived the following theorem.

Theorem 1 *The necessary condition of existing a n nodes non-interferable network where all nodes are capable of hearing each other is that the beamwidth* $\theta \in [0, \frac{2\pi}{n})$.

In general, a larger beamwidth covers more directions, which may reduce the rotating/switching overhead if nodes need to communicate with different nodes. Therefore, for a n nodes network, increasing the beamwidth while keeping the network being non-interferable can optimize the performance. Repeating the analysis of node 1 to all nodes and considering the symmetry, we have,

Corollary 1 *For a wireless network consisting of n directional nodes whose beamwidth approaches to* $\frac{2\pi}{n}$ *and all nodes are capable of hearing each other, the*

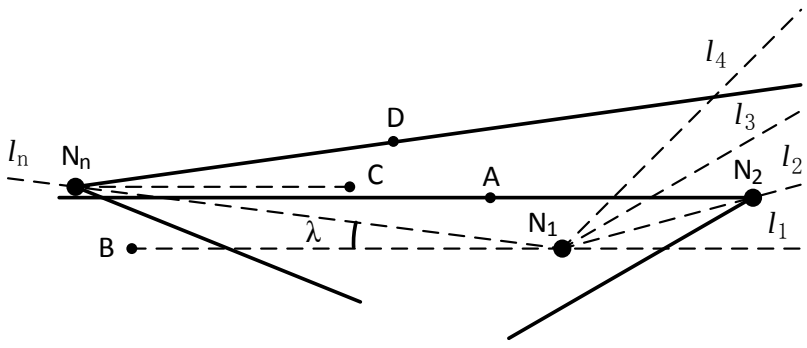


Figure 2.3: Proof of Theorem1.

non-interferable network exists only when all n nodes are allocated at the vertices of a regular n-sided polygon, respectively.

It is worth noting that the non-interferable factor only represents the level of potential spatial reuse of a network topology. The final performance of a directional MAC protocol depends on the protocol design. Therefore, it can be used as a criterion to test the level of the spatial reuse of directional MAC protocols. Moreover, network optimization and planning can also be done based on the network topology.

2.3 Cooperative Design in Uncoordinated Directional Networks

Carrying out seamless transmission in uncoordinated Directional Networks is even more challenging then in coordinated networks [39, 77]. Directional transmission requires the transmitting node and the receiving node to face towards each other before the transmission takes place. However, in uncoordinated networks the deafness problem may happen, which decrease the network performance. Due to the same reason, estimating or tracking nodes' movement becomes a necessity if nodes may move. Moreover, efficiently utilizing the spatial reuse requires nodes to carry out transmissions according to the environments information and behaviors of their neighboring nodes cooperatively, i.e., nodes should be able to perform communications cognitively. Therefore, we pro pose Coop-DMAC to address deafness problem by letting idle nodes switch directions according to the location and the mobility of their neighboring nodes and the network status. Cooperative mechanisms are discussed and evaluated in Coop-DMAC to improve its spatial reuse. Coop-DMAC enables nodes to obtain their local network topology by sharing their neighboring AoA information. Besides, communicating nodes are supposed to announce the duration of their imminent transmission to their neighbors. In this way, nodes can estimate the generated Signal-to-Interference Ratio (SIR). If the caused interference is not harmful to the ongoing transmissions, parallel communications can be carried out and the spatial reuse of the network is improved. In that sense, Coop-DMAC borrows an operational principle from interference limited dynamic spectrum access community. It is worth noting that the caused interference level is not only related to the transmission power and the distance between networked nodes, but also determined by the locations of nodes and the beamwidth of the directional antennas. Hence, the relation between the network topology and the performance of directional MAC protocols is also investigated in Coop-DMAC.

2.3.1 Carrier Sensing

In general, applying carrier sensing with directional antenna is more complicated due to its directionality. Collision avoidance and the deafness problem in omni-directional MAC protocols are intuitively solved by its omni-directional carrier

sensing. However, simply sensing the direction of the destined node can neither solve the deafness problem nor detecting all potential sources of collision in directional networks. The directional antenna can only be interfered via some special directions due to its directionality. Coop-DMAC adopts two types of carrier sensing schemes. The Dual Directional Carrier Sensing (DDCS) scheme is used for detecting potential collisions not only in the direction of the destined node, but also in the opposite direction. The Idle Listening (IL) - Directional Medium Reservation (DMR) scheme is used to address the deafness problem.

2.3.1.1 DDCS Scheme

Collisions may occur at both the transmitter side and the receiver side due to handshake and Acknowledgment (ACK mechanism). Therefore, before transmitting, the transmitter should perform carrier sensing not only towards the direction of the destined node, but also towards the opposite direction to avoid collision. As shown in Figure 2.4 where the dashed circles indicate the transmission ranges of nodes, node A is going to transmit to node B. Obviously, node A has to sense the direction of node B to ensure that no node is transmitting in this direction which may cause a collision at node A when node A is receiving from node B. The area sensed by node A is marked by the sector AGH which angle is equal to the beamwidth of the adopted antenna. Nodes which locate out of AGH cannot generate interference at node A when it is receiving directionally from node B because these nodes are not in the receiving beam of node A. However, simply checking the direction of node B cannot avoid the collision at node B, the destination of this communication. Hence, in order to ensure no node is able to interfere the reception of node B of the imminent communication, node A also needs to sense the opposite direction of node B. As shown in Figure 2.4, node A is supposed to sense the sector ACD that subtends a central angle of twice the beamwidth.

We first show the necessity of performing this opposite carrier sensing. We divide the area BEF where potential interfering nodes for node B may locate into two areas, ACDEF and BCAD. We assume that node X is inside ACEFD and it is transmitting to node B. If only the direction of node B is sensed, node A will transmit to node B without considering the transmission between node X and node B, which will causes a collision at node B. Therefore, node A also needs to sense the direction of the potential interference source of node B before starting transmitting.

After that, we show the sufficiency of this opposite carrier sensing. First, we assume that node Y locates out of sector BEF. Node A does not need to sense the direction of node Y because: (1) node A may not able to detect the transmission from node Y to node B because node A may be not in the range of node Y's transmitting beam; (2) node B does not face towards node A if it is receiving from node Y, i.e., no collision can occur even though node A ignores node Y and starts transmitting to node B while node B is receiving from node Y. In other words, there is no collision to avoid due to the directivity of the antenna. Second, we assume that node Y locates in BCAD. In this case, even through node A is sensing in the direction of node Y, it

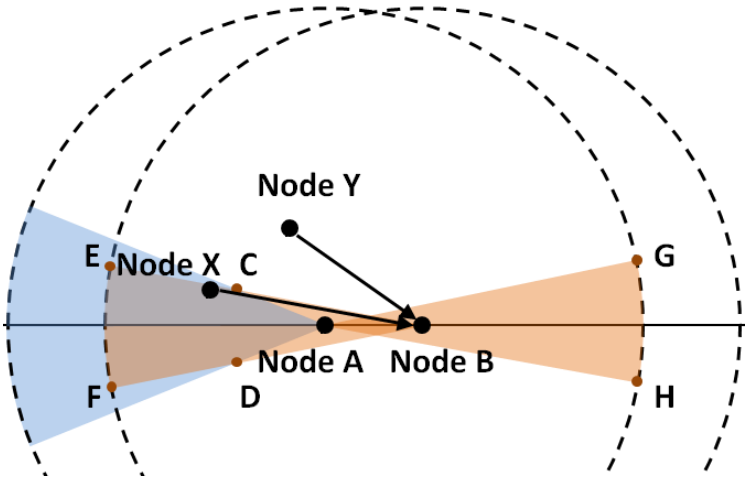


Figure 2.4: Sensed area of DDCS scheme.

is not capable of detecting the transmission from node Y to node B. In other words, the collision cannot be avoided even the directional sensing is performed. Therefore, node A does not need to sense the direction which is out of ACD.

Although DDCS scheme cannot fully observe the medium to perfectly avoid collisions, it significantly improves the efficiency of directional carrier sensing. Compared to the single directional carrier sensing, the detectable area is increased from AGH to AGH+ACEFD. DDCS increases delay by sensing the opposite directions. However, compared to the frame transmission time, the cost is very limited.

2.3.1.2 IL-DMR Scheme

Coop-DMAC uses the IL scheme to eliminate the deafness problem at idle nodes. It requires an idle node to check all directions of all potential transmitting nodes similar to the virtual omni-directional sensing schemes in [23, 44]. The action of the idle node depends on its knowledge of its neighbors. For instance, if the network is fixed, and the directions of all neighbor nodes are known, the idle node only needs to switch its antenna beams towards its neighbors sequentially. On the contrary, if the directions of neighbors are changing very quickly, and new neighbors may emerge randomly, the idle node needs to circularly sense all directions. In both cases, the idle node senses a certain direction periodically. DMR scheme is applied at the transmitter side. It requires the transmitter to send multiple frames in the direction of its intended receiver to ensure the receiver can receive at least one frame. In Coop-DMAC, IL scheme and DMR scheme are always applied together, so in the rest of the chapter we refer it as IL-DMR scheme. In general, idle nodes have two types of actions for changing directions, discontinuously switching between sectors or rotating. Obviously, increasing the switching or rotating speed of idle nodes can reduce the number of repetitions at the transmitter side, which reduces the

control overhead. However, the switching or rotating speed cannot be so fast that no complete frames can be received.

We first consider a simple scenario where nodes are immobile and have the full knowledge of the network topology. If no new node is allowed to emerge, an idle node only needs to sense several fixed directions where its neighboring nodes are located. In order to receive at least one complete frame, the residence time of an idle node in a certain direction t_{rx} should be lower bounded by $t_{\text{fr}} + t_{\text{gap}} + t_{\text{header}}$, where t_{fr}, t_{header}, and $t_{\text{inter_fr}}$ denote the duration of transmitting a frame, the duration of transmitting a frame header, and the inter-frame gap. At the transmitter side, the number of frames n_{tx} need to be sent should be also lower bounded as

$$n_{\text{tx}} \geq \frac{(t_{\text{fr}} + t_{\text{gap}} + t_{\text{header}})(N-1)}{t_{\text{fr}} + t_{\text{gap}}}. \tag{2.2}$$

Most directional MAC protocols [68, 89] assume that idle nodes are receiving in omni-directional mode. In Coop-DMAC, the omni-directional receiving mode is an extreme case where t_{rx} is 0 and n_{tx} is 1.

Idle nodes have to sense in all directions circularly if the network topology changes rapidly. In order to guarantee a successful reception of at least one frame in a direction, the angular velocity of idle node ω_{rx} should be bounded as

$$\omega_{\text{rx}} < \frac{\theta}{t_{\text{fr}} + t_{\text{gap}} + t_{\text{header}}}, \tag{2.3}$$

where θ denotes the beamwidth. The longest transmission time takes place when the rotating idle node just leave the direction of the transmitting node before the transmitting node start sending frames continuously. So we have

$$\begin{aligned} t_{\text{longest}} =& \frac{2\pi - \theta}{\omega_{\text{rx}}} + (t_{\text{fr}} + t_{\text{gap}} + t_{\text{header}}), \\ \geq& \frac{2\pi}{\theta}(t_{\text{fr}} + t_{\text{gap}} + t_{\text{header}}). \end{aligned} \tag{2.4}$$

If the destined node is mobile, the transmitter may need to rotate while it is transmitting to cover all possible directions. The angular velocity of the transmitting node should be upper bounded according to the longest transmission time in a certain duration. Thus we have

$$\omega_{\text{tx}} < \frac{\theta^2}{2\pi(t_{\text{fr}} + t_{\text{gap}} + t_{\text{header}})}. \tag{2.5}$$

In this way, Coop-DMAC eliminates the deafness problem at idle nodes.

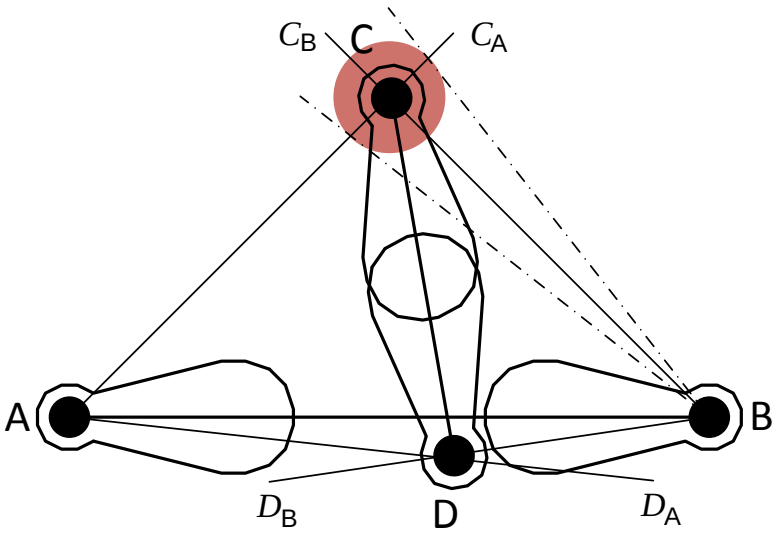


Figure 2.5: An example of two pairs of directional nodes are communicating concurrently.

2.3.2 Topology Analysis

Traditionally, localization is not seen as an essential part of a MAC protocol. However, directional MAC protocols can benefit significantly from localization information, because only knowing directions is not sufficient for supporting efficient spatial reuse.

In order to find the network topology, Coop-DMAC requires nodes to share its cached AoAs of their neighboring nodes by inserting them into frames. As shown in Figure 2.5, if node A receives a frame from node B which contains AoA information of node C, node A is able to localize it with respect to the direction of node B by applying the law of sines as

$$d_{\text{A-C}} = d_{\text{A-B}} \frac{\sin \angle ABC}{\sin(\angle ABC + \angle BAC)} \ , \tag{2.6}$$

Nodes should maintain a table which does not only record the AoA of its neighbors, but also the AoA of the neighbors of its neighbors. The distances to its neighboring nodes are normalized to the distance between the node and its first detected neighboring node. With the help of a rangefinder, the node is able to calculate the precise locations of it neighbors.

By using the network topology, a node is able to estimate the antenna gain of their neighboring nodes towards its direction. Thus it can be used to estimate the caused SIR at a neighboring node if it starts transmitting[2]. As shown in Figure 2.5, node A

[2] It is worth noting that the estimated SINR at the neighborhood is not very accurate due to the uncertainties related to the channel and environment, but the estimate nevertheless useful for enhancing MAC operation.

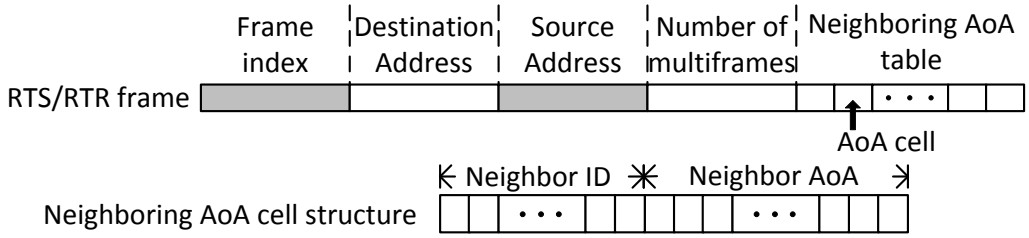


Figure 2.6: Frame structure of RTS frame and RTR frame.

wants to transmit to node B while node C is transmitting to node D. Furthermore, node A knows the transmission between node C and node D. If node A does not know the network topology, it cannot transmit to node B because it may interfere the transmission between node C and node D. However, if node A knows the location of node C and node D, it knows the directions of these two nodes when they are communicating towards each other.

In order to ensure the basic functionalities of directional transmissions and protocol execution, the system is assume to have reasonably accurate capability to estimate the distance between nodes. This could be, of course, provided by some localisation hardware but also radio ranging could be used where one has to make some simplifying assumptions on the radio channel between transceivers. Such ranging techniques have been developed in the context of cellular networks and many indoor localization systems for IEEE 802.11 networks. Although we are agnostic in this chapter for the actual method used for derivation of distance metrics, for the sake of completeness we show the situation under simplified assumption that we have Line of Sight (LOS) links with short distances enough to follow Friis Transmission Equation. This is, of course, a simplification just to illustrate our approach. In this simplified scenario, node A can estimate the SIR $SIR_{\mathrm{D_{AB}}}$ at node D if it is transmitting to node B directly as,

$$SIR_{\mathrm{D_{AB}}} = \frac{G_{\mathrm{CD}} G_{\mathrm{DC}}}{G_{\mathrm{AD}} G_{\mathrm{DA}}} \frac{PL_{\mathrm{CD}}}{PL_{\mathrm{AD}}} = \frac{G_{\mathrm{m}}^2}{G_{\angle \mathrm{CAD}} G_{\angle \mathrm{CDA}}} \frac{PL_{\mathrm{CD}}}{PL_{\mathrm{AD}}}, \tag{2.7}$$

where G_{m} is the gain in the main lobe, and G_θ is the gain in the direction θ. Moreover we have assumed here, without loss of general applicability of the method, that all nodes have same transmission power. It is worth noting that the transmitter also needs to receive controlling frames such as ACK frame. Furthermore, the SIRs at its destined receiver should also be considered. Therefore, node A, as the transmitter, should consider SIRs at all involved nodes, i.e., node A, node B, node C and node D. If all related SIRs, i.e., $SIR_{\mathrm{A_{CD}}}$, $SIR_{\mathrm{A_{DC}}}$, $SIR_{\mathrm{B_{CD}}}$, $SIR_{\mathrm{B_{DC}}}$, $SIR_{\mathrm{C_{AB}}}$, $SIR_{\mathrm{C_{BA}}}$, $SIR_{\mathrm{D_{AB}}}$, and $SIR_{\mathrm{D_{BA}}}$ are lower than the threshold, node A is allowed to carry out a simultaneous transmission towards node B.

2.3.3 Protocol Details

In this subsection, we present the details of Coop-DMAC design which combines all aforementioned techniques. A complete Coop-DMAC procedure includes two 2-way handshakes, an RTS/RTR broadcast mechanism, and an aggregated data frame transmission scheme.

2.3.3.1 Multiframe transmission

Due to the large overhead in directional MAC protocols, Coop-DMAC uses an aggregated scheme, namely multiframe transmission. It is similar to the MAC Protocol Data Unit (MPDU) aggregation in IEEE 802.11ad standard [44]. The number of data frames in the multiframe transmission is inserted in control frames, so the receiver and its neighboring nodes know how long the ongoing transmission goes on. Only a single ACK frame is required to feedback the index of the successfully received data frame after the multiframe transmission. Before the multiframe transmission, the directional connection between the transmitter and its intended receiver is built by the Node Location Request (NLR) - Location Report (LR) handshake.

2.3.3.2 NLR-LR handshake

NLR and LR frames are the shortest frames defined in Coop-DMAC which contains three main fields, i.e., frame type, destination address, and source address. NLR frame is sent by a transmitter to lock its intended receiver after carrier sensing. Before sending NLR frames, the transmitter checks its AoA table. If the direction of its intended receiver has been recorded, the transmitter starts transmitting NLR frames towards the recorded direction. Otherwise, it randomly selects a direction and sends NLR frames circularly for one full turn or until getting the reply from its intended receiver. All idle nodes, including non-destined nodes, are supposed to reply a LR frame immediately after receiving an NLR frame. The destined receiver stays in the direction of the transmitter and waits for the upcoming frame. Otherwise, it turns its beam out of the direction of the transmitter and carries on sensing with IL scheme. After receiving NLR frame and LR frame,

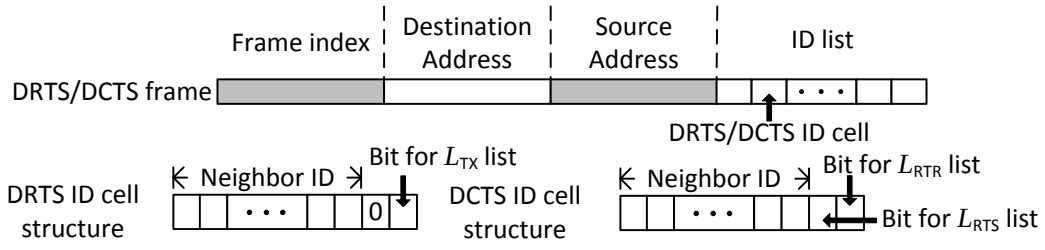


Figure 2.7: Frame structure of DRTS frame and DCTS frame.

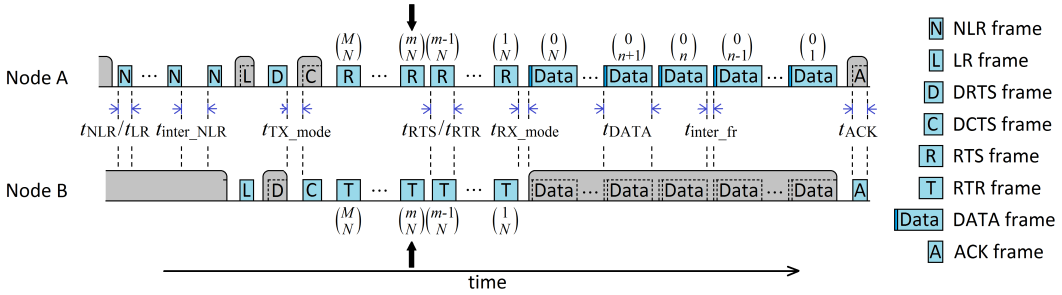


Figure 2.8: A complete transmission cycle of Coop-DMAC in time domain.

nodes update the AoA of the source node. Collisions may occur if two nodes are transmitting NLR frames to each other at the same time with the same inter-NLR interval. Coop-DMAC randomly selects the interval between two NLR frames, which asynchronizes the transmission to increase the probability of building successful connection.

2.3.3.3 RTS-RTR broadcast mechanism and DRTS-DCTS handshake

The multiframe transmission prolongs the transmission duration which aggravates the deafness problem. Informing neighboring nodes about the imminent communications can alleviate the corresponding performance degradation by allowing their neighboring nodes to avoid transmitting to them. Therefore, RTS frame is transmitted by the transmitter, and the RTR frame is transmitted by the receiver after NLR and LR handshake and before multiframe transmission. RTS and RTR frames have the same frame structure which contains the AoA of the neighboring nodes as shown in Figure 2.6. The RTS and RTR transmission also obey the DMR scheme to ensure the reception at idle nodes. In order to avoid unnecessary waiting time caused by the deafness problem, acknowledgement of RTS and RTR frames is not expected. That is, RTS and RTR frames are supposed to send back-to-back with only the minimal inter-frame interval. To efficiently transmit RTS frames and RTR frames, three rules are drawn up. First, do not transmit both RTS frame and RTR frame to a same node. Second, do not transmit to a node which is involved in an ongoing transmission. Third, do not transmit RTS frame and RTR frame so that they collide with each other. Therefore, Coop-DMAC uses the DRTS/DCTS handshake before RTS/RTR broadcast. After receiving the LR frame from its intended receiver, the transmitter inserts the IDs of its available neighboring nodes in a DRTS frame and sends it to its intended receiver. In other words, the list includes all available neighboring nodes, which the transmitter can send RTS frames. After receiving the DRTS frame, the receiver replies with a DCTS frame with the decision of RTS and RTR list according to the three rules. The frame structure of DRTS and DCTS frames are shown in Figure 2.7. After the DRTS/DCTS handshake, the transmitter and the receiver send their RTS and RTR frames to the directions of their neighboring nodes, respectively. The one

who finishes transmitting first turns its beam towards its partner and starts a timer to wait for its partner to finish transmission. After receiving an RTS or RTR frame, a node is able to update the AoA of the source node, estimate the beam direction of the communicating pair, and also estimate the duration of the imminent communication. Moreover, Coop-DMAC allows nodes to adjust the transmission order according to received RTS/RTR frames to alleviate the Head of Line (HoL) blocking problem [77], which further improves its spatial reuse.

2.3.3.4 Channel occupation estimation

A complete transmission procedure is shown in Figure 2.8. Similar to the channel occupation estimation scheme in cognitive MAC protocols [48, 82], the number of DATA frames in the multiframe transmission is inserted in RTS frames and RTR frames. Moreover, the index of RTS and RTR frames are assigned in a reverse order. Therefore, all informed neighboring nodes are able to precisely estimate the duration of the ongoing transmission. For instance, an idle node receives an RTS frame or a RTR frame as marked by the black arrow in Figure 2.8. By knowing the index of the received RTR/RTS frame, and the number of the data frames in multiframe transmission, the remaining time for the ongoing transmission can be calculated.

2.3.4 Mobility

As mentioned in [104], if the speed of a neighboring node is constant, the distance determines the stability of its direction. Coop-DMAC assesses the reliability of the direction of a node by estimating its relative speed and its relative location. For example, node B wants to transmit to node C in Figure 2.5. We assume that node B knows the maximal relative speed V_{BC} of node C by observing its movement for a long enough time. We further assume that the relative location of node C is available at B by using the shared AoA information. After the last communication which determined the last precise location C_0, node C may move towards an arbitrary direction. We use the red circle with radius $r_{\mathrm{BC}} = V_{\mathrm{BC}}t$ to represent all possible locations of node C after time t.

As shown in Figure 2.5, the possible direction of node C after time t is within the two dashed lines. The uncertainty of the direction is defined as the maximal possible change of direction C_{B} which is equal to $\arcsin \frac{V_{\mathrm{BC}}t}{d_{\mathrm{B-C}}}$ in this instance. If the uncertainty of direction is larger than the beamwidth, the transmitter needs to sweep all possible directions according to (2.5) to ensure the reception.

2.3.5 Simulation Results

In this section, we present the MATLAB simulation results to investigate the performance of Coop-DMAC. Antennas are modelled as shown in Figure 2.5 with 25 dBi gain. We assume that the propagation delay is zero. The cumulative goodput

Table 2.1: Simulation Parameters

MAC header	84 bits
RTS/RTR payload	$\lfloor (N^* - 2)/2 \rfloor \times 14$ bits
DRTS/DCTS payload	$\lfloor (N^* - 2)/2 \rfloor \times 6$ bits
ACK payload	$M^{**} \times 6$ bits
DATA payload	2000 Bytes
t_{SIFS}	2 μs
t_{DIFS}	4 μs
- Initial CW size	10 μs
maximal retry number	6

* Number of neighboring nodes
** Data frames in multiframe transmission.

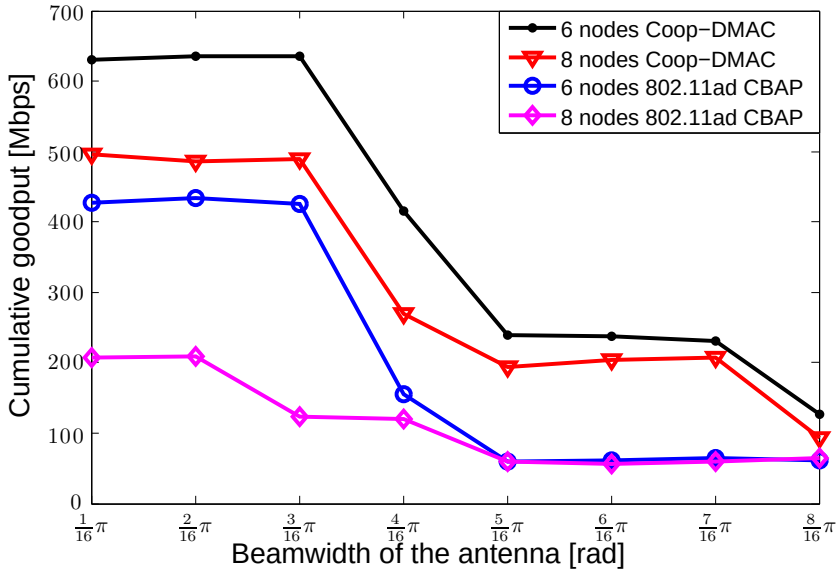


Figure 2.9: Cumulative goodput with respect to the beamwidth of Coop-DMAC and 802.11ad CBAP. The number of multiframes of Coop-DMAC and the length of A-MPDU of 802.11ad are all fixed to 5.

is chosen as the performance metric. Only the saturated network condition is considered in our simulation and each node is supposed to maintain a transmission queue which contains all neighboring nodes. Nodes always try to transmit to the first destination node in their transmission queue. The transmitting node moves the current receiving node to the tail of its transmission queue after a successful transmission. Some key configurations of the simulation are listed in Table 2.1. In order to compare Coop-DMAC with 802.11ad standard, most of the PHY parameters, e.g., the MCS of 802.11ad standard are adopted. The data rate of control PHY is set to 27.5 Mbps, and all data frames are transmitted with MCS4 whose data rate is 1155 Mbps. Omnidirectional mode is used by idle nodes in the first simulation. In other simulations, only directional-only antennas are used so that the

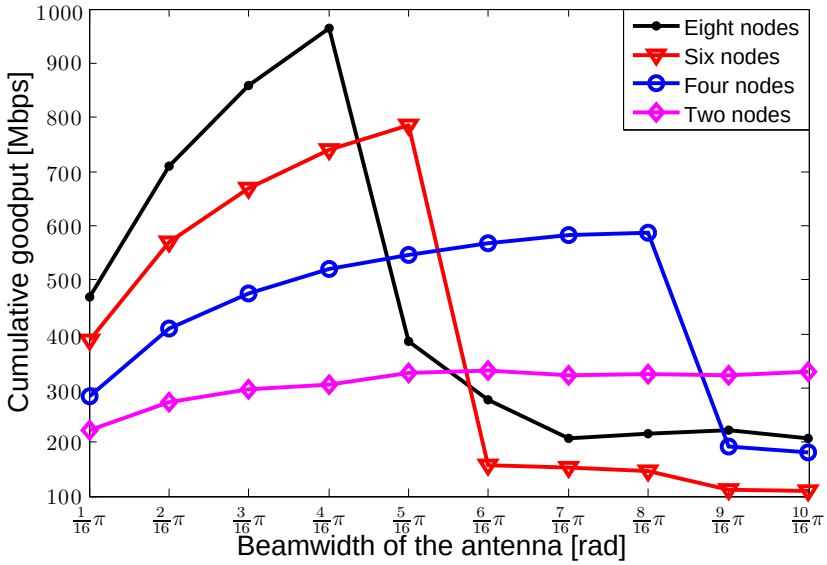


Figure 2.10: Cumulative goodput with respect to the beamwidth of the directional antennas. Nodes are allocated at the vertices of the corresponding n-sides regular polygons whose circumscribed circles have the same radius of 4 m. Nodes are transmitting in pairs. The number of multiframes is fixed to 20.

idle nodes are supposed to continuously rotate according to (2.3). All simulations are repeated enough times to ensure that the results are statistically reliable.

2.3.5.1 Baseline Comparison

In this simulation, we compare the cumulative goodput of Coop-DMAC and the CBAP of 802.11ad. Omni-directional mode is used when a node is idle for both protocols. We assume that all nodes are maintaining a transmitting queue, including all neighboring nodes, i.e., a node is able to transmit to any of its neighbors. In order to compare the two protocols fairly, we use the same configurations such as the DIFS duration, length of addresses in the frame header, and the backoff parameters in both protocols. Nodes are allocated in a grid of two rows with edge length 4 m. It is shown in Figure 2.9 that Coop-DMAC outperforms 802.11ad CBAP in terms of goodput. This enhancement owes to the fact that Coop-DMAC allows more simultaneous transmissions than 802.11ad by analyzing the network topology and estimating the channel busy duration. It is also shown that the cumulative goodput of both protocols decrease as the beamwidth increases. It is due to the fact that large beamwidth causes more interference which reduces the non-interferable factor of the network.

2.3.5.2 Beamwidth and number of nodes

In this simulation, n immobile nodes are allocated at the vertices of the corresponding n-sides regular polygons whose circumscribed circle has radius 4 m.

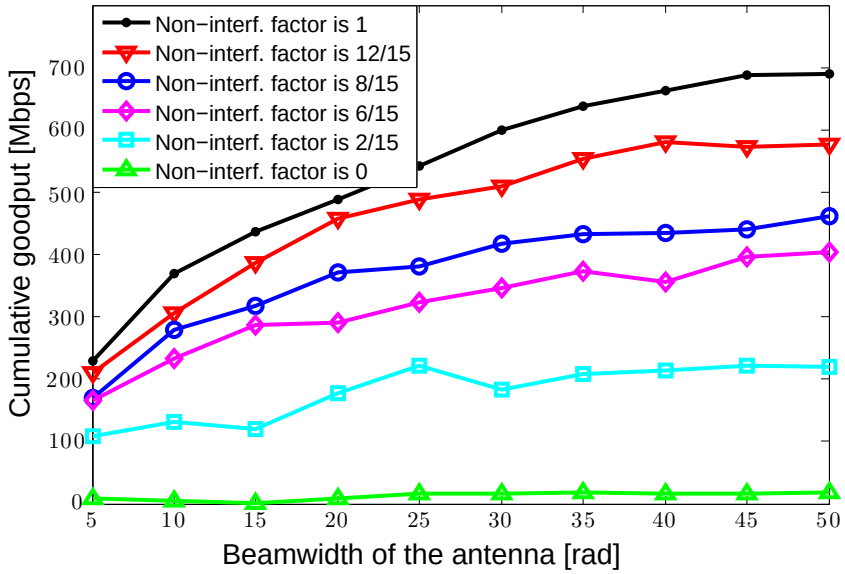


Figure 2.11: Cumulative goodput with respect to the non-interferable factors. Six transceivers are allocated according to the non-interferable factors. The number of multiframes is fixed to 5.

According to Theorem 1, the network switches from non-interferable network to interferable network at some key points according to the number of nodes, e.g., the fall between the fourth point and fifth point of the eight nodes case. Two major conclusions can be drawn from this simulation. First, the goodput of Coop-DMAC in a non-interferable network increases as the beamwidth increases. It is due to the fact that the large beamwidth reduces the time for sensing the medium in IL-DMR scheme, thus further lowers down the number of RTS and RTR frames sent per neighbor. However, the network turns into interferable when the beamwidth is greater than or equal to $\frac{2\pi}{n}$. It occurs when beamwidth is greater than $\frac{1}{4}\pi$, $\frac{5}{16}\pi$, and $\frac{1}{2}\pi$ in eight nodes case, six nodes case, and four nodes case, respectively.

2.3.5.3 Topologies

In this simulation, we evaluate the performance of Coop-DMAC with different topologies. We first allocate six immobile nodes at the vertices of a hexagon which has side 4 m. We adjust the locations of nodes according to the desired non-interferable factor. For instance, the scenario in which the non-interferable factor is 0 takes place when all nodes are allocated in a common line. Figure 2.11 shows that the cumulative goodput of the network decreases as the non-interferable factor decreases. It is worth noting that the performance is not perfectly linear to the non-interferable factor. It also relies on the detailed design of the evaluated directional MAC protocol. The relation of network topology and performance is also investigated in the following simulations while the mobility of nodes is jointly considered.

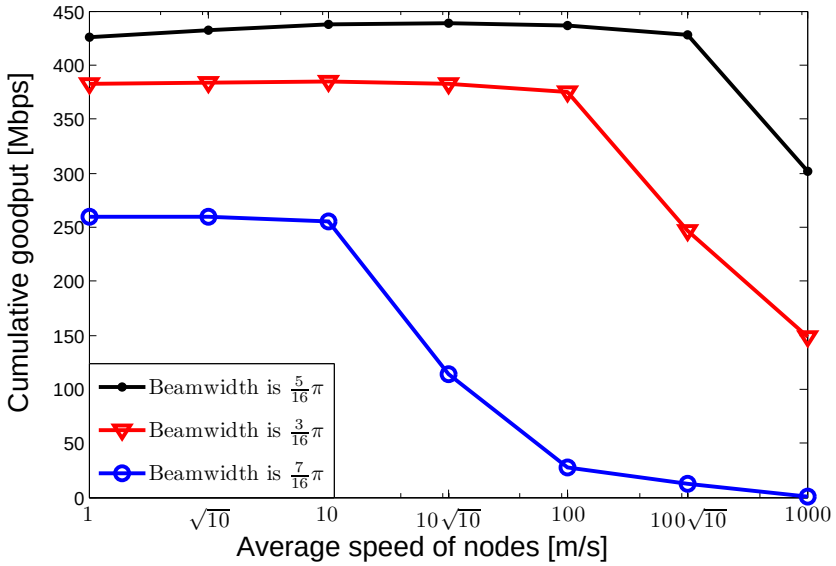


Figure 2.12: Cumulative goodput with respect to the speeds of nodes. Four transceivers are moving in uniform circular motion of radius 1 m. The center of their orbits are at the vertices of a square with side 8m. The number of multiframes is fixed to 20.

2.3.5.4 Mobility

Coop-DMAC is also designed to provide reliable data communication among mobile nodes. In the following two simulations, we shortly evaluate the performance of Coop-DMAC with different node speeds and network topologies which are caused by node movement.

In this simulation, the performance of Coop-DMAC with different nodes speeds are investigated. As shown in Figure 2.12, nodes speeds are varied from 1 m/s, which is less than the average pedestrian walking speed, to 1000 m/s, which is around triple the sound speed in air. In order to keep the network topology constant, four mobile nodes are assumed to move circularly around the four vertices of a square with side 8 m, respectively. The radius of their orbit is 1 m. As expected, the larger beamwidth has better resistance against the node's movement. It is due to the factor that larger beamwidth covers more directions which allows larger uncertain directions. Besides, the larger beamwidth can effectively shorten the disconnected time.

In order to jointly evaluate the impacts of the node mobility and the network topology, we fixed three nodes and let the fourth node move at a constant velocity. All nodes are assumed to have the full neighboring information at the beginning. The instant goodput from 0 ms to 3000 ms is shown in Figure 2.13, where node D moves from D_0 to D_5. As shown in the mini-figure, the non-interferable factor is 1, $\frac{1}{2}$, $\frac{1}{6}$, $\frac{1}{2}$, and 1 in the intervals [0 ms 500 ms), [500 ms 1166 ms), [1166 ms 1834 ms), [1834 ms 2500 ms), and [2500 ms 3000 ms], respectively. Considering the delay which arises from the backoff operation, the instant goodput reflects the node's movement.

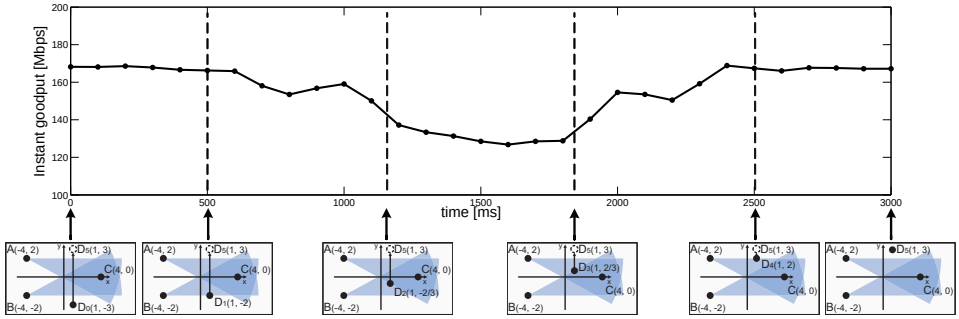


Figure 2.13: Instant goodput with respect to the executed time. Node A, node B, and node C are fixed at points $(-4\,\mathrm{m}, 2\,\mathrm{m})$, $(-4\,\mathrm{m}, -2\,\mathrm{m})$, and $(4\,\mathrm{m}, 0\,\mathrm{m})$, respectively. Node D is initially located at point D_0, $(1\,\mathrm{m}, -3\,\mathrm{m})$. The beamwidths of all nodes are set to $\arctan\frac{8}{15}$. Node D is supposed to move along the x-axis at speed $2\,\mathrm{m/s}$. The number of multiframes is fixed to 5.

2.3.6 Summary

In this section, we propose Coop-DMAC, a completely distributed asynchronous directional MAC protocol and analyze the relation between the network topology and the performance of a directional-only MAC protocol. We propose the directional carrier sensing scheme to let directional nodes avoid potential interference before transmission. We also propose a set of cooperative mechanisms to share the AoA information and the transmitting duration, which can significantly improve spatial reuse and address mobility problem. We also define the interference level of a network based on its topology, and the beamwidth of the wireless nodes. The simulation results validate our theoretical analysis.

2.4 Spatial Reuse in Centralized Directional Networks

Spatial reuse is significantly enhanced in directional network in comparison to omnidirectional networks [23, 68, 84]. IEEE 802.11ad standard introduces the PBSS to organize a group of STAs to communicate at 60 GHz bands. PBSS performs in a centralized manner that only the PCP/AP is able to send control frames and schedule the transmissions, and the contention based transmissions are allowed in CBAP for all STAs. However, as mentioned in [77, 84, 94], carrier sensing is not efficient in directional-only networks, and the deafness problem may lead to a severe performance degradation. The aforementioned weaknesses can be alleviated in SPs due to its centralized mechanism and Time Division Multiple Access (TDMA) based scheduling method. During SPs, all STAs are only allowed to communicate in their own time slots which are preassigned by PCP/AP. Considering the uplink communications, all STAs are supposed to communicate with PCP/AP with different data rate, which are also scheduled by PCP/AP according to their channel conditions. One drawback is that for the STAs which are far away from PCP/AP, their SINR are poor so they can only transmit with low MCSs which

lead to low data rates. In the IEEE 802.11ad standard, the disrupted links can be relinked by inserting a relay STA, which is so-called link switching mechanism. However, how to improve the STAs which has low data rate is not addressed in the standard.

2.4.1 Spatial Buffering

In order to address the aforementioned challenges and problems, we design and verify SBMAC, an efficient MAC protocol which is fully compatible with the IEEE 802.11ad standard. SBMAC expends the link switching mechanism of IEEE 802.11ad standard that the relay STA can also be used for STAs with low data rate. In SBMAC, these relay transmissions are carried out between poor STAs and STAs with better uplink connections in parallel with uplink transmissions between PCP/AP and other STAs. In this manner, the data of low data rate STAs is actually buffered in the STAs which have higher uplink data rate. It evident that the average uplink date rate is improved from the average data rate of all STAs to the average data rate of the selected STAs. In order to demonstrate its feasibility, we build a 60 GHz wireless communication system including the original IEEE 802.11ad MCSs by using COTS devices. MCSs based link measurements are carried out to provide a comprehensive understanding of IEEE 802.11ad standard and the 60 GHz communications. By implementing the key feature of SBMAC on that paltform, we thereby demonstrate the practical feasibility of the proposed SBMAC protocol. To the best of our knowledge, SBMAC is the first directional MAC protocol which is verified on a real-hardware communication system over 60 GHz bands. At last, we compare SBMAC with IEEE 802.11ad standard and a relay based protocol. The numeric results show that SBMAC can achieve much higher uplink data rate than other works.

We consider a general PBSS which is defined in IEEE 802.11ad standard as shown in Figure 2.14. The PBSS contains one PCP/AP and N non-PCP/non-AP DMG STAs. To simplify our analysis, we assume that the PCP/AP and all DMG STAs are equipped with the same half-duplex antenna which can transmit and receive directionally. We assume that the whole PBSS is distributed in a plane so we also use the flat-top antenna model [64]. In this model, the directivity of an antenna is defined by its beamwidth θ with uniform directional gain $G_{\mathrm{M}} = \eta\frac{2\pi}{\theta}$ in its beam. Thus the gain out of the main lobe is $G_{\mathrm{S}} = (1-\eta)\frac{2\pi}{2\pi-\theta}$, where $\eta \in [0,1]$ denotes the efficiency of the radiation in the mainlobe. The assumed flat-top antenna model is used in both the following analysis and also the simulations in Section 2.4.6. It is evident that higher directional gain and narrower beamwidth leads to less CCI, which provides a higher potential of spatial reuse.

In our performance analysis, we only focus on the uplink performance during the SPs. We assume that all non-PCP/non-AP STAs are able to receive the beacon frame during the BTI correctly, and all beamforming operations, such as the Sector Level Sweep (SLS) phase and the operations in A-BFT are completed without error. Therefore, all STAs know the direction of their neighbors in the remaining periods of

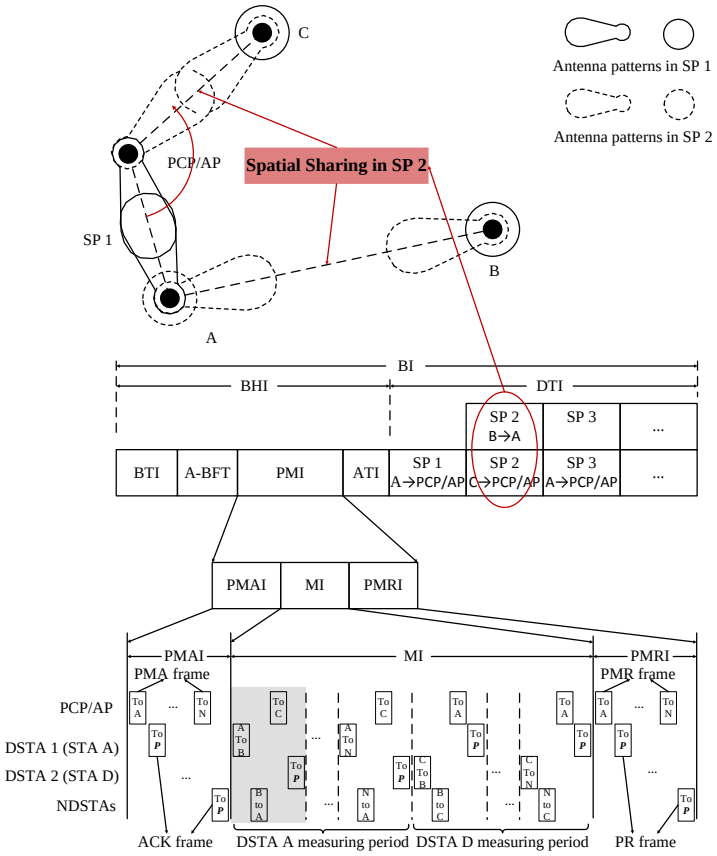


Figure 2.14: A example of a modified IEEE 802.11ad PBSS when SBMAC is applied. Parallel transmissions and the new intervals are also shown.

the BI. Furthermore, we assume that the PCP/AP knows the channel status between it and all non-PCP/non-AP STAs from the link measurement report mechanism. Therefore the PCP/AP is able to schedule a transmission with the maximum MCS scheme which the link can support. An example is shown in Figure 2.14, where the IEEE 802.11ad OFDM PHY is operated by all STAs. The PCP/AP assigns SPs to STAs with the highest MCS level they can achieve. Parallel transmissions may cause CCI, which may decrease the MCS level that a single transmission can achieve. In order to achieve reliable transmissions, the PCP/AP should specify appropriate MCS schemes for parallel transmissions during DTI while considering the existence of CCI.

We assume all STAs have the same source data rate c. Non-PCP/non-AP DMG STAs are numbered from 1 to N in the descending order of their uplink data rates. An STA is able to upload with its maximum data rate, if it is allowed to transmit and its transmission buffer is not empty. Obviously, the uplink performance is also dependent on the scheduling method at PCP/AP. If a Round Robin (RR) Scheduler is adopted, which equally assigns time durations to all non-PCP/non-AP DMG STAs, the overall uplink data rate R is

$$R = \begin{cases} Nc, & r_N \geq Nc \quad (2.8) \\ Lc + \frac{1}{N} \sum_{n=L+1}^{N} r_n, & r_L \geq Nc \geq r_{L+1} \quad (2.9) \\ \frac{1}{N} \sum_{n=1}^{N} r_n, & Nc \geq r_1 \quad (2.10) \end{cases}$$

As shown in (2.8-2.10), the overall uplink throughput is dependent on the average link status between PCP/AP and all STAs. Obviously, STAs with low MCS are the bottleneck of the network. That is, assigning time slots to these 'bad' STAs is inefficient. Nevertheless, in order to guarantee fairness among STAs, transmission opportunities should be assigned fairly. In contrast, we find that the links between 'good' STAs and PCP/AP are idle in most of the time, which is obviously an underutilization of the spectrum resources. Inspired by these drawbacks, we design SBMAC, which uses parallel transmissions to improve the overall efficiency of a 802.11ad network.

The key idea of SBMAC is to divide all non-PCP/non-AP STAs into two categories, namely Direct STAs (DSTA) and non-direct STAs (NDSTA) according to their uplink data rates. In SBMAC, the PCP/AP only communicates to DSTAs directly. If the PCP/AP selects DSTAs appropriately, the uplink data rate can be significantly enhanced from the average uplink data rate of all non-PCP/non-AP STAs to the average uplink data rate of the selected DSTAs. On the other hand, NDSTAs can only transmit their data to a DSTA when the chosen DSTA is not communicating with other STAs. An example of SBMAC with two DSTAs is shown in in Figure 2.14. In this PBSS, STA A and STA C are the DSTAs, STA B is a NDSTA. In the first SP, only STA A is allowed to transmit to the PCP/AP. The directional antenna orientations of the PCP/AP and STA A in the first SP are marked by the solid line. In the second SP, STA C is allowed to transmit to the PCP/AP, while STA B, a NDSTA, transmits to STA A which has a higher data rate to the PCP/AP. Therefore, STA A keeps its own frames and the frame from STA B at the end of the second SP. These frames are supposed to be transmitted to the PCP/AP in the third SP. In other words, by exploiting the spatial reuse of IEEE 802.11ad DMG transmissions, STA A is able to buffer the data of STA B while PCP/AP is still receiving.

Two major issues exist when the above spatial buffering principle is applied. First, DSTA selection determines the system performance. However, finding the optimized DSTA selection is nontrivial, especially in a dynamic wireless environment. Second, after selecting DSTAs, the influence of CCI should be measured for PCP/AP to determine the new MCS of parallel transmissions, which is not specified in the existed IEEE 802.11ad standard. We explain how SBMAC addresses that two issues in the reminder of this section.

Algorithm 1 DSTA Selection Algorithm

```
𝔻 ← {PCP/AP}
for n < N do
    distance check between STA n and 𝔻
    if pass then
        𝔻 ← 𝔻 ∩ STA n
        if |𝔻| = M then break
        end if
    end if
end for
```

2.4.2 DSTA Selection

Selecting the STAs which have higher uplink PCP/AP as DSTAs can improve the overall performance. However, DSTA selection is nontrivial in a PBSS because the CCI level, DSTA selection, and NDSTA grouping are interdependent. Furthermore, measuring CCI for all possible DSTA combinations is costly, especially in a large network. Therefore, the performance of a PBSS can only be improved in multiple BIs by testing different combination of DSTAs and NDSTAs. However, a good starting point may shorten the optimization procedure. To this end, we present a DSTA selection initialization method in Algorithm 1, where we select M DSTAs out of N STAs only based on the channel conditions between PCP/AP and STAs which can be measured in the conventional A-BFT interval. We design the algorithm following two key principles. First, we always priorize the STAs with good connectivity. Second, if a DSTA candidate appears to be too close to any STAs in the group of selected DSTAs and the PCP/AP, namely $\mathbb{D}$, we do not select it because it may severely interfere or be interfered by the transmissions between other DSTAs and PCP/AP. In general, PCP/AP can use the direction of the candidate DSTA and its measured signal power to estimate its location. We assume it is a feasible approach because DSTA candidates are normally the STAs which have the LOS links with short distance from PCP/AP, where more simple and straightforward propagation models can be used for location estimation. For these types of STAs, PCP/AP can estimate the distance by using the received power or triangulation information. On the other hand, if a DSTA candidate is far away from the PCP/AP or it has no LOS, nominating it as a DSTA or not is less relevant to the overall performance.

2.4.3 Co-Channel Interference Measurements

After DSTA selection, CCI measurements should be carried out to determine the reasonable MCS for reliable parallel transmissions. In order to measure the CCI caused by concurrent transmissions, SBMAC inserts one interval in the conventional IEEE 802.11ad BHI structure before ATI, namely Power Measurement Interval (PMI). PMI is further divided into three subintervals, Measurement Announcement Interval (MAI), Measurement Interval (MI), and Measurement Report Interval (MRI). In PMI, CCI is obtained by comparing the power of signals and interference

measured in MI, while the scheduling of MI is announced during MAI and the results are reported from all non-PCP/non-AP STAs to PCP/AP during MRI. An example of detailed PMI scheduling including two DSTAs is shown in Figure 2.14.

MI is the key interval of PMI because it allows STAs to measure the potential CCI level when concurrent transmissions are carried out. The MI is further divided into DSTA measuring periods according to the number of DSTAs. In a DSTA measuing period, the selected DSTA is always supposed to tranmsit with PCP/AP, while other DSTAs are transmitting with different NDSTAs. In other word, the selected DSTA is always facing towards PCP/AP and other DSTAs are changing orientations to NDSTAs in the measuring period of the selected DSTA. As shown in Figure 2.14, in DSTA A measuring period, the SINR at STA A, STA B, STA C and PCP/AP are obtained by comparing the reveiced power of the Power Measurement (PM) frames. For all STAs including PCP/AP, if the received frame is from its own transmitter, the measured power is the power of the useful signal. Otherwise, it is the power of interference. All estimated SINRs are sent back to PCP/AP during MRI.

2.4.4 NDSTA Grouping

Algorithm 2 NDSTA Grouping Algorithm

$\mathbb{ND} \leftarrow$ set of all NDSTAs
$\{\theta_n, n = 1, ..., N\} \leftarrow$ AoA of all STAs at PCP/AP
$\mathbb{D}_m \leftarrow \emptyset, m \in \mathbb{D}$, NDSTA groups of DSTAs
for $n_{\mathrm{ND}} = 1 : |\mathbb{ND}|$ **do**
  for $n_{\mathrm{D}} = 1 : |\mathbb{D}|$ **do**
    $\delta_{n_{\mathrm{ND}}, n_{\mathrm{D}}} \leftarrow |\theta_{n_{\mathrm{ND}}} - \theta_{n_{\mathrm{D}}}|$
  end for
end for
while $\mathbb{ND} \neq \emptyset$ **do**
  for $n_{\mathrm{D}} = 1 : |\mathbb{D}|$ **do**
    $j = \mathrm{argmax}(\min \delta_{k,l}, k \in \mathbb{ND}, l \in \mathbb{D}, l \neq n_{\mathrm{D}})$
    $\mathbb{D}_m = \mathbb{D}_m \cup j$
    $\mathbb{ND} = \mathbb{ND} \setminus j$
    if $\mathbb{ND} = \emptyset$ **then break**
    end if
  end for
end while

After selecting DSTAs, the remaining STAs are NDSTAs, i.e. they are only allowed to transmit their data to a DSTA. In general, a NDSTA should be assigned to a DSTA with the best connectivity. Nevertheless, the NDSTA grouping is nontrivial due to the same reason as for DSTA selection. Moreover, as illustrated in [84], the performance of a directional network is highly dependent on the network topology. Therefore, only partial CCI information cannot support the optimization of NDSTA grouping. Machine learning based algorithm with positioning technologies could

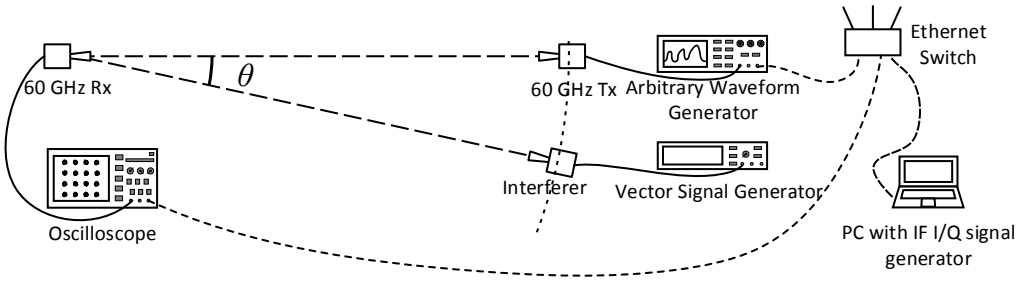


Figure 2.15: Experimental setup including one transmitter, one receiver and one interferer.

be the future work of NDSTA optimization. However, the directional nature of the interference caused by parallel transmissions provides us an opportunity to massively reduce the possible interference in the initialization phase. Considering a variety of existing angle of arrival (AoA) detection technologies [68], we let the PCP/AP perform NDSTA grouping by only awaring of the directions of all STAs, i.e., including both DSTAs and NDSTAs. As shown in Algorithm 2, the PCP/AP assign a NDSTA to a DSTA set $\mathbb{D}_n, n \in \mathbb{D}$ which maximize the minimum distance between it and other non-assigned DSTAs because it is the most significant potential interfered link. Moreover, balancing the grouping size can also improve the performance. Therefore, SBMAC assigns NDSTAs to DSTAs in a round robin manner.

We would like to note the a special case where some NDSTAs may not be able to communicate with any selected DSTA due to the limitation of communication range and the unbalanced DSTA allocation. In this case, a simple solution is for the PCP/AP to assign an individual time slot to each of these unreachable NDSTAs, and allow them to communicate directly to PCP/AP. The existence of unreachable NDSTAs lowers the average uplink data rate because some time slots are assigned to links with very low data rate. This problem cannot be avoided in the early stage of the network, i.e., the scheduling is made based on partial knowledge of CCI and network topology, but could be continually improved. For DSTA selection improvement, it is worth noting that alternate STAs which have the same MCS as DSTAs are safe because STAs with same MCS has the same data rate as regulated in IEEE 802.11ad DMG PHY specification. After NDSTA grouping, the scheduling of buffering transmission can be scheduled according to the NDSTA assignment and measured CCI.

2.4.5 Hardware Verification

In order to practically verify the feasibility of using directional antenna to carry out parallel transmissions, we carry out several hardware verifications using 60 GHz transceivers. We measure the relation between Bit Error Rate (BER) and Signal to Interference and Noise Ratio (SINR) by varying the signal strength of interference signals. We also measure the relation between the network topology and the date

rate by changing the Angle of Arrival (AoA) of the interference. For fair comparison between the IEEE 802.11ad standard and our protocol, we implemented a custom OFDM PHY layer, which follows the IEEE 802.11ad Orthogonal Frequency Division Multiplexing (OFDM) PHY specification principle including 32 subcarriers in 50 MHz bandwidth at 61.5 GHz frequency band. The original IEEE 802.11ad OFDM PHY are considered in all measurements so that SBMAC is tested upto MCS 24, the MCS with the highest data rate of IEEE 802.11ad standard. The experiment setup is shown in Figure 2.15. Random bitstreams are generated, coded and oversampled in a PC with Matlab. The sampled digital OFDM signals are passed to an Agilent 81180A arbitrary waveform generator which outputs I/Q signals at 1.5 GHz. The I/Q signals are fed to a SiversIMA FC1005V/00 mm-Wave upconvertor board [100] with a horn anntenna which has 25 dBi directional gain and 10°beamwidth [29]. Finally, the signals are transmitted over-the-air in 61.5 GHz band. At the receiver side, the received 61.5 GHz signal is down-converted by a SiversIMA FC1005V/00 board and processed by an Agilent 90804A Oscilloscope. Finally, the processed baseband signals are demodulated, decoded, and compared to the original bitstreams at the host-PC.

2.4.5.1 BER vs. SIR

We have first compared the BER in relation to SIR of the applied measurement system. The experimental setup is similar to Figure 2.15, but we allocate the interferer as close to the Tx device as possible, i.e., we set the angle θ to a very small value so that both the transmitter and the interferer point to the receiver and in the mainlobe of the receiver. By changing the power of the interferer, we manually set the SIR at the receiver. The results of the coded and uncoded BER are shown in Figure 2.16. As we consider a link stable when the coded BER is higher than a predefined threshold, e.g., 10^{-6}, the measurements with SIR values whose corresponding coded BER is lower than the threshold are not shown. It can be seen from the figure that in order to enable MCS 24, the MCS with the highest data rate of IEEE 802.11ad standard, the SIR should be at least 28 dB. Because the measurements in MI (cf. Subsubsection 2.4.3) require the same SIR based measurements, Figure 2.16 can also be used as a look-up-table for PCP/APs to determine the appropriate MCS of parallel transmissions if the SIRs of the considered links are known.

2.4.5.2 Topology vs. Data rate

We also measured the generated interference with different topologies. Three topologies are measured, namely Sidelobe to Sidelobe (SS) interference topology, the Sidelobe to Mainlobe (SM) interference topology, and the Mainlobe to Mainlobe (MM) interference topology. The topologies are named by the path of the interference, for example, SS interference topology means that the interfering signal are transmitted from the sidelobe of the transmitter and received by the sidelobe of the receiver. After a number of measurements, we have verified that the SS

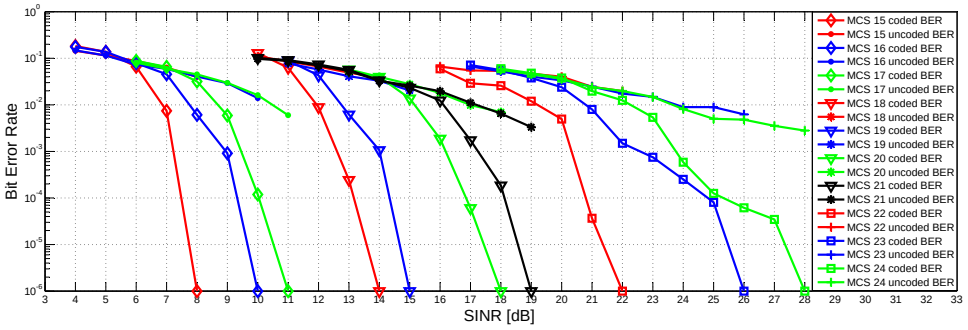


Figure 2.16: Coded BER and raw BER of IEEE 802.11ad OFDM PHY MCSs. One transmitter, one receiver and one interferer are used in this measurement. The transmitter and the interferer are allocated close by and turned in the direction of the receiver so the receiver has the LOS connections to both of them. The distances between the receiver and the transmitter/interferer are 3 m. SINRs are changed by changing the gain of the transmitting power of the interferer.

interference is neglectable for the used 25 dBi horn antenna, even when the interferer and the transmitter are placed 10 cm away from each other. Therefore, we designed the following measurements for investigating the relationship between the data rate degradation where the transmitter and receiver are located with the MS and MM interference topologies. In the measurement, we kept the positions of the transmitter and receiver same as the last measurements, i.e., 3 m away in the center of a typical indoor environment and force them to face to each other. We also kept the distance between the interferer and the receiver to 3 m with the interferer always pointed at the receiver, while changing the angle θ to change the interference strength. Coded BERs of different angles are measured and compared to the predefined threshold. The highest MCS at different angles is recorded in Figure 2.17. It is evident that in most of the cases, the interferer cannot generate harmful interference so that the communicating pair are able to achieve the highest data rate. However, when the interferer is entering the beam of the receiver, the interference increases significantly and the achievable data rate decrease rapidly.

The measurement results show that the used 60 GHz devices own good enough directivity for supporting co-channel parallel transmissions which is the key principle of SBMAC.

2.4.6 Simulation Results

In order to validate SBMAC and evaluate its performance in the environment with a large number of nodes, a number of simulations of an PBSS and a large number of STAs are carried out in a MATLAB simulator. In the following simulations, we assume that all the non-PCP/non-AP STAs are uniformly distributed in the range of a IEEE 802.11ad PBSS, while the PCP/AP is located at the center point. The

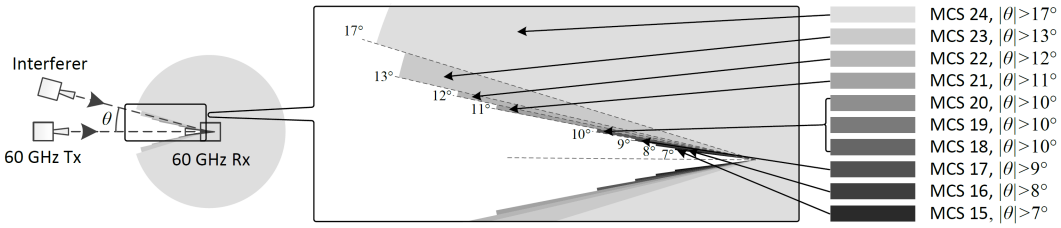


Figure 2.17: Maximal achievable MCSs vs. angle of interferer at the receiver in azimuth plane are measured. One transmitter, one receiver and one interferer are used in this measurement. The distances between the receiver and the transmitter/interferer are 3 m. The transmitter is allocated in the direction of 0°, and the interferer is allocated in the direction of θ. During the measurement, the transmitter and the receiver are always facing towards each other.

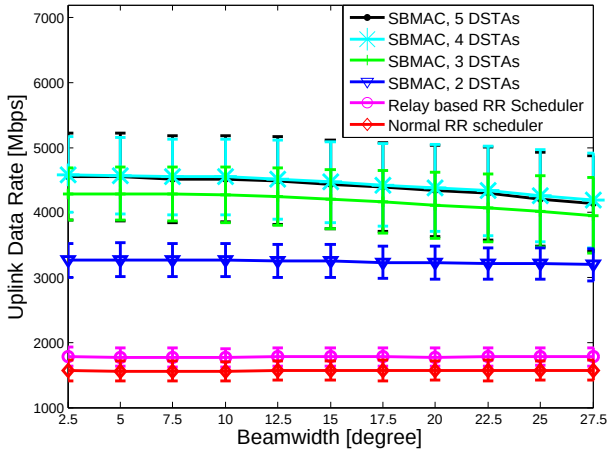


Figure 2.18: Uplink data rate with respect to the beamwidth of the directional antennas. The data rate of all STAs is 100 Mbps. The directional gain of the antenna is 25 dBi. 60 STAs are randomly allocated in the coverage of one PCP/AP. The directional gain of the antenna is 25 dBi and the beamwidth is 5°. The data rate of all STAs varies from 20 Mbps to 220 Mbps.

system configurations used in our simulations are selected according to the IEEE 802.11ad OFDM PHY and our measurement campaign in Subsection 2.4.5. In the simulations, we only consider LOS channels in a typical indoor environment with path loss exponent 2.25, which is set according to the channel model in [8] and the measurement results in [106]. All simulations have been repeatedly executed over 20000 times to ensure that the results are statistically reliable. In all simulations, we compare the SBMAC performance of the initialization BI to the original IEEE 802.11ad standard with the general RR scheduler (2.8)-(2.10) and a relay based scheduler. In the relay based scheduler, a cooperative weight vector which is same as the weight vector in [88] is used to determine relay transmissions.

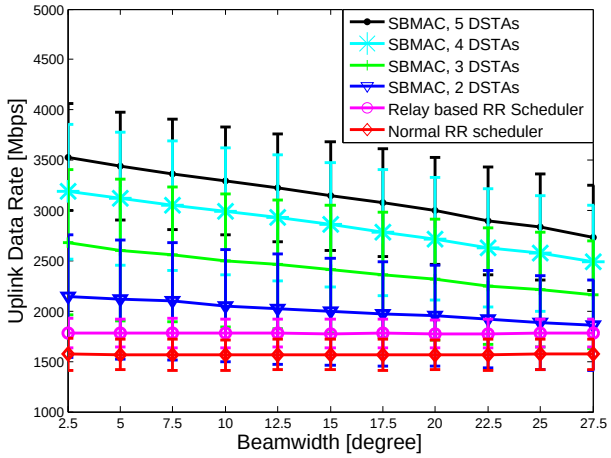


Figure 2.19: Uplink data rate with respect to the beamwidth of the directional antennas. The data rate of all STAs is 100 Mbps. The directional gain of the antenna is 10 dBi. 60 STAs are randomly allocated in the coverage of one PCP/AP. The directional gain of the antenna is 25 dBi and the beamwidth is 5°. The data rate of all STAs varies from 20 Mbps to 220 Mbps.

2.4.6.1 Beamwidth

In the first simulation, we evaluate the performance of SBMAC with different antenna beamwidths. Both Figure 2.19 and Figure 2.18 shows that the uplink data rate of the network decreases as the beamwidth increases. Due to the lower CCI caused by the sidelobes, the uplink data rate of the networks with a high directional gain outperforms the network with a low directional gain. The result shows that SBMAC outperforms conventional scheduler upto 3-fold with some well selected parameters. Moreover, the flat curves in Figure 2.18 where the performance only drops less than 10% when the beamwidth increases from 2.5 °to 27.5 °reveals that SBMAC performs constantly with different beamwidth. Therefore, SBMAC can be applied for wireless networks with different directional antenna technologies.

2.4.6.2 Source Data Rate

We also evaluate the performance of SBMAC with different source data rates at all non-PCP/non-AP STAs. As shown in Figure 2.20, when the source data rate is low, a PBSS with more DSTAs achieves higher data rates. For example, when the source data rate is 80 Mbps, the average uplink data rate of a PBSS with 3, 4, and 5 DSTAs outperforms the PBSS which has only 2DSTAs around 1000 Mbps. This is due to the fact that when the source data rate is low, the demands of carrying out concurrent transmissions is also low, which reduces the overall CCI level. Therefore, the PCP/AP should decide the number of DSTAs according to the overall load of the PBSS. Not surprisingly, SBMAC outperforms conventional methods at least 2-fold in most cases.

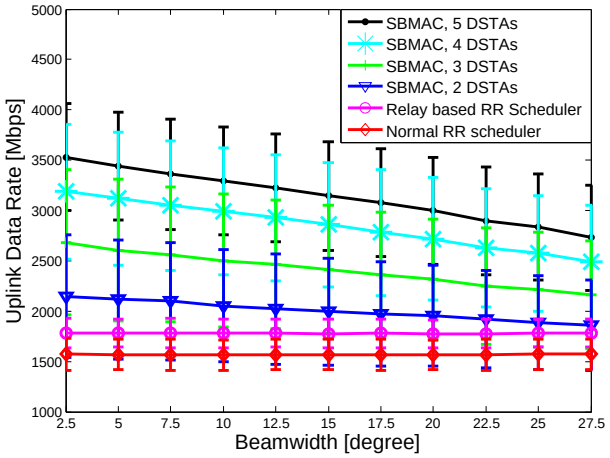
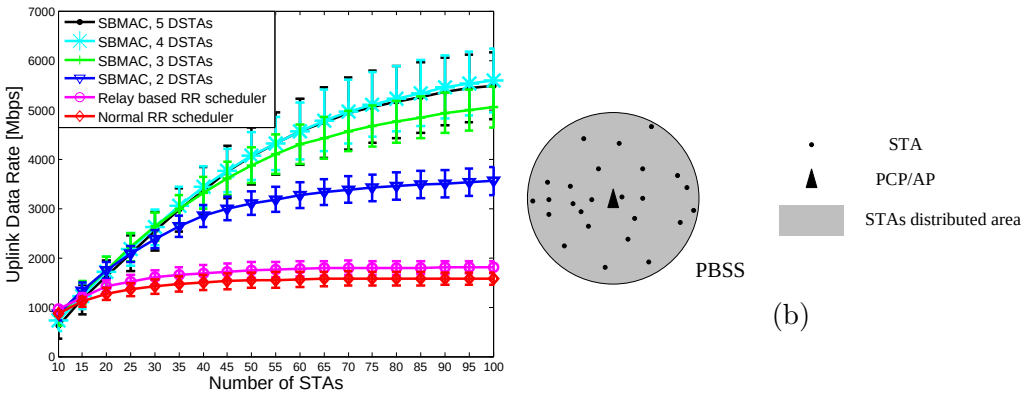


Figure 2.20: Uplink data rate with respect to the source data rate of each STAs. 60 STAs are randomly allocated in the coverage of one PCP/AP. The directional gain of the antenna is 25 dBi and the beamwidth is 5°. The data rate of all STAs varies from 20 Mbps to 220 Mbps.

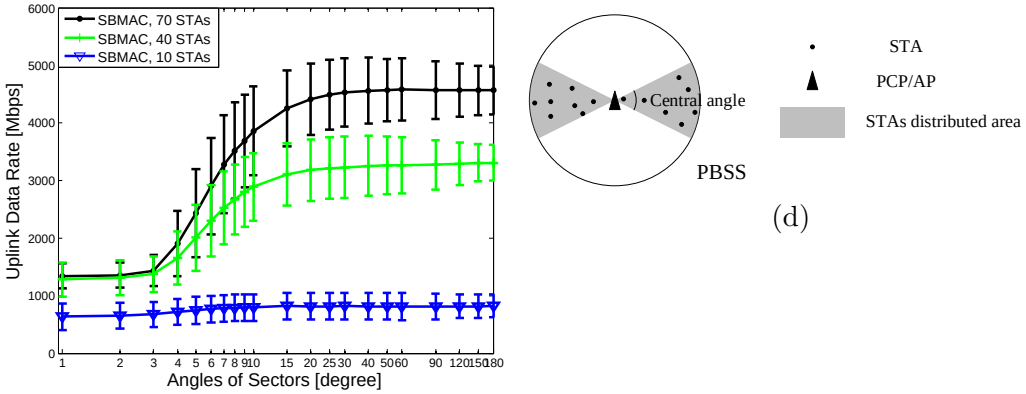


(a) (b)

Figure 2.21: (a) Uplink data rate with respect to the number of STAs. 10 to 110 STAs are randomly allocated in the coverage of one PCP/AP as shown in (b). The directional gain of the antenna is 25 dBi and the beamwidth is 5°. The data rate of all STAs is set to 100 Mbps. The directional gain of the antenna is 25 dBi and the beamwidth is 5°.

2.4.6.3 Number of STAs

We evaluate the uplink data rate performance with the number of STAs in one PBSS. All STAs are uniform distributed in the coverage of the PBSS as shown in Figure 2.21(b). Figure 2.21(a) shows that when the number of STAs is relatively small, networks with less DSTAs perform better because of their low CCI level. It is also owning to the fact that fewer STAs correspond to a lower demand for buffering data. If the number of STAs is large, then the importance of DSTAs is increased



(a) (d)

Figure 2.22: (a) Uplink data rate with respect to the source data rate of each STAs. 10, 30, 70 STAs are randomly allocated in the coverage of one PCP/AP which are composed by two sectors as shown in (b). The directional gain of the antenna is 25 dBi and the beamwidth is 5°. The directional gain of the antenna is 25 dBi and the beamwidth is 5°.

because more DSTAs can effectively reduce the number of unreachable NDSTAs, and offering higher capacity of buffering data. Besides, deploying more DSTAs can also enhance the uplink data rate of each DSTA group because NDSTAs are easier to be assigned to a nearby DSTA where they have better channels. The result proves that the performance of SBMAC outperforms the conventional schedulers significantly in terms of uplink data rate in all simulated cases.

2.4.6.4 Topology and Data Rate

At last we show the relations between network topology and performance. In a real deployment, the distribution of STAs is normally constrained by the environment such as the layout of a building or the frequent visiting area. For example, if the PCP/AP is located in the middle of a corridor and all STAs are distributed in the corridor, the coverage of the PBSS likes a rectangular or two sectors. To investigate the protocol performance in this typical environment, we randomly allocate STAs in two symmetric sectors, and very the angle of the sector from zero to half circle, i.e., two sectors cover the whole circle. An example of the simulated PBSS is shown in Figure 2.22(b). As shown in Figure 2.22(a), the uplink data rate increases with the sector angle, and converges to the maximal data rate when central angle is greater than the beamwidth. The main cause of the low uplink data rate when central angle is small is that the small central angle leads to more MM interference between parallel transmissions, which has been proved harmful to the data rate of parallel transmissions in the hardware verification section.

In the second topology-based simulation, we allocate STAs in two sectors with fixed central angle of 10 °, and change the angle between the two sectors. The simulation results is shown in Figure 2.23(a), and an example of the simulated PBSS is shown

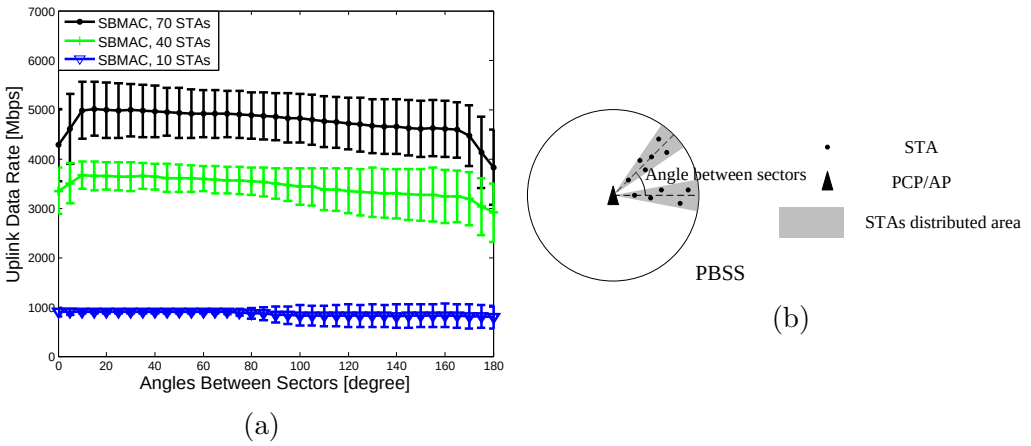


Figure 2.23: (a) Uplink data rate with respect to the source data rate of each STAs. 10, 30, and 70 STAs are randomly allocated in the coverage of one PCP/AP which are composed by two sectors as shown in (b). The directional gain of the antenna is 25 dBi and the beamwidth is 5°.

in Figure 2.23(b). We find that when the angle is smaller than or compatible to the beamwidth, the performance increases as the angle increases, and reaches the maximal value soon when it is greater than the beamwidth. It is due to the fact that small angles lead to more MM interference. We also can notice that the performance drops rapidly when the angle is approaching to 180° where the two sectors lay in the opposite sides of the PCP/AP. It is owing to the fact that when the two sectors are in opposite sides, their STAs are able to interfere each other. Despite of the significant effects of MM interference when angles are approaching to 0 or 180°, we also notice a general trend in the middle range which covers most of the angles that the overall uplink data rate decreases and its standard deviation increases as the angle between sectors increases. This is because the larger angles lead to longer average distances among STAs, which eventually decreases the data rate between STAs.

The above two simulations verified the theorem which are proposed in [84] that the performance of a directional wireless network is dependent on the topology. The conclusion can be used for future network planning and optimization.

2.4.7 Summary

In this section we proposed SBMAC, a directional MAC protocol which coordinates the uplink channel access in SPs of an IEEE 802.11ad PBSS. SBMAC makes a virtue out of the directional transmissions of IEEE 802.11ad standard to enhance the uplink date rate of the network. Measuring and analyzing the potential interference in SBMAC enables concurrent transmissions with maximum available data rate. We have presented measured results on a real 60 GHz hardware platform to verify the feasibility of SBMAC. We also presented detail simulation to show that SBMAC outperforms the conventional IEEE 802.11ad MAC in terms of uplink data rate

by up to 3-fold. In our future work, we plan to apply advanced machine learning techniques for enhancing the DSTA selection and NDSTA grouping in the long term, thereby further improve the performance of the proposed SBMAC protocol.

2.5 Directional Frame Aggregation

Frame aggregation was first introduced in IEEE 802.11n, and is inherited by the later amendments. It allows the transmitter to send multiple independent frames back-to-back in one channel access. By grouping frames together and using a single PHY header, frame aggregation can greatly reduce the overhead. The deployment of frame aggregation in the IEEE 802.11ad standard is given in Subsection2.1.4.

Here, we jointly consider the directivity of mmW-communication technologies and the frame aggregation scheme. We propose CTMAC which enables the parallel transmissions even when the STAs are inside the mainlobes of each other. By using the localization and beamforming information, CTMAC utilizes the receiving time of the receiver of an ongoing communicating pair which is significantly prolonged by the frame aggregation, to let another pair of STAs also transmit even in the main beam of the former transmitting STA. In this study, the CTMAC principle is applied and evaluated on top of the IEEE 802.11ad standard. However, it can be widely used by any mm-Wave communication systems, including uncoordinated random access protocols and 5G radio access networks. To the best of our knowledge, this is the first work that explicitly allows parallel transmissions within the mainlobes of the communicating STAs.

The structure of an IEEE 802.11ad frame which is named as PHY layer Protocol Data Unit (PPDU) in the standard is shown in Figure 2.24(a). It consists of a PHY Layer Convergence Procedure (PLCP) preamble, a PLCP header, a MAC header, a MAC Service Data Unit (MSDU) which is the MAC layer payload, and a checksum sequence. In order to show the transmission efficiency, we use the darker shaded blocks to represent the useful payload, and lighter blocks to mark the control overhead. As shown in the Figure 2.24(a), the overhead of transmitting one MSDU without aggregation is high. Frame aggregation is proposed to reduce the average overhead for each MSDU by grouping them together and then transmitting in one channel access. There are two types of frame aggregation in IEEE 802.11 standards, namely aggregated MAC Protocol Data Unit (AMPDU) and aggregated MSDU (AMSDU). AMPDU is done by first adding a MAC header and generating the checksum sequence to each MSDU to create an MPDU which is the payload of PHY layer, and then transmits multiple MPDUs in a single PPDU. Unlike the AMPDU, AMSDU allows the transmitter to send multiple MSDUs to the same receiver in a single MPDU, i.e., the aggregated MSDUs are transmitted by using only one PHY layer header, one MAC layer header and being proved by one MAC layer checksum sequence.

The most significant difference between AMSDU and AMPDU is the time of performing the MAC layer encapsulation [115]. For AMSDU, the MAC layer

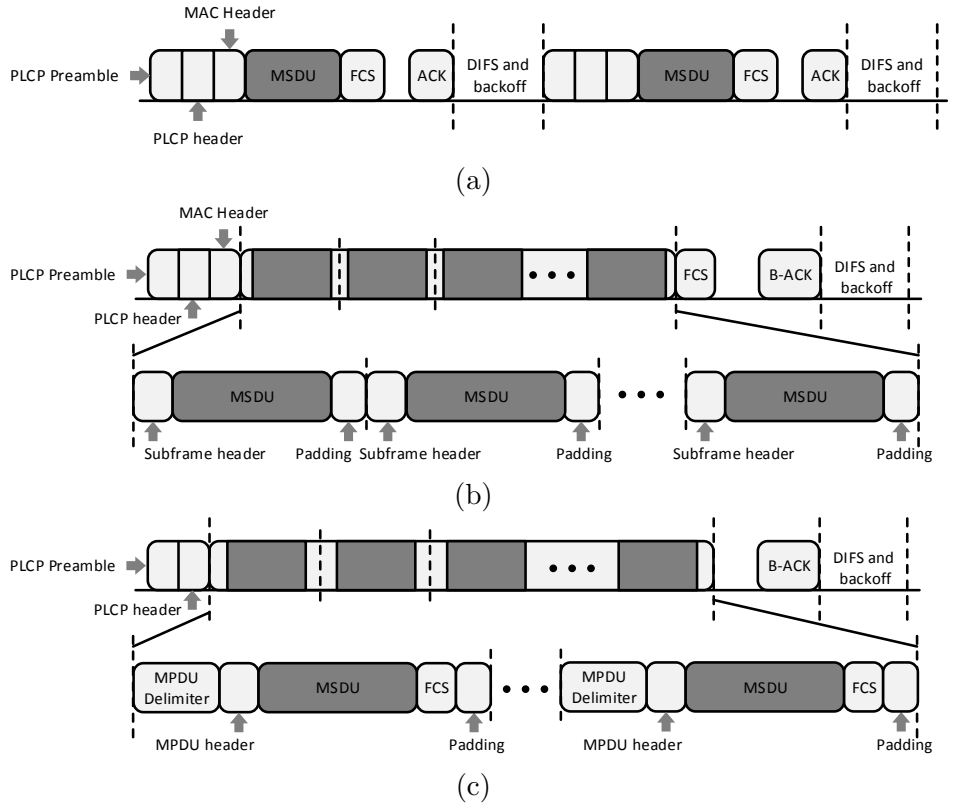


Figure 2.24: Frame format of IEEE 802.11ad standard. (a) frame format of normal MPDU; (b) frame format of AMSDU; (c) frame format of AMPDU.

encapsulation is done after the aggregation, therefore all encapsulated MSDUs has the same destination address and one checksum sequence. On one hand, its compact structure has a high channel utilization ratio. On the other hand, it has low resistance to the bit error rate because the packet error results in retransmit all MSDUs in the failed AMSDU. In contrast, due to the separately encapsulated MPDU, AMPDU only need to retransmit the collapse MPDU after checking the MPDU individually. Therefore, it is proved in a number of literatures that AMPDU performance better than AMSDU in error-prone wireless channels. [12, 19]. The comparison of AMSDU and AMPDU is out of the scope of this study, so we only analyze the frame aggregation scheme by using AMPDU in the rest of this study.

2.5.1 Interference analysis

We select a widely used flat-top antenna model [64, 90] which has a mainlobe and sidelobes with unique gain as the antenna model in this study. As shown in Figure 2.25, the flat-top antenna has a mainlobe with beamwidth θ and gain $G_{\mathrm{M}} = \eta\frac{2\pi}{\theta}$. The gain out of the main lobe is $G_{\mathrm{S}} = (1-\eta)\frac{2\pi}{2\pi-\theta}$. In this antenna model, the radiation efficiency η [3] is defined in [64, 90] to describe the difference

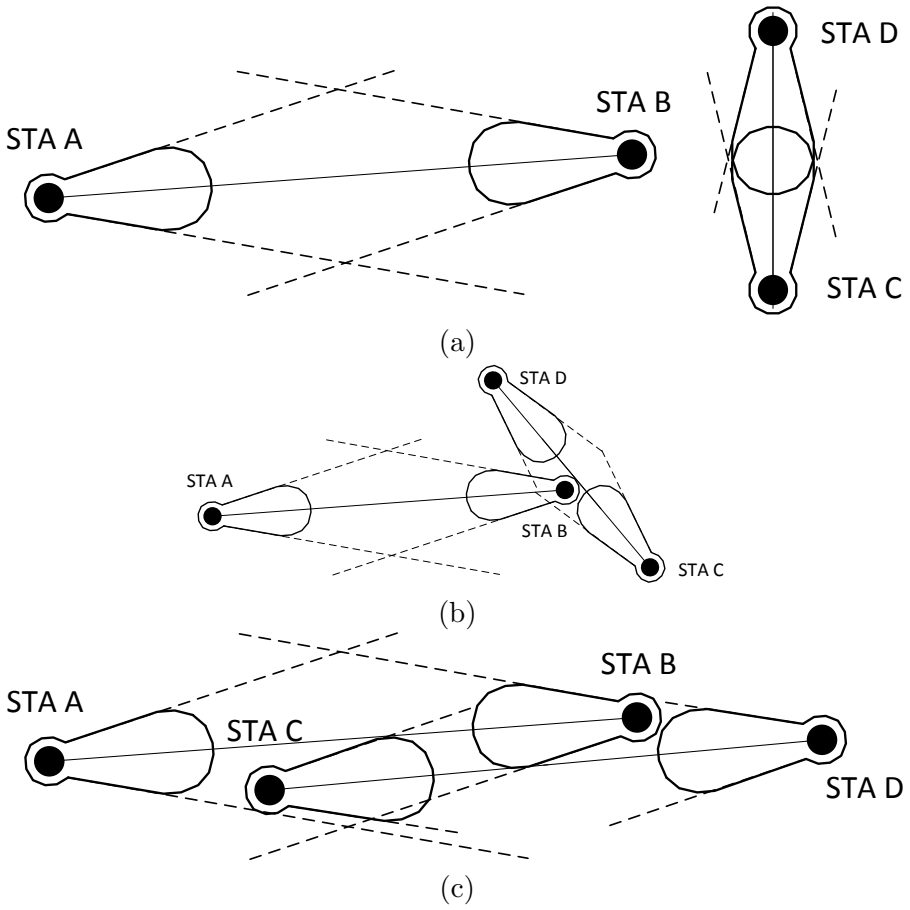


Figure 2.25: Three types of interference which are generated by parallel transmissions. (a) sidelobe-sidelobe interference; (b) sidelobe-mainlobe interference; (c) mainlobe-mainlobe interference.

between the gain in the mainlobe and the gain in the sidelobes. According to the antenna model, we can divide the interference scenarios into three types, namely SS interference, SM interference, and MM interference (cf. Figure 2.25). Most of the existing directional MAC studies allow STAs which are located with SS and SM topologies to carry out concurrent transmissions [79, 84] because the received signal power of MM connections are much higher than the interference power from SS connection or SM connections. For example, consider STA A is transmitting to STA B while STA C is also transmitting to STA D in all three cases in Figure 2.25. If the antenna training is performed appropriately, the antenna gains of the useful signal transmissions are always the gain of the mainlobe. In this case, the received

[3] Even through the term radiation efficiency is defined in IEEE Standard for Definitions of Terms for Antennas [4] as the ratio of the total power radiated by an antenna to the net power accepted by the antenna from the connected transmitter, we stick to the definition of radiation efficiency in [64, 90] to improve the readability.

signal power P_{s} at STA B is

$$P_{\mathrm{s}} = G_{\mathrm{AB}} G_{\mathrm{BA}} P L_{\mathrm{AB}} = 4\pi^2 \frac{\eta^2}{\theta^2} \mathrm{PL}_{\mathrm{AB}}, \tag{2.11}$$

where $\mathrm{PL}_{\mathrm{AB}}$ denotes the pathloss between STA A and STA B. The received interference power P_{i} depends on the directions of the interfering STAs. According to the three topologies shown in Figure 2.25, the interference power at STA B which generated from STA C is,

$$P_{\mathrm{i}} = \begin{cases} 4\pi^2 \dfrac{(1-\eta)^2}{(2\pi-\theta)^2} \mathrm{PL}_{\mathrm{CB}}, & \text{Fig. 2.25(a)} \quad (2.12) \\ 4\pi^2 \dfrac{\eta(1-\eta)}{\theta(2\pi-\theta)} \mathrm{PL}_{\mathrm{CB}}, & \text{Fig. 2.25(b)} \quad (2.13) \\ 4\pi^2 \dfrac{\eta^2}{\theta^2} \mathrm{PL}_{\mathrm{CB}}, & \text{Fig. 2.25(c)} \quad (2.14) \end{cases}$$

Thus, the corresponding SIR[4] is

$$\mathrm{SIR} = \begin{cases} \dfrac{\eta^2(2\pi-\theta)^2}{(1-\eta)^2\theta^2} \dfrac{\mathrm{PL}_{\mathrm{AB}}}{\mathrm{PL}_{\mathrm{CB}}}, & \text{Fig. 2.25(a)} \quad (2.15) \\ \dfrac{\eta(2\pi-\theta)}{(1-\eta)\theta} \dfrac{\mathrm{PL}_{\mathrm{AB}}}{\mathrm{PL}_{\mathrm{CB}}}, & \text{Fig. 2.25(b)} \quad (2.16) \\ \dfrac{\mathrm{PL}_{\mathrm{AB}}}{\mathrm{PL}_{\mathrm{CB}}}, & \text{Fig. 2.25(c)} \quad (2.17) \end{cases}$$

In principle, the directional antennas with high directivity, e.g., the standard gain horn antenna with 25 dBi [29], can provide sufficient SIR to support parallel transmission for SS and SM interference topologies.

A number of studies show that IEEE 802.11 devices are able to find the MCS which can maximize the usage of the SIR and control the packet error rate. For example, Halperin et al. in [20] proposed a packet delivery prediction method. It shows that the transmission parameters, including the MCS, can be accurately predicted based on the channel measurements. Their algorithm is verified on commodity 802.11 Network Interface Controllers (NICs) based on only the channel information which is read from the device. At least, the rate adaptation algorithm can also be performed between BIs so as to find the best rate in the long run. The detailed method of mapping the SIR and MCS are out of the scope of this study. We assume that after beamforming training and all necessary measurements, PCP/AP knows the appropriate MCS when parallel transmissions are carried out.

It is worth noting that if the interference takes place via MM connections, the SIR (cf. (2.17)) is normally too low for carrying out valid concurrent transmissions if the

[4] Due to the relatively high interference level, noise is ignored in this study

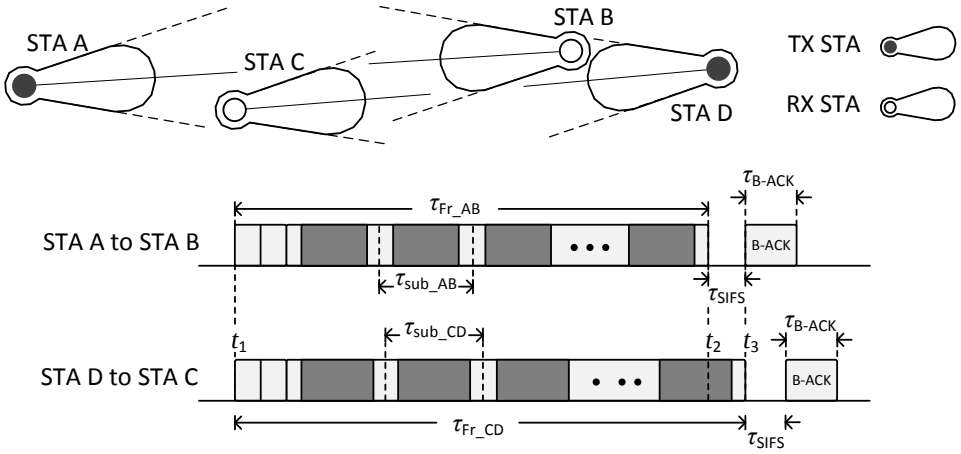


Figure 2.26: A typical scenario of mainlobe-mainlobe interference and the enabled parallel transmissions.

pathloss of the signal link and interference link are at the same level. To the best of our knowledge, no study explicitly allows STAs which are interfering each other with MM connections to transmit at the same time.

2.5.2 CTMAC principle

CTMAC enhances the IEEE 802.11ad by enabling the concurrent transmissions in the main beam. Although CTMAC is targeted for the IEEE 802.11ad standard, its principle can be used for any wireless networks which have good directivity. it is well known that RTS-CTS-DATA-ACK four-way handshakes is not efficient in directional only networks because of the repetitively transmitted directional RTS and the prolonged carrier sensing time [77,84]. Therefore, only the DATA-ACK two-way handshake is considered in this chapter. A typical MM interference topology is shown in the upper part of Figure 2.26, where two pairs of STAs are going to communicate. We first assume that STA A are scheduled to transmit to STA B with frame aggregation. As shown in the lower part of Figure 2.26, STA A starts transmitting to STA B at t_1, and stops transmitting at t_2. Afterwards STA B replies with a block ACK frame which contains the acknowledgements of all MPDUs in the received AMPDU. The block ACK starts being transmitted at t_3 and stops at t_4. The duration $\tau_{\text{Fr_AB}}$ is the transmission duration of the AMPDU, and $\tau_{\text{B-ACK}}$ represents the transmission duration of the block ACK.

According to the interference topology, the AMPDU transmission from STA A and block ACK transmission from STA B generate different levels of interference to their neighboring node. STA A causes SM interference towards STA C and MM interference towards STA D. Because STA B is receiving, STA C may cause MM interference and STA D may cause SM interference towards STA B if they start transmitting. After transmitting AMPDU, STA B transmits a block ACK back to STA A. During the block ACK transmission, STA B causes MM interference towards

STA C and causes SM interference towards STA D. If we follow the same principle that only allows parallel transmissions under the SS and SM interference scenarios, no transmission between STA C and STA D is able to be carried out. However, if the transmission between STA D and STA C are carried out following the lower part of Figure 2.26, parallel transmissions are feasible because no MM interference is generated between any two STAs. This way, the throughput is significantly increased.

2.5.3 Adaptive frame aggregation

Transmitting with the highest possible MCS can optimize the performance of the parallel transmitting STAs. However, STA pairs may transmit with different MCSs due to the different SIR values. Therefore, if two transmitting STAs transmit the same number of MPDU with different data rate, one STAs may stop first and wait for transmitting its block ACK frame. Obviously, it wastes of spectrum resources. Hence, the most efficient strategy is that we select the number of MPDU so as to minimize the differences between the transmission time of both pairs. If the number of MPDU of one pair N_1 is given, then the number of MPDU of another pair N_2 can be calculated as,

$$N_2 = \underset{N_2}{\operatorname{argmin}} |\tau_{\text{Fr_AB}} - \tau_{\text{Fr_CD}}| \tag{2.18}$$

$$= \underset{N_2}{\operatorname{argmin}} |N_1\tau_{\text{sub_AB}} - N_2\tau_{\text{sub_CD}}| \tag{2.19}$$

$$= \underset{N_2}{\operatorname{argmin}} |N_1\frac{L_{\text{MPDU}}}{R_1} - N_2\frac{L_{\text{MPDU}}}{R_2}| \tag{2.20}$$

where $\tau_{\text{Fr_AB}}$ and $\tau_{\text{Fr_CD}}$ are the transmission time of a AMPDU sent by STA A and STA D, respectively. According to the SIR measurement results, the data rate of STA A and STA D are R_A and R_D. The length of MPDU is L_{MPDU} bits.

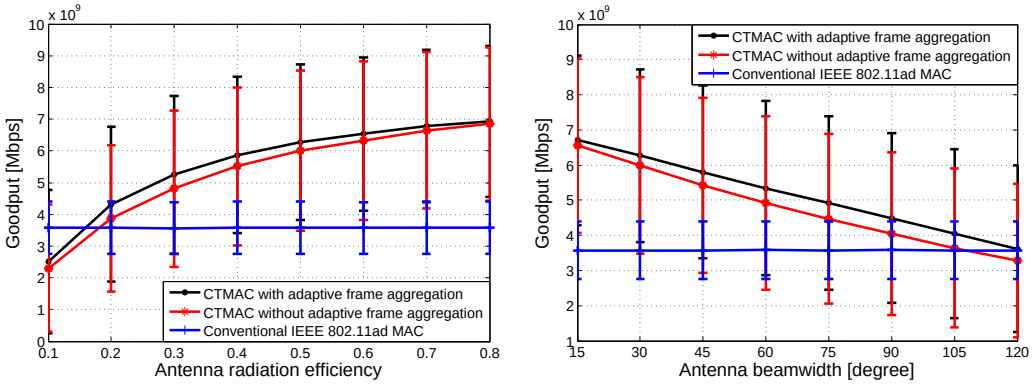
2.5.4 Simulation Results

In order to evaluate the performance of CTMAC, a number of simulations are carried out in a customized MATLAB simulator. The system configurations used in our simulations are selected according to the IEEE 802.11ad OFDM PHY specifications. In the simulations, we only consider LOS channels in a typical indoor environment with path loss exponent 2.25, which is set according to the channel model in [8] and the measurement results in [106]. The SNR and MCS mapping are set according to the results in [108] where the single input single output (SISO) packet error rate (PER) performance of all 802.11ad MCS versus the SNR for the LOS channels are given. In the following simulations, STAs are allowed to communicate with a certain MCS when the corresponding PER is lower than 1%. AMPDU scheme is applied in all simulations with N sub MPDUs. All simulations were repeated more

Table 2.2: Simulation Parameters

T_c SC chip time	0.57 ns
T_{preamble}	$3328 \times T_c$
PLCP header	64 bits
MAC header	288 bits
AMPDU delimiter	32 bits
Padding	16 bits
MSDU length	7920 Bytes
ACK payload	$8 \times N^*$ Bytes
t_{SIFS}	3 μs
t_{DIFS}	13 μs
OFDM PHY data rate	MCS 15-24

* Number of MPDUs in AMPDU

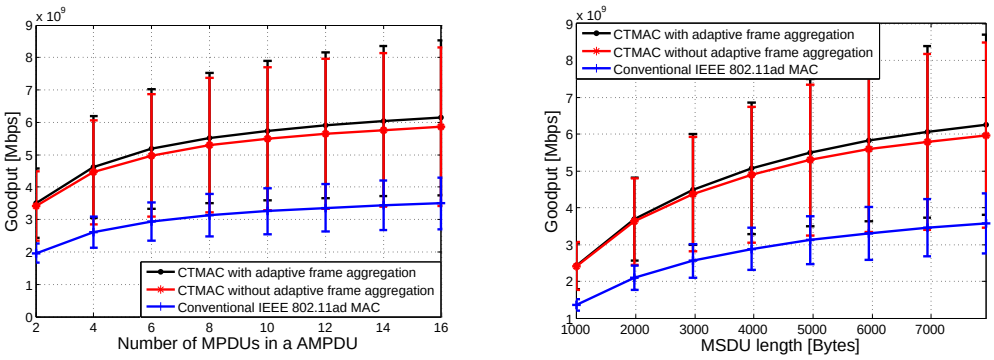


(a) (b)

Figure 2.27: (a) Achieved goodput with respect to the antenna radiation efficiency. The beamwidth of the antenna is 30 °, the number of MPDUs in a AMPDU is 20. (b) Achieved goodput with respect to the antenna beamwidth. The beamwidth of the antenna is varying from 15 ° to 120 °. The radiation efficiency is 0.5, and the number of MPDUs in a AMPDU is 20.

than 10000 times to ensure the stationarity. In all simulations, we compare the CTMAC performance with and without adaptive frame aggregation algorithm and the original IEEE 802.11ad standard which allows no parallel transmissions. Several key parameters used in the simulations are listed in Table 2.2.

Because the antenna directivity and the length of frame aggregation are the two key factors that can impact CTMAC performance, we performed four simulations to investigate the relations between CTMAC and them. In all simulations, the goodput of two pairs of STAs which are in the mainlobes of each other as shown in Figure 2.26. We assume that STA A and STA D has infinite data to transmit to STA B and STA C, respectively. We randomly allocate two pairs of STAs in a line and ensure that all STAs are in the communication range of each other with the lowest MCS under the MM topology.



(a) (b)

Figure 2.28: (a) Achieved goodput with respect to the number of MPDUs in a AMPDU. The beamwidth of the antenna is 30 °. The radiation efficiency is 0.5, and the number of MPDUs in a AMPDU is varying from 2 to 16. (b) Achieved goodput with respect to the length of MSDU. The beamwidth of the antenna is 30 °. The radiation efficiency is 0.5, and the number of MPDUs in a AMPDU is 20. The length of the MSDU is varying from 990 Bytes to 7920 Bytes.

The directivity of the deployed antenna can significantly impact CTMAC performance. High gain in the mainlobe may enhance the SIR at the receiving STA and low gain in the sidelobe can reduce the interference level. In the first simulation, we evaluate the performance of CTMAC with different antenna radiation efficiencies which is defined in Section 2.5. As shown in Figure 2.27(a), the goodput of CTMAC increases fast as the radiation efficiency increases and the enhancement is converging to 2-fold if the radiation efficiency is close to 1, i.e., nearly no transmission power is able to leak through the sidelobes. In this case, nearly no interference can be received through sidelobe so the two pairs are almost communicating independently. We also see that when the radiation efficiency is too small, e.g., less than 0.1, the performance of CTMAC is lower than the original IEEE 802.11ad MAC because the high interference level.

We also evaluate the performance of CTMAC with different beamwidth which is another important factor of antenna directivity. As shown in Figure 2.27(b), the goodput of the two pairs of STAs decreases rapidly as the beamwidth increases. It is owing to the fact that large beamwidth leads to small directional gain and high interference level through sidelobes. We also notice that the performance of CTMAC is worse than the original IEEE 802.11ad MAC when the beamwidth is greater than unit70° which is greater than most of the COTS mm-Wave antennas.

The number of subframes in the frame aggregation scheme is another key configuration of CTMAC. We carried out a simulation to investigate the relations between the number of MPDUs in a AMPDU and the goodput of two STA pairs. As shown in Figure 2.28(a), the performances of the original IEEE 802.11ad MAC and CTMAC increase as the number of MPDUs increases. It's due to the fact that more MPDUs can lower down the overhead in both PHY layer and MAC layer. It is worth noting that in the realistic deployment, the length of frame aggregation scheme

should be determined by jointly considering the channel conditions and packet error rate together.

In the last simulation, we evaluate the performance of CTMAC with the different MSDU length. As shown in Figure 2.28(b), the performance of all tested schemes increase when the MSDU length increases. The performance of CTMAC dominates the original IEEE 802.11ad MAC with all simulated MSDU lengths.

2.5.5 Summary

CTMAC enables STAs to carry out parallel transmissions even when their mainlobes are pointing to each other. By using a packet level scheduling algorithm with an adaptive frame aggregation scheme, CTMAC completely avoids the interference between mainlobes and optimizes the data transmission. This way, CTMAC achieves the parallel transmissions even under the strongest interfered topologies. It proves that directional transmission technologies have a great potential to enhance the spatial reuse of mm-Wave wireless networks. The numeric results show that CTMAC can significantly outperform the original protocol up to 2-fold in throughput.

3 Multichannel Cognitive MAC protocol

Cognitive radio technology is an attractive solution to increase spectrum utilization efficiency and enable cooperation between different user groups. One of the basic tenets in this concept is that the users are divided to Primary User (PU) and Secondary User (SU). The primary users have a priority for the spectrum due to a licence or other arrangement, while the secondary users can access spectrum opportunistically. For recent reviews, we refer the reader to [11, 17, 43]. One of the key requirements for cognitive communications is to have spectrum aware cognitive MAC protocol. Besides classical cognitive radio operations, enabling efficient spectral co-existence of multiple networks in the ISM band can be also targeted by many cognitive MAC protocols. A number of cognitive MAC protocols have been proposed in recent years [5, 86]. In order to carry out secondary transmission while tracking the activity of PU, most of the works include functionalities such as spectrum sensing, PU transmission detection, and coordination among SUs.

One approach to enhance cognitive radio multiple access is to introduce Multichannel Preamble Reservation (MPR) schemes as done in [48]. We upgrade MPR scheme to a new Multichannel Multiframe Transmission (MMT) scheme and extend such approach by introducing Multichannel Virtual Carrier Sensing (MVCS) scheme with strict time controls, and call this new MAC scheme CogMAC+ [82, 83]. It is an updated version of CogMAC [48] which was proposed by the our colleagues with revolutionary design and significant improvements. The approach improves performance in throughput and allows better control on channel access and energy savings, which are important for different Machine to Machine (M2M) and Internet of Things (IoT) applications using cognitive radios.

To evaluate CogMAC+ and prove the compatibility of CogMAC and CogMAC+, we have implemented both protocols in a SDR testbed using LabVIEW [2] and Universal Software Radio Peripheral 2 (USRP2) [26]. In order to verify the validity of LabVIEW-USRP2 testbed, we carry out a number of baseline comparison on both Wireless Open Access Research Platform (WARP) [7] and LabVIEW-USRP2 testbed. In order to provide a comprehensive insight into the new features of CogMAC+, a number of experiments are carried out with respect to a number of key parameters. We use also realistic spectrum models that are statistically characterized from practical measurement campaigns to emulate the behaviors of external transmissions. Experimental results show that CogMAC+ is able to offer reliable data communication services through opportunistic medium access while keeping the interference caused to PU under the permitted level. Especially for unicast and multicast scenarios, CogMAC+ significantly enhances the system performance compared to CogMAC by enabling the parallel transmission. The rest of this chapter is organized as follows. We review related work in Section 3.1

and describe CogMAC+ in Section 3.2. Afterwards, we briefly introduce the experimental setups in Section 3.3. The experimental results are analyzed in Section 3.4. The chapter is finally concluded and future work is discussed in Section 3.5.

3.1 Related Work

A number of cognitive MAC protocols have been proposed in the literature [11, 17, 42, 43]. In general, two major problems have to be addressed by a cognitive MAC protocol, first how to gather necessary information of the radio environment and other users around and second how to utilize the spectrum resources, i.e., determining at what time, in which frequency band, and by what strategy to carry out transmission. The first issue has been widely addressed by adopting different CSMA based schemes [96, 103]. The listen-before-talk principle enables the user to verify the absence of other ongoing traffic in the shared frequency band down to the spectrum sensing limit of the devices used, of course. Coordination mechanisms are used to address the hidden node problem. Thus coordination allows users to share their observations and decisions to improve the overall network performance [98]. Park et al. proposed a measurement reporting MAC protocol called as Truncated Time Division Multiple Access (TTDMA) in [53] which enables dynamic control information exchange. The second problem is addressed in centralized protocols [11,16,17] by using a central coordinator. However, the wireless networks often have a decentralized nature in a sense that observation and utilization are all performed individually by distributed terminals. Therefore the environment observation and control information have to be exchanged among central coordinator and users which inevitably causes delay and bandwidth wastage. Distributed protocols, which require no cooperative infrastructure, are widely used in wireless ad hoc networks. In distributed protocols, the major challenge is to coordinate users in a multichannel environment so that different users are able to receive transmission of other users in different channels seamlessly. This is called as rendezvous problem [56,65,93]. Song et al. proposed a fully distributed CSMA based broadcast protocol BRACER [113], which addresses the rendezvous problem by downsizing the set of deployed channels and designing the corresponding broadcasting patterns and scheduling algorithm. Relying on the neighboring location information, it achieves high successful broadcast radio with variant broadcast delays.

A number of protocols address rendezvous problem by assuming the existence of a dedicated Common Control Channel (CCC). Intuitively, using dedicated CCC is an effective way to exchange control information for SUs which enables them to coordinate their behaviors. In this category, we can further divide the protocols into two subcategories, licensed CCC based protocol [50, 70, 97] and unlicensed CCC based protocols, e.g., C-MAC [15]. However, the assumption of availability of licensed CCC is not always easy to hold for SUs in licensed bands due to spectrum allocation policies. For unlicensed CCC based protocols, the problem is that the commonly available channel may not always simultaneously exist for

all SUs which have different locations and channel characteristics. Moreover, how to schedule transmissions in the CCC is another challenge because the CCC may become saturated [86]. Without a dedicated CCC, the rendezvous problem becomes challenging in multichannel scenario where nodes only equipped with a half-duplex radio. RODMAC [22] aims at improving CogMAC through an on-demand-channel hopping scheme to allow simultaneous transmission. However, highly important issue of external transmission is not considered and evaluated in the case of RODMAC. Moreover, the control information in RODMAC is exchanged by BEACON packets from server node to client node in an unlicensed CCC, thus the communication of control information is not necessarily reliable.

Majority of the studies mentioned above are evaluated through simulation and theoretical studies. Numerous studies have shown that simulation and theoretical studies fail to depict the reality of practical wireless communication systems, especially due to inaccurate or non-realistic models of the real wireless channels and physical layer behavior [28, 52, 60, 102]. Therefore, it is highly important to evaluate cognitive MAC protocols in actual deployment scenarios. SDR platforms follow the hardware/software co-design architectures which allow them to provide flexibility and reconfigurability to cognitive MAC protocol prototyping. Furthermore, SDR platforms provide realistic physical functionalities which enables access to the real wireless channels. Currently, many SDR platforms are available with different architectures and features [66], and an increasing number of cognitive MAC protocols have been verified through them. One of them is CogMAC protocol that has been implemented on WARP board [47, 48]. It is a decentralized CSMA-based cognitive MAC protocol that achieves reliable transmission by considering the existence of external transmissions, i.e., PU transmission and other secondary transmissions. CogMAC is designed for wireless nodes equipped with a single half-duplex radio interface, i.e., it can only access one of the available channels for transmission or reception at a particular time. CogMAC uses a MPR scheme to solve the rendezvous problem without a CCC or cooperative infrastructures. It uses the CSMA based principle to avoid interruption of ongoing transmissions. Before attempting to transmit a frame, the transmitting node first needs to sense all the channels to ensure that no other transmission is carrying out. After selecting a channel, it repetitively sends a frame back-to-back for long enough duration so that all receiving nodes in its transmission range can receive it. At the receiving node, the protocol uses multichannel carrier sensing method in which a node sequentially senses all available channels. When a valid transmission is detected, the receiving node stays in the channel until all frames have been received. In this manner, the repetitive frame transmission implicitly synchronizes all receiving nodes.

3.2 CogMAC+ Design

Our CogMAC+ protocol is a significantly enhanced version of CogMAC. CogMAC+ upgrades the MPR scheme of the previous CogMAC protocol by adopting a stricter timing analysis, and a more accurate carrier sensing scheme. Several frame based

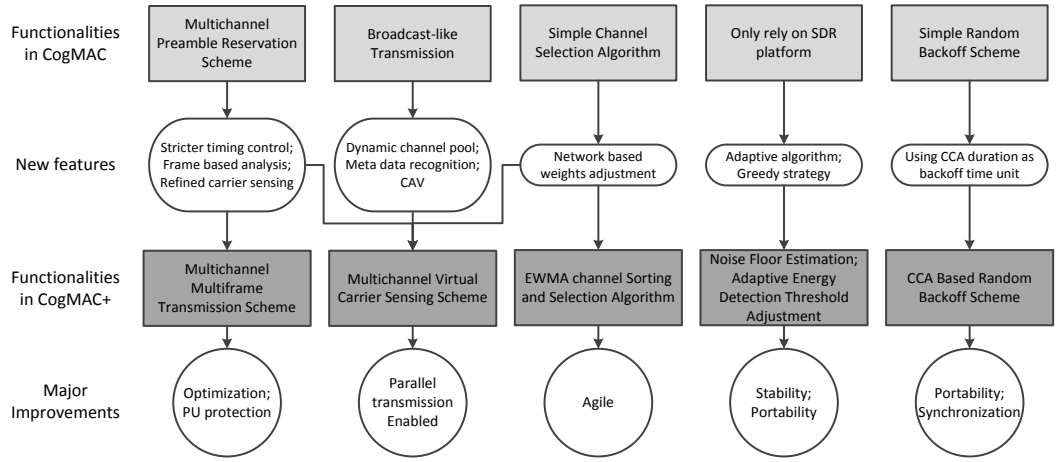


Figure 3.1: Functionalities comparisons of CogMAC and CogMAC+ and the major improvement of CogMAC+.

bounds are derived in the new scheme which optimize the secondary transmissions while achieve better PU protection. Moreover, its accurate timing control enables idle SUs to estimate the transmission duration of the transmitting SUs. Therefore, parallel transmissions are able to be carried out in different channels according to the meta-data of ongoing transmissions to substantially enhance the overall network performance. Besides, CogMAC+ includes an adaptive noise floor estimation and energy detection mechanism which significantly enhances the stability and portability of its implementation on different SDR platforms. We enhance the channel selection algorithm by using an Exponential Weighted Moving Average (EWMA) scheme. The portability and the synchronization of CogMAC+ is further improved by the new backoff scheme. The new functionalities added into CogMAC+ are shown as the darker shaded blocks in Figure 3.1.

In following we briefly review the principle of MPR scheme and its multichannel carrier sensing method then analyze the timing parameters of the protocol and introduce our MMT scheme. In Subsection 3.2.3, we describe the virtual carrier sensing scheme to illustrate how parallel transmissions are scheduled based on the timing information from MMT scheme. Subsection 3.2.4 shows the noise floor estimation and clear channel assessment algorithms, and how these algorithms stabilize the hardware implementation. In Subsection 3.2.5, the EWMA based channel weighting algorithm is introduced. Subsection 3.2.6 presents the random backoff scheme of CogMAC+. We list the notations and their corresponding explanation in Table 3.1.

3.2.1 Multichannel Preamble Reservation

MPR scheme is adopted to solve the rendezvous problem without a dedicated CCC. In order to guarantee seamless transmission, it requires the receiving node

Table 3.1: Some Notations Used in CogMAC+.

Variable	*Meaning*
C	Total number of channels
$\boldsymbol{C} = \{1, ..., C\}$	Set of all channels
$c, c \in \boldsymbol{C}$	Currently used channel
l	Weight threshold for removing a channel from pool
m	Reversed counter for frame repetition
n	Reversed frame index
N_{Fr}	Total number of multiframes in a transmission
$N_{\text{Re_Fr}}$	Number of frame repetition
N_{PU}	Number of frames in one PU allowance interval
N_{sa}	Number of samples for noise estimation
$N_{\text{de_sa}}$	Number of samples for frame detection
P	Targeted channel pool size without CAV
$\boldsymbol{P} = \{1, ..., P\}$	Set of sorted channel pool
q^+	Energy detection threshold increment
q^-	Energy detection threshold decrement
$r_d, d = 1, ..., N_{\text{sa}}$	Detected energy of the dth sample
$R_c, c \in \boldsymbol{C}$	Detected energy of the cth channel
$S_c, c \in \boldsymbol{C}$	Channel status of the cth channel
$T_c, c \in \boldsymbol{C}$	Energy detection threshold in the cth channel
t_{CCA1}	CCA1 carrier sensing duration
t_{CCA2}	CCA2 carrier sensing duration
t_{PU}	PU allowance time interval
t_{probing}	Time interval between two successive CCA2
$t_{\text{TX_mode}}$	Time required to switch to transmission mode
$t_{\text{RX_mode}}$	Time required to switch to reception mode
t_{switch}	Channel switching duration
$t_{\text{inter_Fr}}$	Inter-frame time duration
t_{header}	Header transmission duration
t_{Fr}	Frame transmission duration
$t_{\text{Re_Fr}}$	Repetitive transmission duration
$t_{\text{Mul_Fr}}$	Frame transmission duration after repetition
$t_{\text{inter_Fr}}$	Inter-frame transmission interval
$W_c, c \in \boldsymbol{C}$	Weight of the cth channel
$w_c(t), c \in \boldsymbol{C}$	Weight adjustment of the cth channel at time t
$\tau_c, c \in \boldsymbol{C}$	Timestamp for the last update of W_c
α	Decay factor of noise floor estimation algorithm
λ	Decay factor of channel weighting algorithm

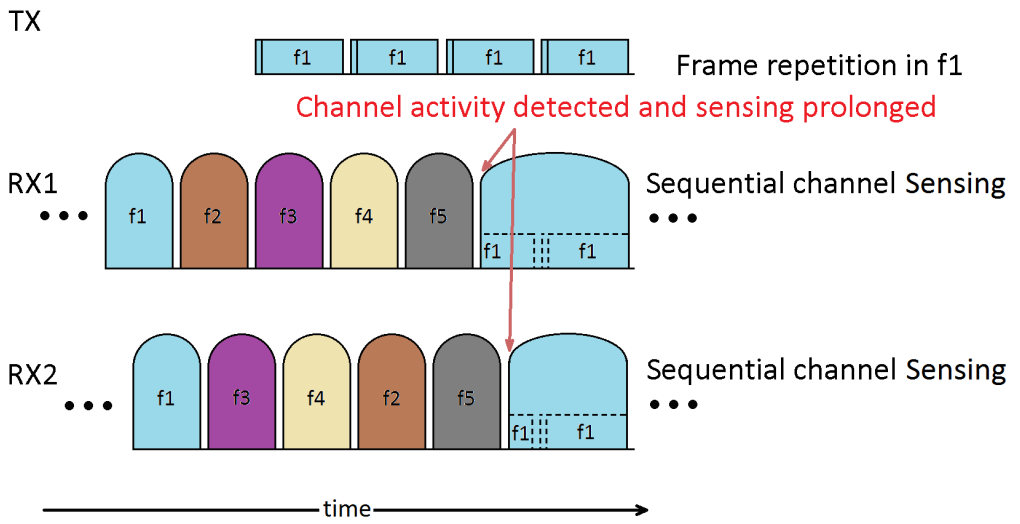


Figure 3.2: Multichannel preamble reservation scheme. As the two (unsynchronized) receivers detect an ongoing transmission, they stay in the channel f1 until a complete frame is received.

continuously sense all available channels to detect potential transmissions. We call the set of available channels as a channel pool of a node. Therefore the operation of sequentially sensing these channels is termed as a pool sensing. Correspondingly, the transmitting node is supposed to send multiple copies of the same frame back-to-back for a long enough duration in order to reserve the channel so that other nodes are able to detect its transmission during the pool sensing. The operation of transmitting the same frame repetitively is termed as a repetitive frame transmission. As shown in Figure 3.2, the transmitting node (TX) carries out repetitive frame transmission in channel f1 while two idle nodes (RX1 and RX2) are carrying out the pool sensing, i.e., they sequentially sense channels from f1 to f4. As shown in the figure, both RX1 and RX2 detect the transmission and successfully receive at least one complete frame. In MPR scheme, the length of repetitive frame transmission should be long enough to guarantee the reception of the frame by all nodes, which may also have different pool sizes. Since the repetitive frame transmission of the first frame implicitly synchronizes nodes, the frame repetition for the following frames is not required. In order to efficiently support high data traffic loads and minimize the number of frame repetitions, CogMAC+ continuously transmits frames without repetition after reserving the channel, namely aggregation transmission. It is similar to the MPDU aggregation in IEEE 802.11ac standard [45]. The number of frames in repetitive frame transmission and aggregation transmission are marked with $N_{\text{Re_Fr}}$ and $N_{\text{Mul_Fr}}$ in Figure 3.4, respectively. In order to avoid interrupting external transmissions, the transmitting node has to regularly suspend frame transmission to sense the channel. In this work, we assume that nodes do not have any signal classification or feature detection functionalities [107] to distinguish different transmission technologies. However, if such a capability is supported on the platform, it can be added straightforwardly to further improve the PU/SU detection.

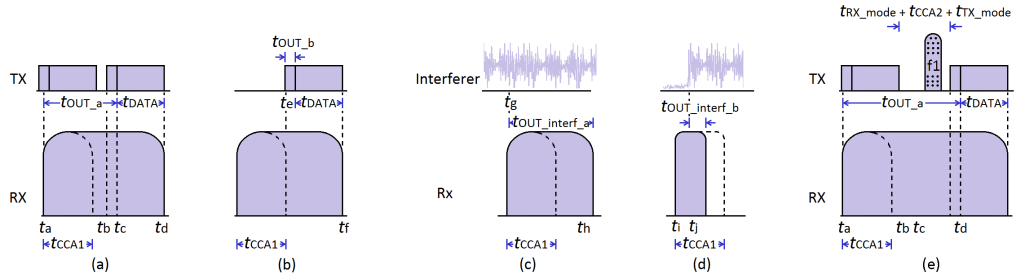


Figure 3.3: Different TIMEOUT durations in CogMAC+: (a) start sensing in the middle of a frame (b) start receiving during default CCA operations (c) start sensing when channel is interfered or PU transmission is carrying out (d) interference or PU transmission start during default CCA operations (e) the maximal timeout duration.

In the following we assume that we uses an energy detector based sensing. Energy detection does not require any apriori knowledge of the PU signal [103] which makes CogMAC+ to be capable of working with co-existed heterogeneous transmission technologies. If the received signal energy is lower than the threshold, channel is detected to be idle. Otherwise, the channel is identified to be busy and the node starts a timeout timer and tries to decode the received signal. Detailed information of the carrier sensing and energy detection of CogMAC+ is shown in Subsection 3.2.2 and Subsection 3.2.4, respectively. As shown in Figure 3.4, two types of carrier sensing operations are used in CogMAC+, namely CCA1 and CCA2 with duration t_{CCA1} and t_{CCA2}, respectively. CCA1 is used in pool sensing to sense an individual channel, and CCA2 is a shorter sensing operation used between frame transmissions in order to detect external transmission in the same channel. If there is no data to transmit or receive, a node repetitively carry on pool sensing. Before starting a transmission, a node is required to perform pool sensing, and then sense the chosen channel again to ensure the availability of the channel.

MPR scheme is effective in distributed multichannel MAC protocols. Therefore, CogMAC adopts it to achieve the seamless transmission in multichannel environment and CogMAC+ also follows its principle. In CogMAC+, we re-design the MPR scheme and rename it as MMT scheme by re-analyzing the timing parameters of the protocol. In MMT scheme, we derive several bounds to optimize the protocol performance and achieve better protection to PU transmissions. Moreover, we use a frame-based analysis method to instead the previous time-based analysis method of CogMAC, which is more straightforward and achieves better portability among different hardware platforms. We show the details of MMT scheme in the next subsection.

3.2.2 Multichannel Multiframe Transmission

A well designed multichannel cognitive MAC protocol should ensure seamless communications among SUs while keeping the interference caused to PU under a

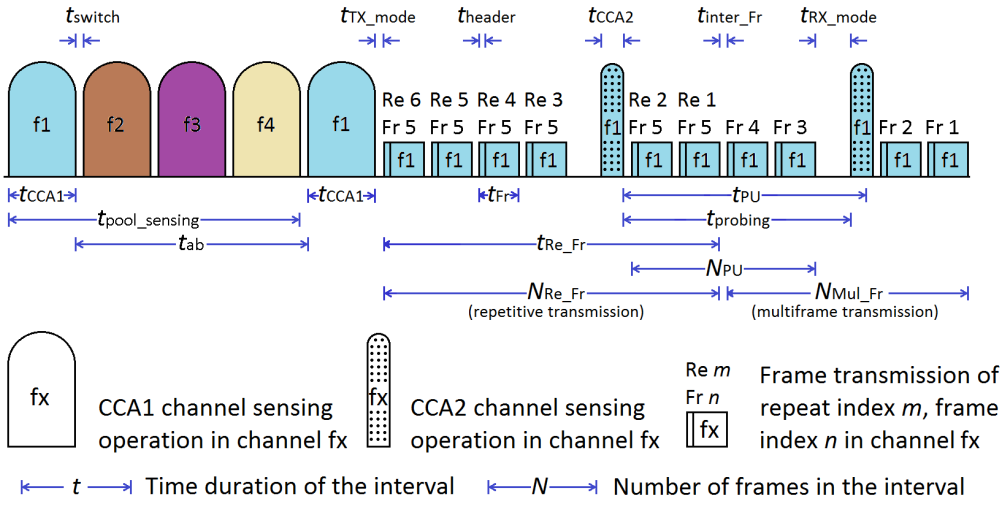


Figure 3.4: Different timing parameters.

permitted level. In CogMAC+, we use MMT scheme, the upgraded version of MPR scheme to meet both demands and optimize the performance. MMT scheme fellows the MPR scheme principle, a transmitting node should repetitively transmit the first frame long enough so that even in the worst case the receiving node can receive at least one copy of the frame while carrying on pool sensing operation. The duration of a node not sensing a specific channel, t_{ab}, depends on the total number of channels. If the channel pool size is P, duration of CCA1 is t_{CCA1}, and the interval between two consecutive carrier sensing operations is t_{switch} as shown in Figure 3.4. Note that t_{ab} is equal to $(P-1)(t_{\text{CCA1}}+t_{\text{switch}})+t_{\text{switch}}$. It is worth noting that the above duration only holds in an ideal case, i.e., when no other transmission takes place. In the latter part of this subsection, we will analyze the scenario where external transmissions exist. Let N_{PU} denotes the upper bound of the number of frames that is allowed to be transmitted in a PU allowance interval, which represents the longest possible duration for SUs to occupy a channel without needing to sense the channel. Considering the minimal duration between two successive frames $t_{\text{inter_Fr}}$, we have,

$$N_{\text{PU}} = \lfloor \frac{t_{\text{PU}} - t_{\text{RX_mode}} - t_{\text{TX_mode}} + t_{\text{inter_Fr}}}{t_{\text{Fr}} + t_{\text{inter_Fr}}} \rfloor, \tag{3.1}$$

where N_{PU} denotes the upper bound of the number of frames that are allowed to be transmitted in one PU allowance interval. Thus t_{probing}, the actual duration between two consecutive CCA2 operations or between the last CCA1 operation and the first CCA2 operation is given by,

$$t_{\text{probing}} = N_{\text{PU}} t_{\text{Fr}} + (N_{\text{PU}} - 1) t_{\text{Inter_Fr}} + t_{\text{TX_mode}} + t_{\text{RX_mode}}. \tag{3.2}$$

Obviously, t_{probing} is upper bounded by t_{PU}. The length of repetitive frame transmission should be lower bounded to guarantee the frame reception at all nodes. Let $N_{\text{Re_Fr}}$ be the lower bound of the number of frames in repetitive frame transmission, the duration of transmitting $N_{\text{Re_Fr}} - 1$ frames should larger than or

equal to t_{ab} so that the reception of the last copy of the first frame can be guaranteed. Since it is not trivial to calculate $N_{\text{Re_Fr}}-1$, we divide $N_{\text{Re_Fr}}-1$ by N_{PU} to calculate how many PU allowance intervals are needed for repetitive frame transmission,

$$\frac{N_{\text{Re_Fr}}-1}{N_{\text{PU}}} = g_{\text{pu}} \mod n_{\text{re}}, \tag{3.3}$$

where g_{pu} denotes the number of complete PU allowance intervals that is needed for repetitive frame transmission, and n_{re} denotes the number of remaining frames required to be sent in the last PU allowance interval. We define the transmission duration function $T(k)$ as a function of the number of frames k. For transmitting $N_{\text{Re_Fr}}-1$ frames, we have

$$T(N_{\text{Re_Fr}}-1) = g_{\text{pu}}(t_{\text{probing}} + t_{\text{CCA2}}) + n_{\text{re}}t_{\text{Fr}} + (n_{\text{re}}-1)t_{\text{inter_Fr}}. \tag{3.4}$$

As mentioned above, the time of transmitting $N_{\text{Re_Fr}}-1$ frames should be larger than t_{ab},

$$T(N_{\text{Re_Fr}}-1) > (P-1)(t_{\text{CCA1}} + t_{\text{switch}}) + t_{\text{switch}}. \tag{3.5}$$

Therefore, g_{pu} and n_{re} can be deduced from (3.4) and (3.5) as,

$$g_{\text{pu}} = \lfloor \frac{(P-1)(t_{\text{CCA1}} + t_{\text{switch}}) + t_{\text{switch}}}{t_{\text{probing}} + t_{\text{CCA2}}} \rfloor, \tag{3.6}$$

and

$$\begin{aligned} n_{\text{re}} = & \lceil \frac{(P-1)(t_{\text{CCA1}} + t_{\text{switch}}) + t_{\text{switch}}}{t_{\text{Fr}} + t_{\text{inter_Fr}}} \\ & - \frac{n(t_{\text{PU}} - t_{\text{CCA2}}) - t_{\text{inter_Fr}}}{t_{\text{Fr}} + t_{\text{inter_Fr}}} \rceil. \end{aligned} \tag{3.7}$$

Since we use the ceiling function in (3.7), n_{re} can be larger than N_{PU}. It is worth noting that n_{re} denotes the remaining frames in the last PU allowance interval, therefore it must be less than or equal to N_{PU}. If n_{re} is larger than N_{PU}, i.e., the receiving node starts sensing again in either $t_{\text{TX_mode}}$, $t_{\text{RX_mode}}$ or t_{CCA2} durations, n_{re} is enforced to be N_{PU}. Thus we get the number of repetitive frames,

$$N_{\text{Re_Fr}} = g_{\text{pu}}N_{\text{PU}} + \min(n_{\text{re}}, N_{\text{PU}}) + 1. \tag{3.8}$$

To avoid missing repetitive frame transmission, the duration of CCA1 operation has to be lower bounded as below,

$$t_{\text{CCA1}} > t_{\text{TX_mode}} + t_{\text{RX_mode}} + t_{\text{CCA2}}. \tag{3.9}$$

An example is shown in Figure 3.5 to illustrate the worst case. If the CCA1 duration is less than $t_{\text{TX_mode}} + t_{\text{RX_mode}} + t_{\text{CCA2}}$, the sensing period of Node 2 may fall in the no transmission interval around CCA2 duration of Node 1. In this instance, Node 2 can miss the repetitive frame transmission from Node 1.

The existence of interference or parallel transmission can also lead to the prolongation of carrier sensing operations in pool sensing. Figure 3.3(a) and

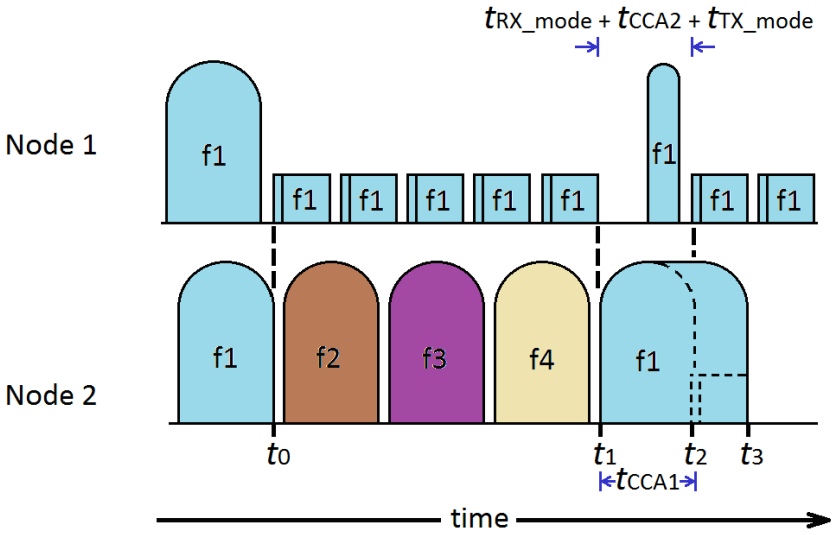


Figure 3.5: Relation between two types of carrier sensing durations in CogMAC+.

Figure 3.3(b) show two scenarios in which repetitive frames are transmitted over the air. Carrier sensing operations are prolonged when a signal is detected. In Figure 3.3(a), RX starts sensing at t_a while TX is transmitting, i.e., RX detects signal at the beginning of its sensing period. Due to repetitive frame transmission, RX should prolong its sensing until t_c to receive the frame header of the next frame. After receiving the frame header, RX should stay in the channel to receive the following data. In Figure 3.3(b), RX receives the frame in its regular sensing duration, thus it can receive the complete frame header. In this instance, the sensing duration is equal to the transmission time of the frame header after detecting the signal. Please note that for the sake of simplicity, we do not take the processing time which is inevitable in realistic wireless platforms into account. Figure 3.3(c) and Figure 3.3(d) indicate the scenarios of sensing an external transmission. Same as Figure 3.3(a) and Figure 3.3(b), the carrier sensing durations in Figure 3.3(c) and Figure 3.3(d) should be set to $\max(t_{\text{OUT_a}})$ and $\max(t_{\text{OUT_b}})$, respectively. If the frame size is known by RX, then $\max(t_{\text{OUT_a}})$ is equal to $t_{\text{Fr}} + t_{\text{inter_Fr}} + t_{\text{header}}$ and $\max(t_{\text{OUT_b}})$ is equal to t_{header}. Considering the CCA2 operation, the longest sensing duration is equal to $t_{\text{Fr}}+t_{\text{header}}+t_{\text{TX_mode}}+t_{\text{RX_mode}}+t_{\text{CCA2}}$ as shown in Figure 3.3(e). Therefore (3.5) has to be rewritten as (3.10) if an external transmission exists.

Due to the prolongation of the sensing operation, the transmitting node has to transmit more frames repetitively in order to guarantee the seamless transmission. Nodes can determine the existence of external transmission according to the false positive frame detection ratio. If the ratio is significantly higher than the default ratio, which is caused by the adaptive threshold adjustment for energy detection (cf. Subsection 3.2.4), the transmitting node should use (3.10) instead of (3.5) to deduce

the number of repetitions.

$$T(N_{\text{Re_Fr}} - 1) > (P-1)(t_{\text{Fr}} + t_{\text{header}} + t_{\text{TX_mode}} + t_{\text{RX_mode}} + t_{\text{CCA2}} + t_{\text{switch}}) + t_{\text{switch}}. \quad (3.10)$$

CogMAC+ enables parallel transmission if nodes have independent unicast/multicast traffic (cf. Subsection 3.2.3). In the instance, the prolonged sensing duration is related to the number of parallel transmission and the channel pool size. If a node detected j transmitters in its detection range, then it may detect parallel transmission in $\min(j, P-1)$ channels. Therefore, (3.5) should be replaced by (3.11) as,

$$T(N_{\text{Re_Fr}} - 1) > (P - 1 - \max(0, P-1-j))(t_{\text{CCA1}} + t_{\text{switch}}) + t_{\text{switch}} + \max(0, P-1-j)(\max(t_{\text{Fr}} + t_{\text{inter_Fr}} + t_{\text{header}}, t_{\text{CCA1}})). \quad (3.11)$$

In the next subsection, we show how to carry out parallel transmissions by using the timing information specified by MMT scheme.

3.2.3 Multichannel Virtual Carrier Sensing

In CogMAC, all nodes in the range of an ongoing transmission have to wait in the channel until the ongoing transmission ends. This broadcast-like scheme effectively propagates the channel selection information and implicitly synchronizes all nodes in the neighborhood for broadcast communication pattern. However, for unicast or multicast scenarios, non-addressed nodes unnecessarily wait in a channel for receiving unwanted frames. By analyzing the sequence number of the received frame and using MMT schemes reversely, CogMAC+ nodes are capable of estimating the duration of ongoing transmissions accurately. CogMAC+ provides MVCS scheme to reduce the unnecessary waiting time and allows parallel transmissions in different channels in order to enhance the cumulative throughput for independent unicast and/or multicast transmissions. MVCS scheme allows a node to use meta-data from the received frame header to decide whether to stay in the current channel for receiving the following frames or to temporarily remove the current channel from its channel pool until the end of the ongoing transmission. The blocking time is exactly equal to the duration of the ongoing transmission. In other words, if a node is the destination of the ongoing transmission, obviously it should stay in the current channel for receiving the remaining frames. Otherwise, it blocks the current channel and then continues pool sensing. Temporarily removing a channel which is including an ongoing transmission reduces the channel pool size, and avoids timeout for the channel in the following pool sensing operations. Furthermore, after blocking the currently used channel, a node is still able to sense the remaining channels in its channel pool and carry out parallel unicast/multicast transmissions to nodes which are not transmitting or receiving. A timer is set after blocking a channel, which is called Channel Allocation Vector (CAV) of the channel. CAV is set according to the estimated duration of the ongoing transmission. When CAV expires, the

node reallocates the corresponding channel into its channel pool and senses it in the following pool sensing operations.

The duration of CAV is estimated by using the frame indices of the first received frame header of the ongoing transmission. In CogMAC+ frames are assigned two indices, the index of the frame in repetitive frame transmission and the index of frame in aggregation transmission. In order to accurately figure out the position of the received frame in the transmission sequence, we assign the two indices in a reversed order to frames when they are generated, i.e., the transmitting node generates and transmits frames from the highest index to 0 as shown in Figure 3.6. An example of three node transmission is illustrated in Figure 3.6. At t_{1_tx}, Node 1 starts transmitting to Node 2 in channel f1. Node 2 and Node 3 start to sense channel f1 at t_{2_s1} and t_{3_s}, respectively. In this example, Node 2 and Node 3 receive one header of the repetitively transmitted frame at t_{t2_rx} and t_{t3_CAV}, respectively. After receiving the header which includes the destination information, Node 2 stays in the channel and waits until all frames from Node 1 are received at $t_{2_rx_end}$. However, after extracting destination information at t_{t3_CAV}, Node 3 determines that it is not the destination of the ongoing transmission. Therefore, after calculating the CAV duration $t_{3_f1_CAV}$, Node 3 temporarily removes channel f1 from its channel pool. After the reduced pool sensing and an extra CCA1 operation in channel f2, Node 3 carries out its repetitive frame transmission that is targeted to another node in the network in channel f2. At this moment, the network carries out two parallel transmissions. After $t_{1_tx_end}$, which equals $t_{2_rx_end}$, Node 1 and Node 2 start new pool sensing operations at the same time. Having a different order of channels in pool sensing, Node 1 and Node 2 detect the ongoing transmission from Node 3 at different time instances and successfully receive the frame header at t_{1_CAV} and t_{2_CAV}, respectively. By reading the destination information in the header, Node 1 and Node 2 temporarily remove channel f2 from their channel pools so that the following pool sensing operations are carried out only for three channels. In general, the CAV duration t_{CAV} can be calculated from the repeat index m and frame index n as,

$$t_{\mathrm{CAV}} = T(N_{\mathrm{Re_Fr}} + N_{\mathrm{Fr}}) - t_{\mathrm{header}} - T(N_{\mathrm{Re_Fr}} - m + N_{\mathrm{Fr}} - n + 1), \tag{3.12}$$

where N_{Fr} denotes the total number of frame transmissions in both repetitive frame transmission and aggregation transmission as shown in Figure 3.4, which is not equal to the number of frames in aggregation transmission.

Moreover, if a node is transmitting or receiving in a particular channel, it is not able to receive data in other channels during its CAV duration. Therefore CAV mechanism not only blocks the occupied channel for the estimated CAV duration, but also blocks the nodes that are either the transmitting node or the receiving node of the ongoing transmission for the same duration. Therefore, a node should not only check the destination address of the ongoing transmission, but also record both the destination address and the source address from the received frame header. Therefore, a blocking list is maintained to record the nodes which are temporarily blocked by MVCS scheme. A node should not transmit to the nodes in its blocking list.

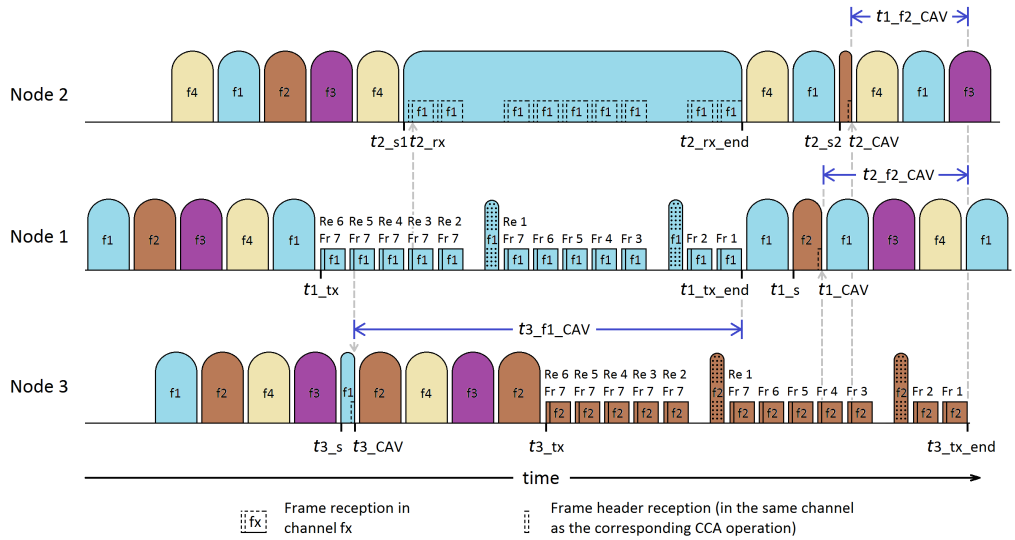


Figure 3.6: Channel allocation vector in multichannel virtual carrier sensing scheme for parallel transmissions. Node 2 and Node 3 detect an ongoing transmission from Node 1 to the specified destination Node 2 in channel f1 at t_{2_s1} and t_{3_s}, respectively. Following the multichannel preamble reservation scheme, Node 2 stays in the channel f1 to receive the remaining frames. However, Node 3 temporarily removes channel f1 from its channel pool for the interval $t_{3_f1_CAV}$ and carries out its own transmission in channel f2 to another node.

Due to the prolongation of the timeout duration as mentioned in the last subsection, external transmissions can also cause prolongation of the carrier sensing duration. The prolonged carrier sensing leads to correspondingly longer repetitive frame transmission, thus decreasing the performance of the protocol. In order to avoid wasting time for sensing the occupied channel, MVCS scheme temporarily blocks the channel if no valid frame header is received during the prolonged timeout duration. Due to the unknown external transmission pattern, an exponential backoff scheme is applied to the channel as an exponentially increased multiple of CCA1 durations as shown in Algorihm 3. Unlike IEEE 802.11 protocol, which aims at differentiating the retransmission time to avoid collision, CogMAC+ keeps the same channel pool to address the rendezvous problem. Therefore the exponential backoff scheme is designed without randomness.

3.2.4 Noise Floor Estimation and Adaptive Energy Detection Threshold

As mentioned in [67, 103], due to the simplicity and low hardware requirement, energy detection is widely used. For example, a simple noise floor estimation and clear channel assessment algorithm is provided in [54] for embedded wireless sensor nodes, where carrier sensing is performed by comparing the received power level with a predefined threshold. In general, low detection threshold leads to a large

Algorithm 3 multichannel virtual carrier sensing

```
while In reception mode do
    for channel c ∈ C do
        if Channel c is not expelled then
            Sensing channel c
            if BUSY then
                if Received a frame header then
                    if destination address matches then
                        Stay in the channel
                        Receive the whole frame
                    else
                        Leave the channel, P \ c
                        t_c_CAV ← header information
                    end if
                else
                    Leave the channel, P \ c
                    Backoff coefficient u_c ← u_c + 1
                    v_c ← 2^{u_c} − 1
                end if
            end if
        else
            if v_c > 0 then
                v_c ← v_c − 1
            end if
            if t_c_CAV = 0 AND v_c = 0 then
                P ← P ∩ c
            end if
        end if
    end for
end while
```

detection range and enables the device to cooperate with neighboring devices which has low transmission power or long distance. However, the low detection threshold also causes high false positive frame detection ratio. In CogMAC+, the false positive detection leads to prolonging carrier sensing duration, thus asynchronizes nodes and causes unnecessary delays. In order to build a stable testing system with more than ten USRP2 devices, the false positive frame detection ratio should be strictly controlled. On the other hand, high detection threshold also has drawbacks, e.g., it lowers down the communication range which is also important for carrying out large scale experiments. Therefore, a appropriate detection threshold should be applied on every devices. Nevertheless, the differences of hardware characteristics and fidelities leads to the differences in measuring received signal strength, noise level, and noise distribution in a real deployment, even if the same type of hardware are in use. In addition, the characteristic of noise may vary with different positions, and even changes over the course of operations. The uncertainties mentioned above makes

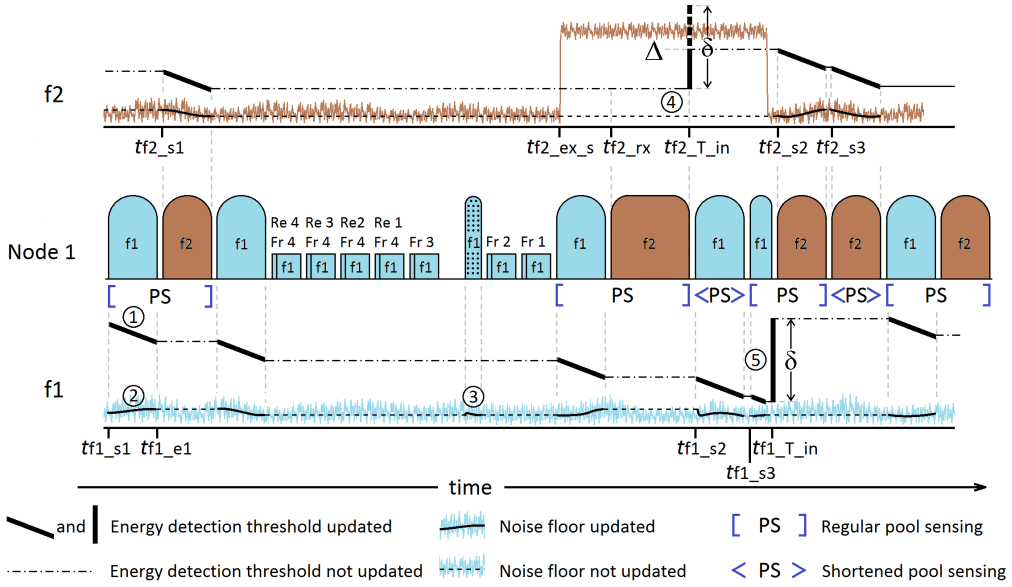


Figure 3.7: An example of noise floor estimation and adaptive energy detection threshold adjustment. Node 1 are the only node in the scenario which has two channels, f1 and f2 in its channel pool. The node starts a frame transmission in channel f1 after the first round pool sensing operation. An external transmission starts from $t_{\text{f2_ex_s}}$ in channel f2.

carrier sensing scheme with fixed detection threshold ineffective and unportable. The benefit of changing energy detection thresholds has been investigated and the concept of adaptive detection threshold methods have been proposed in [3, 63]. In order to achieve a stable experimental environment and improving portability, we propose and implement Algorithm 4 which aims at empirically balancing the false positive detection ratio and the detection range, while keeping the protocol stable at run-time. It requires no a priori knowledge of the pattern of noise and external transmission and allows nodes to dynamically adjust the threshold based on the parameters and states from MAC layer. The algorithm includes two major objectives, real-time Noise Floor Estimation (NFE) and Adaptive Energy Detection Threshold Adjustment (AEDTA). NFE updates the noise level during carrier sensing operations if the channel is determined idle. AEDTA is following a greedy strategy that it continuously reduces the threshold to increase its detection range until detects an external transmission or hits the noise signal, which are indistinguishable as we assumed above. The key principle of NFE and AEDTA is that both of them should be performed when the channel is idle. Therefore, MAC layer status, especially the carrier sensing results are used to determine the operations of NFE and AEDTA. We will use an example to describe their behaviors in the rest of this subsection.

As illustrated in Figure 3.7, NFE and AEDTA are only carried out when the channel is idle. For instance, Node 1 senses channel f1 from $t_{\text{f1_s1}}$ to $t_{\text{f1_e1}}$, where the bold and straight line marked by ① represents the threshold adjustment, and the curve marked by ② is the estimated noise floor which updated by an EWMA with

coefficient α. One specially situation is marked by ③, where only the noise floor is estimated. It is due to the fact that decreasing threshold may increase the risk of hitting the noise level as shown between $t_{\text{f1_s3}}$ to $t_{\text{f1_T_in}}$. Therefore, AEDTA is not performed during CCA2 operations in order to preserve transmissions.

Algorithm 4 adaptive multichannel noise floor estimation and adaptive energy detection threshold adjustment

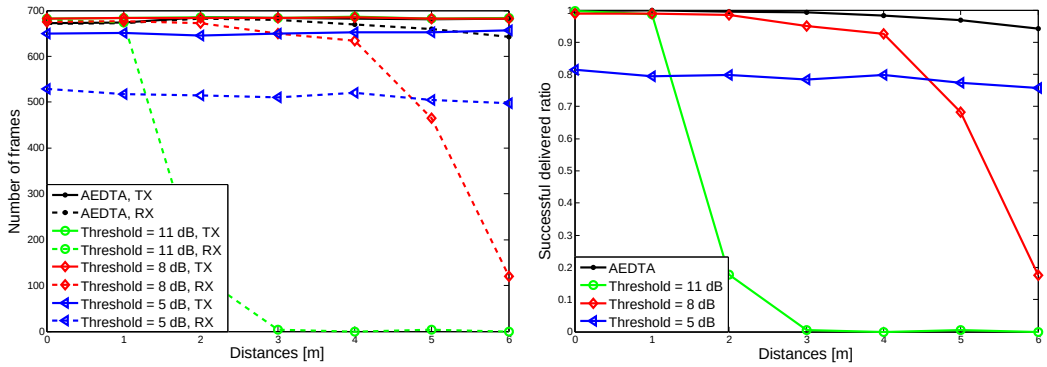
```
Initialize all n_c, ∀c ∈ C
Initialize all T_c, ∀c ∈ C
while CCA1 & CCA2 in channel c do
    if TIMEOUT timer is not activated then
        R_c ← Σ_{n=1,...,N_sa} r_n/N_sa
        if Σ_{n=1,...,N_de_sa} r_n/N_de_sa < T_c then
            if Σ_{n=1,...,N_de_sa} r_n/N_de_sa < T_noise then
                n_c ← αn_c + (1 − α)R_c
            end if
            S_c ← IDLE
            if CCA1 in channel c then
                T_c ← T_c − q^Ne-
            end if
        else
            S_c ← BUSY
            Start timeout timer
        end if
        if timeout ends, no header received then
            if CCA1 in channel c then
                T_c ← T_c + q^CCA+
            end if
        end if
        if header received during timeout then
            if Receives the corresponding frame then
                T_c ← T_c − q^CCA-
            else
                T_c ← T_c + q^CCA+
            end if
        end if
    end if
end while
```

Symbols ④ and ⑤ marks the two instances where the energy detection threshold is increased. In channel f1, Node 1 carries out a CCA1 operation at $t_{\text{f1_s3}}$ with a relatively low threshold. After a short timeout duration as described in Figure 3.3, the energy detection threshold is increased δ in order to be away from noise level. The similar operation appears in channel f2 which starts from $t_{\text{f2_rx}}$, when Node 1

begins to sense the channel. As shown in the figure, an external transmission is being carried out since $t_{\text{f2_ex_s}}$. Node 1 detects it at the beginning of the following CCA1 operation. After the timeout duration without receiving a decodable frame header, Node 1 increases the threshold of channel f2 by δ. However, the threshold should not be beyond a upper bound Δ, which represents the estimation of the lowest energy level of external transmission. Therefore the increased threshold is set to Δ at $t_{\text{f2_T_in}}$.

In our LabVIEW/USRP2 testbed as shown in Figure 3.9, the receiving functionality of USRP2 devices are uncalibrated, i.e., the noise level are different at different USRP2 devices. Moreover, the numerical readings of the received signal strength may be non-linear to the actually received signal strength. Therefore, it is impossible to set a fixed threshold with a absolute value to all USRP2 devices as the energy detection threshold, especially when multiple uncalibrated USRP2 devices are deployed in one experiment. Accordingly, NFE and AEDTA are essential for stabilizing our USRP2 implementation. In order to show the advantages, we have carried out a straightforward experiment in a typical indoor deployment. Two nodes which run the original CogMAC are supposed to communicate with different distances. The two nodes have the LOS channel and NFE scheme is carried out for all experiments. For each distance, the goodput of original CogMAC with AEDTA and several fixed SINR thresholds are tested and compared. As shown in the figure, AEDTA outperforms other fixed threshold experiments nearly in all distance. During the experiment, we found that the minimal effective threshold with our experimental settings is 4 dB, i.e., the deployed USRP2 devices always report channel busy if the threshold is only 4 dB higher than the average noise level. As shown in Figure 3.8, low threshold curve (threshold is 5 dB) remain constant when the distance between nodes varies. However, their performance suffers due to the high false positive frame detection ratio for all distances. The false positive frame detection ratio affects the performance of the whole protocol from two sides. First, the transmitter may detect channel busy then postpones or cancels the imminent transmission. Second, the receiver can miss the frame transmission by entering the receiving state machine at the wrong time. High threshold performs quite well when two nodes are closed to each other, but they performs very bad in the experiments with long distances, where the received signal strength is low. This experiment shows that the performance of NFE and AEDTA schemes is definitely dominant.

The mechanism of blocking occupied channel, which is mentioned in the end of the last subsection is also shown in Figure 3.7. As described above, timeout timer of Node 1 expired at $t_{\text{f2_T_in}}$ without receiving any valid frame header. Therefore, channel f2 is removed from its channel pool in the next pool sensing operation. The same operation appears for channel f1 at $t_{\text{f1_T_in}}$. In both instances, the busy channel is removed from the channel pool so that the following pool sensing operation is shortened.



(a) (b)

Figure 3.8: One transmitter-receiver pair is used in this experiment for validate NFE and AEDTA. The achieved goodput with respect to different distance between the transmitter and the receiver with AEDTA and fixed energy detection thresholds are compared. The transmitter generating randomized traffic with a fixed frame size of 500 bytes. Fixed CCA2 duration of 30 ms and PU allowance time of 500 ms are used. PU allowance time of is 500 ms.

3.2.5 Channel Weighting and Dynamic Channel Selection

Before transmitting in a channel, CogMAC+ requires the node to carry out pool sensing, and then sense again the chosen channel to ensure the availability of the channel. We design for this a heuristic-based algorithm including channel sorting and channel selection. Channel weights are updated by a time-based EWMA, which keeps the historical information while giving higher importance to recent channel activities. Channel weights are adjusted by the following equation,

$$W_c = W_c e^{-\lambda(t_1 - t_0)} + w_c(t_1), \tag{3.13}$$

where W_c is the current weight of channel c and $w_c(t_1)$ is the weight adjustment of channel c at time t_1. Time point t_1 is after t_0. Algorithm 5 shows all the events and the corresponding weight adjustments. In CogMAC+, pool sensing follows the descending order of channel weights and the transmission is always carried out in the channel with the highest weight. Therefore, channel selection is determined by the channel weights. In order to keep the sensing order stable during pool sensing, channel sorting algorithm is performed only after the key time points such as at the end of pool sensing and at the end of aggregation transmission. A node removes a channel out of its pool if its weight is lower than a predefined threshold. In order to keep the channel pool size constant, the node should include a unused channel into its pool according to the last visited time. Therefore, a node should maintain a local timestamp list of all channels and record the timestamp of a channel when it is removed from the channel pool. The weight of new channel should be initialized to a predefined value.

If all nodes are able to know the channel pool of other nodes, network coordination and synchronization become easier, and the rendezvous problem is efficiently

Algorithm 5 channel weighting

$w_c \leftarrow 0, \forall c \in \boldsymbol{C}$
loop
  $\tau \leftarrow$ current time
  for all $(c \in \boldsymbol{P})$ **do**
    if CarrierSensing(c) **then**
      if $S_c =$ Free **then**
        $W_c \leftarrow W_c * e^{-\lambda(\tau-\tau_c)} + 1$
      else
        $W_c \leftarrow W_c * e^{-\lambda(\tau-\tau_c)} - 7$
      end if
    end if
  end for
  if Start Tx or Rx **then**
    $W_c \leftarrow W_c * e^{-\lambda(\tau-\tau_c)} + 5$
  end if
  if Rx **then**
    if Destination address matches **then**
      for all $j \in$ received channel map **do**
        $W_j \leftarrow W_j * e^{-\lambda(\tau-\tau_j)} + 3$
      end for
    else
      for all $j \in$ received channel map **do**
        $W_j \leftarrow W_j * e^{-\lambda(\tau-\tau_j)} - 3$
      end for
    end if
    $\tau_j \leftarrow \tau$
  end if
  $\tau_c \leftarrow \tau$
end loop

handled. We use a vector named channel map to record the channel pool and insert the vector in the frame header. Therefore, the channel pool of a node is able to be received is known by all receiving nodes within its transmission range. The receiving nodes increase the channel weights which are marked in the received channel map. By exchanging channel pool information, nodes in the same network will soon converge to the same channel pool therefore the wasted time of carrier sensing is reduced, and the rendezvous problem is better addressed.

3.2.6 CCA Based Random Backoff Scheme

The random backoff scheme is used in CogMAC+ for collision avoidance. Before starting transmission, e.g., $t_{1_tx_end}$ of Node 1, and $t_{3_tx_end}$ of Node 3 as shown in Figure 3.6, nodes have to wait a random time to avoid collision. In order to keep



Figure 3.9: A snapshot of our LabVIEW-USRP2 testbed.

synchronization after transmission and reception, CogMAC+ chooses the duration of CCA1 plus the channel switching duration as the unit time of the random backoff scheme. Moreover, choosing this duration as the time unit of backoff detaches the whole backoff scheme from unnecessary hardware limitations, which improves the portability of the protocol. Nodes start to transmit when they have data to transmit while the backoff counter is 0. Nodes freeze the backoff counter during the reception period, but continuously countdown during the CAV duration.

3.3 Protocol Implementation

We have developed an SDR testbed consisting of PCs with LabVIEW 2012 and multiple USRP2 devices as shown in Figure 3.9. In order to verify CogMAC+ in a relatively large scale network, up to 16 USRP2 devices are used in the experiments. The RF front-end of the testbed is implemented by USRP2 devices which provides a full physical layer configurability which enables an integrated MAC protocol design, e.g., NFE and AEDTA of CogMAC+. The software is implemented using LabVIEW, which is inherently concurrent language offering a high degree of flexibility for protocol design. Especially, it enables a parallel design which is able to address the delay problem caused by baseband processing as mentioned in [28, 112]. Moreover, we can use its interactive Graphical User Interface (GUI) to visualize the features of the implemented applications in real-time. In LabVIEW, MAC protocols are implemented as Virtual Instruments (VIs). Therefore, we are able to compare different MAC protocols with the same testbed and same configurations by switching the corresponding VIs. Table 3.2 lists some of the key parameters used both for CogMAC+ and CogMAC. A detailed description of the experimental testbed can be found in [82].

Table 3.2: PHY/MAC parameter values.

Parameter	*Value*
Transmit mode switching duration ($t_{\text{TX_mode}}$)	15 ms
Reception mode switching duration ($t_{\text{RX_mode}}$)	150 ms
Time duration for obtaining a channel sample (t_{cs})	1 ms
Channel switching duration (t_{switch})	15 ms
Inter-frame time duration ($t_{\text{inter_Fr}}$)	20 ms
Header transmission duration (t_{header})	1 ms
Max. frame size used ($L_{\text{Fr_max}}$)	1500 bytes
Max. frame transmission time ($t_{\text{Fr_max}}$)	300 ms
Interferer timeout duration ($t_{\text{int_timeout}}$)	10 ms

3.4 Experimental Result

In order to get ergodic results, we have carried out an experimental campaign for three months on the testbed, in which each set of parameters is at least taken 1000 seconds. In this section, we first show the baseline comparisons between original CogMAC on WARP testbed and the new LabVIEW-USRP2 implementations. Then we present the results on performance evaluation CogMAC+ against CogMAC. For the experiments with external transmission, we implemented a multichannel interferer by also using LabVIEW and USRP2 devices which are able to interfere multiple channels simultaneously to emulate PU transmission. The performance of co-existed CogMAC+ networks which use different modulation schemes are also evaluated.

3.4.1 Baseline Comparison

In order to empirically verify CogMAC on the USRP2-LabVIEW testbed, we compare the performance of CogMAC to the results with previous evaluation of CogMAC done using WARP based testbed [48]. It is worth noting that the supported PHY-layer performance levels are different on USRP2 and WARP devices. The baseline comparisons are aimed at studying the performance trends with respect to MAC parameters.

Figure 3.10(a) and Figure 3.10(b) show the goodput for a single transmitter and receiver pair with different numbers of multiframes and different channel pool sizes. Both CogMAC results from Figure 3.10(a) LabVIEW-USRP2 testbed and Figure 3.10(b) WARP testbed show the same trend and scaling as the number of multiframes increases. It is also evident from the figures that for larger channel pool size, the goodput goes down because of longer pool sensing duration.

Besides the number of multiframes and channel pool size, we also compare on two testbeds the effect of different CCA2 durations. The comparison between

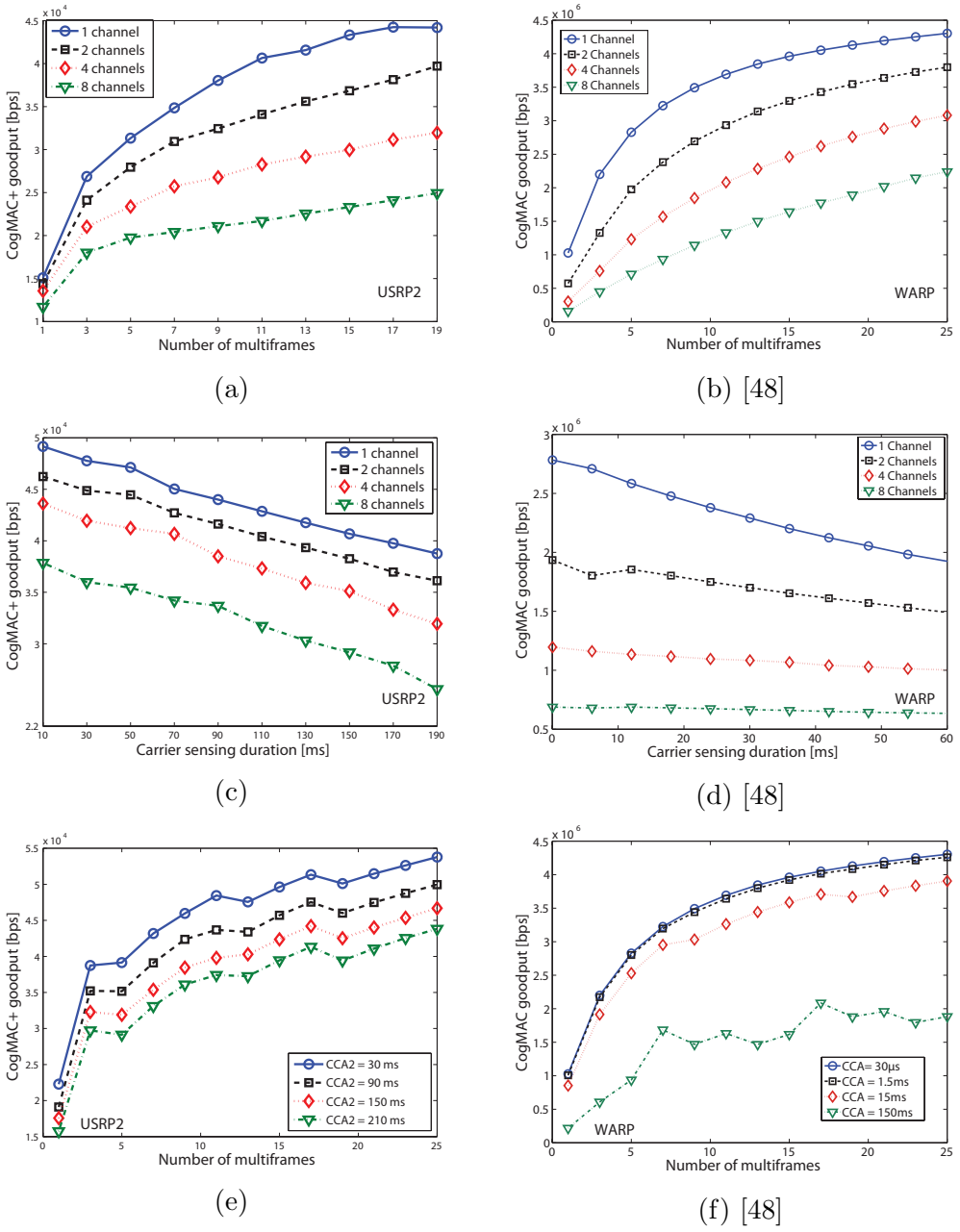


Figure 3.10: Comparison of the achieved goodput with respect to different parameters on USRP2 and WARP testbeds. One transmitter-receiver pair is used in these experiments with the transmitter generating saturated traffic with a fixed frame size of 1000 bytes. Fixed CCA2 duration of 30 ms and 30 μs and PU allowance time of 500 ms and 15 ms are used in (a) USRP2 and (b) WARP testbeds, respectively. PU allowance time of 500 ms and 15 ms are used in (c) USRP2 and (d) WARP testbeds, respectively. PU allowance time of 500 ms and 15 ms are used in (e) USRP2 and (f) WARP testbeds, respectively.

Figure 3.10(c) and Figure 3.10(d) and the comparison between Figure 3.10(e) and Figure 3.10(f) show that the goodput goes down as the carrier sensing duration increases following the same trend as on WARP testbed. In conclusion, larger number of multiframes, smaller carrier sensing duration, and smaller channel pool size lead to higher channel utilization ratio and therefore improve the goodput of CogMAC. In general, we are confident that our USRP2 testbed is implementing earlier [48] protocol properly and that fair comparison is possible against CogMAC+ by using this implementation.

3.4.2 CogMAC+ Enhancement

For both CogMAC+ and CogMAC, a number of parameters such as the number of multiframes and frame size affect the cumulative goodput as previously shown. In this subsection, we empirically characterize CogMAC+ and compare it with CogMAC by varying these key parameters in exactly the same manner. All of the following experiments indicate that for unicast traffic, CogMAC+ substantially outperforms CogMAC in multichannel environment where multiple transmitters and receivers exist.

The number of multiframes determines the length of transmission after reserving the channel, which averages the cost caused by repetitive frame transmission. Figure 3.11 shows the cumulative goodput for two transmitter-receiver pairs scenario and four transceivers scenario, where each transceiver randomly chooses a node as the destination for each transmission. For both scenarios, the transmitters and transceivers generate saturated unicast traffic. We ensure that all nodes are placed within the transmission range of each other. In the experiment, we vary the number of multiframes and the CCA2 duration but fixed channel pool size to four. Due to the parallel transmission by MVCS scheme, CogMAC+ achieves significantly higher cumulative goodput then CogMAC. Both protocols scale similarly with the number of multiframes, but CogMAC+ is always providing higher throughput values.

PU allowance interval affects the performance of SU, because it determines how frequently the transmission is required to be interrupted and sensing is performed. A longer PU allowance interval means that the SU is able to carry out more frame transmissions between two sensing operations, which leads to a higher channel utilization ratio and higher throughput as can be seen from Figure 3.12. In our experiments, CCA2 duration was fixed to 30 ms and channel pool size was fixed to four. Two transmitter-receiver pairs are used in the experiment, where each transmitter generates saturated unicast traffic to a fixed destination. As shown in Figure3.12, both CogMAC+ and CogMAC are able to transmit more data with larger PU allowance interval. For each value of PU allowance interval, the improvement of CogMAC+ compared to CogMAC stays significant.

Similar to the number of multiframes and the PU allowance interval, frame size also affects the channel utilization. Figure 3.13 shows the cumulative goodput of two transmitter-receiver pairs with respect to different frame sizes. As expected, as the frame size increases, the goodput increases.

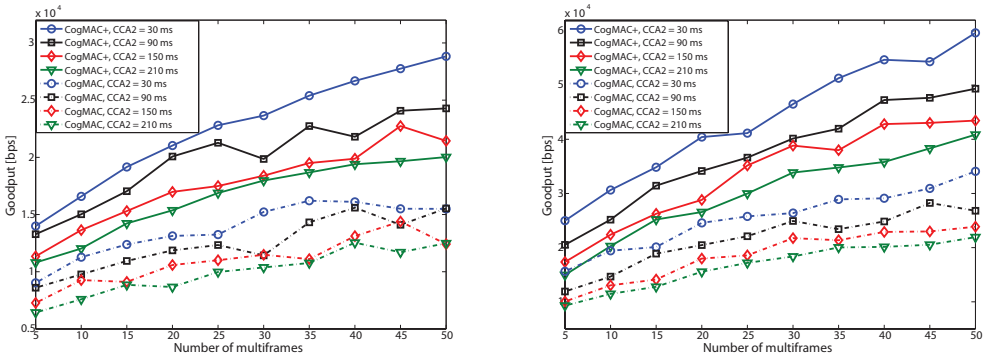


Figure 3.11: Goodput comparison of (a) two transmitter-receiver pairs scenario and (b) four transceivers scenario with respect to different numbers of multiframes and different carrier sensing durations. All transmitting nodes generate saturated unicast traffic with a fixed frame size of 1000 bytes. Fixed CCA2 duration of 30 ms and PU allowance time of 500 ms are considered.

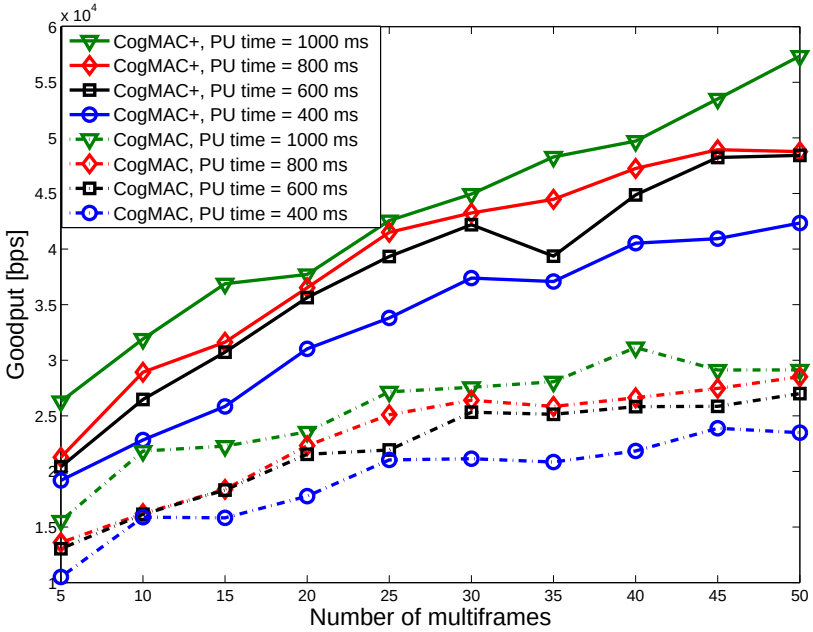


Figure 3.12: Achieved goodput with respect to different numbers of multiframes and different PU allowance time. Two transmitter-receiver pairs is used in the experiment with the transmitters generating saturated unicast traffic with a fixed frame size of 1000 bytes. Fixed CCA2 duration of 30 ms is considered.

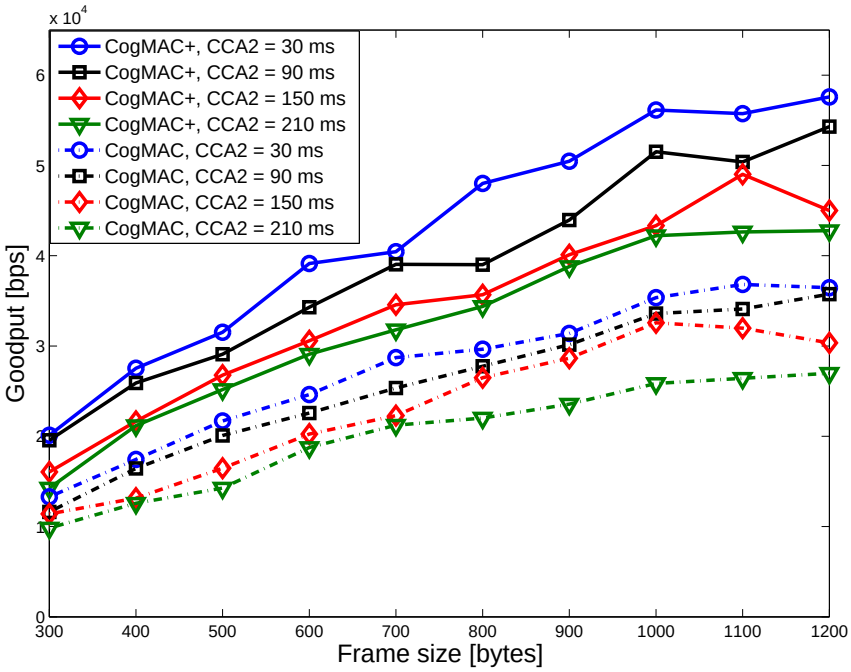


Figure 3.13: Achieved goodput with respect to different frame sizes. Four transceiver scenario is considered in the experiment with the transmitter generating saturated unicast traffic. A fixed PU allowance time of 500 ms and a fixed number of multiframes are considered.

For a network, the number of users affects its performance by causing congestion and collisions. In order to evaluate the performance of CogMAC+ in a larger network, we carry out an experiment with up to 16 USRP2 nodes. The result are shown in Figure 3.14. Number of channels is initialized as four, but increases as the number of transmitter-receiver pairs goes up to ensure that the number of channels is larger than or equal to the number of pairs. Therefore, each pair is able to find a non occupied channel to carry out its own transmission even in the worst case that all other pairs are transmitting at the same time. It can be seen from the result that as the number of pairs increases, the cumulative goodput of CogMAC+ increases due to the fact that more links are active. The figure also shows a trend that as the number of pairs increases, the increment of cumulative goodput of the new pair is reduces, i.e., the cumulative goodput tends to converge when the number of nodes increases. This happens because the contention increases as the number of pairs increases, the pool sensing duration is also prolonged according to the channel pool size and parallel transmissions. A more clear convergence behavior can be seen from the graphs of CogMAC in the same figure. Because CogMAC does not support parallel transmission and virtual carrier sensing, the contention in CogMAC is stronger.

In the above experiment, we always keep the number of channels to be larger than or equal to the number of node pairs. However, wireless spectrum is a very scarce

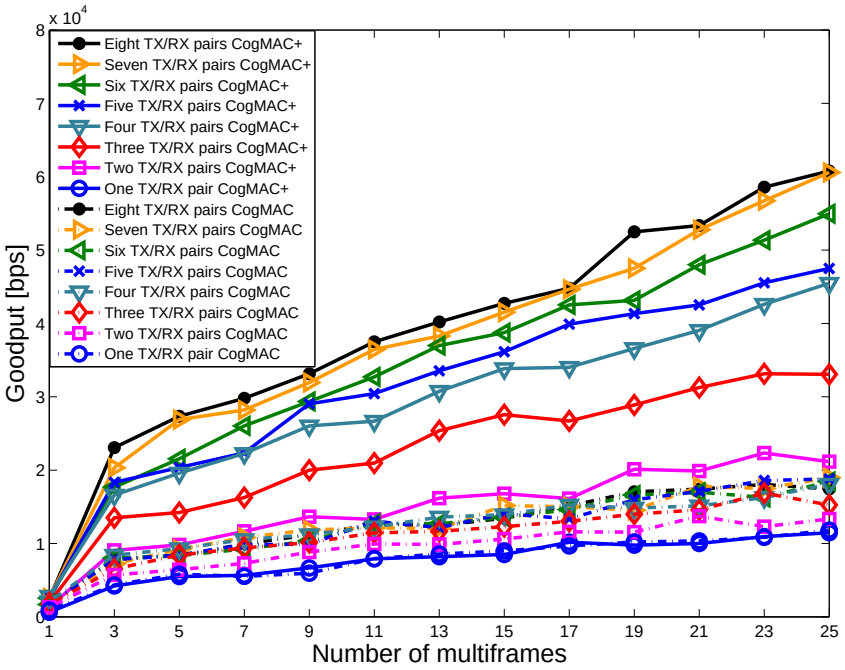


Figure 3.14: Achieved goodput with respect to the number of multiframes. Up to 8 transmitter-receiver pairs are considered in the experiemnt as all of the transmitters generating saturated unicast traffic. A fixed PU allowance time of 500 ms and a fixed CCA2 duration 30 ms are considered. The number of transmitter-receiver pairs is changed from one to eight.

resource in reality. We evaluate the performance of CogMAC+ in a congested network. Figure 3.15 shows the performance of CogMAC+ and CogMAC in such a congested scenario where the number of node pairs vary from one to six and the number of channels is fixed to three. Curves with different numbers of pairs show that the cumulative goodput tend to converge when the number of pairs increases. Due to the limited number of channels, the convergence in this experiment appears earlier compared to the previous one, where more channels are available.

3.4.3 Performance Evaluation of Secondary Usage

In order to evaluate the performance of CogMAC+ with the existence of external transmission, we use a realistic spectrum model to emulate the behavior of external transmission. During the last years an increasing number of studies on spectrum occupancies were carried out. Stabellini made spectrum measurements in 2.4 GHz ISM band while considering heterogenous traffic [61]. Their analysis statistically characterized that the Hyper-Exponential Distribution (HED) model provides a good fit to the empirical data, which assumes that the duration of busy state follows the Exponential Distribution (ED), and the idle state follows HED. The parameters

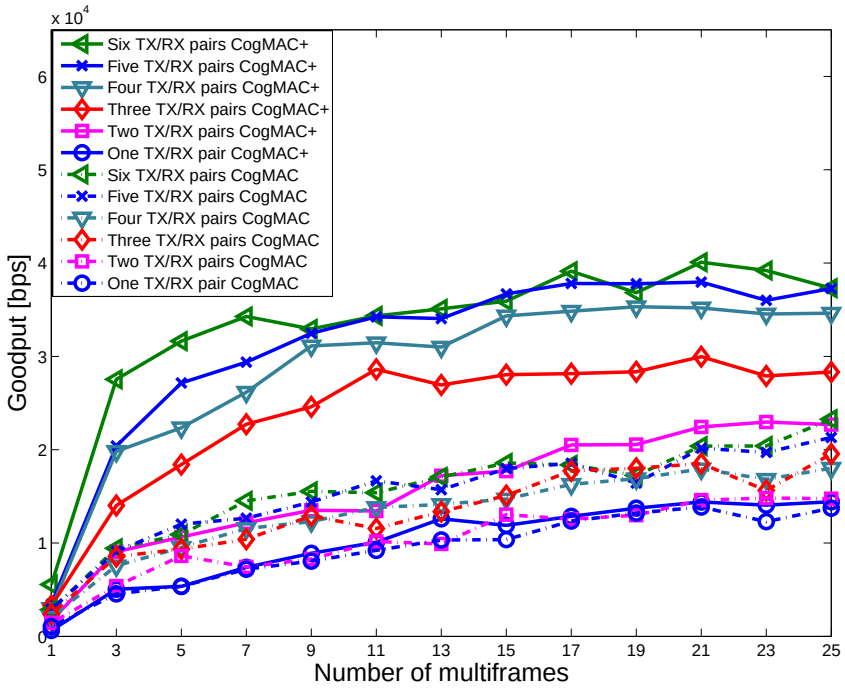


Figure 3.15: Achieved goodput with respect to the number of multiframes. Up to 6 transmitter-receiver pairs are considered in the experiment as all of the transmitters generating saturated unicast traffic. The number of channels is fixed to three in this experiment. A fixed PU allowance time of 500 ms and a fixed CCA2 duration 30 ms are considered. The number of transmitter-receiver pairs is changed from one to six.

of HED is related to the average sequence load which is the ratio of the duration of busy state and the overall experimental time. Due to the speed limitation of USRP2-LabVIEW testbed, we proportionately increase the HED parameters from millisecond to second. We implement an interferer by using the same setup as the testbed which is able to generate different traffic loads according to the HED model to emulate the external transmission.

CogMAC and CogMAC+ are compared by varying the average sequence load in Figure 3.16. The average sequence load is defined in [61] as the ratio between the number of busy samples and the total number of collected samples from a measured sample sequence. In our experiments, we use this term to denote the ratio between the channel idle duration and busy duration. The spectrum model is chosen according to [61], where the three levels of the average sequence load, i.e., the average sequence load of 0.2 for the medium load interval, the average sequence load of 0.4 for the high load interval, and the average sequence load of 0.7 for the very high load interval are tested and the non-interference scenario is also carried out as a reference. It shows that the performance degradation of CogMAC+ in the crowded

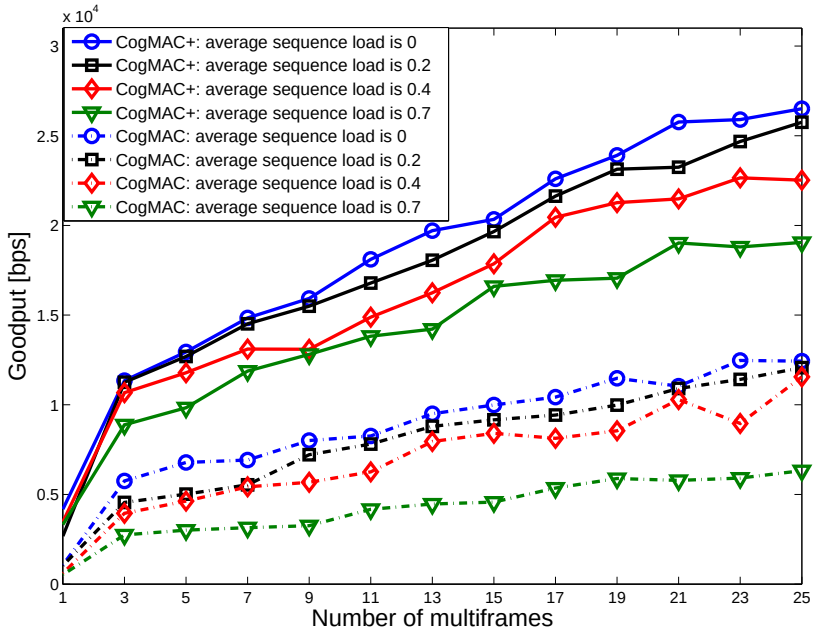


Figure 3.16: Achieved goodput with respect to the number of multiframes. Two transmitter-receiver pairs are considered in the experiment with the transmitter generating saturated unicast traffic. A fixed PU allowance time of 500 ms and a fixed CCA2 duration of 30 ms are considered. Different average sequence loads are changed in the experiment.

environment is lower than the degradation of CogMAC in the same environment.

Since the transmitting node only transmits when the channel is idle, the utilization of channels with different average sequence loads should be different, e.g., the cumulative goodput should be lower in the heavier loaded channel. In order to study the relation between loads and goodput, we design an experiment in which two pairs of nodes carry out transmission in three channels with different loads as shown in Figure 3.17. The result shows a positive correlation between the average sequence loads and cumulative goodput of both CogMAC+ and CogMAC. Moreover, it can be seen from the figure that the channel utilization of CogMAC+ has higher linearity than CogMAC with the loads. It indicates that CogMAC+ is better on exploiting the crowded spectrum environment and it adapts the external transmission in the long-term better than CogMAC.

In order to evaluate the performance of CogMAC+ of colocated heterogeneous cognitive networks, we designed an experiment with two networks which are using different modulation schemes. Therefore, the nodes in one network cannot distinguish the transmission of another network and PU transmission. Four channels are considered in the experiment with different average sequence loads. Following the classification of the average sequence load in [61], two channels are set in the very high loaded interval with the average sequence load of 0.7, one channel falls in the

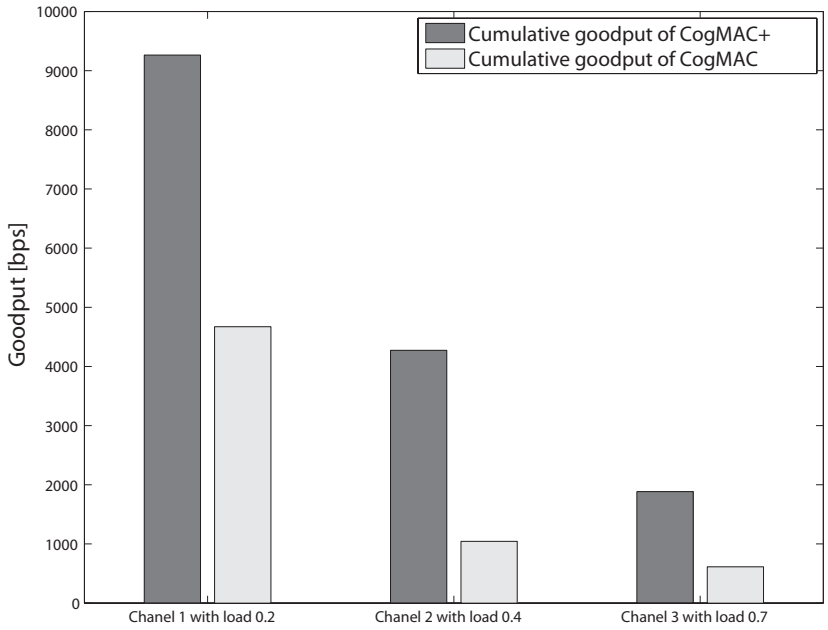


Figure 3.17: Cumulative goodput comparison of CogMAC+ with CogMAC in the presence of external transmission in different channels with different average sequence loads. Two transmitter-receiver pairs are considered in the experiment. Transmitters generating saturated unicast traffic. A fixed PU allowance time of 500 ms and a fixed CCA2 duration of 30 ms are considered.

high load interval with the average sequence load of 0.4, and the last channel is set to the medium load interval with the average sequence load of 0.2. The experimental result shown in Figure 3.18 verifies that both CogMAC+ and CogMAC are able to support colocated heterogeneous networks. The performance of CogMAC+ follow the same trend as in the single network scenario and it is able to achieve substantial improvement in both networks when compared against CogMAC. The results prove that CogMAC+ is able to support the colocated heterogeneous networks and provide reliable and efficient data services for each participating network.

3.5 Summary

We have presented the design and implementation of a novel decentralized cognitive MAC protocol, CogMAC+, which enables nodes to efficiently utilize spectrum holes opportunistically. In CogMAC+, each SU is able to sense all channels in its channel pool and detect external transmissions. It uses repetitive frame transmission for channel reservation, while carrying out aggregation transmission to lower down the frame repetition cost. By using the cyclic sensing scheme, the interruption caused

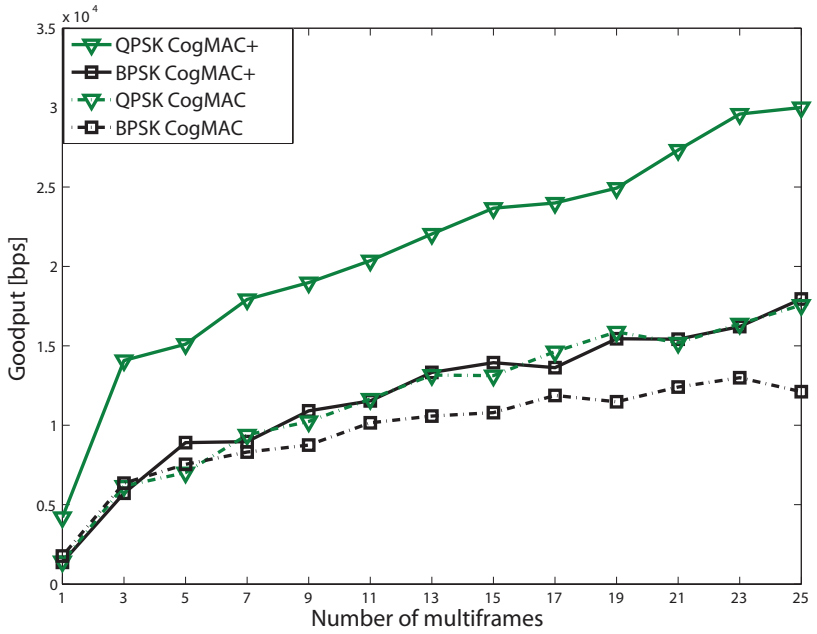


Figure 3.18: Achieved goodput of two networks with respect to different numbers of multiframes. Four transmitter-receiver pairs are considered in the experiment, in which two pairs use BPSK modulation, and another two pairs use Quadrature Phase Shift Keying (QPSK). Frame size is fixed to 500 bytes for both of the networks. All transmitters generate saturated unicast traffic to their preassigned receivers. A fixed CCA2 duration of 30 ms and a PU allowance time of 500 ms are considered.

by SU to PU can be controlled. Moreover, virtual carrier sensing scheme is used to achieve accurate channel sensing and dynamic channel pool control which enables address-specified parallel transmission. It is also able to avoid sensing the occupied channels. We have also integrated the MAC layer states and parameters for adaptive noise floor estimation and clear channel assessment algorithm to allow a relatively low false positive frame detection ratio. We have empirically evaluated CogMAC+ on the USRP2-LabVIEW testbed including up to 16 USRP2 devices under different traffic loads and spectral conditions. A number of experiments revealed the fact that CogMAC+ is able to offer reliable data communication services and significant performance improvement. Our evaluation with realistic spectrum models confirms that CogMAC+ is able to efficiently utilize spectrum holes and enables efficient spectral coexistence.

4 Decomposite MAC Description Language

4.1 Introduction

The technological progresses in integrated circuit design and manufacturing make the communication devices become increasingly powerful while costing less, which enables the deployments of artificial intelligence and environmental cognition in future communication systems. The new emerging features provide an unprecedented capacity and robustness to the next generation of communication systems, however, they also triggered a never-terminating evolution in system design and implementation. To meet the versatile service requirements and achieve utmost utilization of the fast evolving technologies, the development circle of MAC layer schemes has been continuously shortened. The platform-independent methodology also increasingly gain significance in conducting MAC solutions with high universality and compatibility. As a consequence, the modular design principle drawn attention due to its flexibility and extensibility. Moreover, there are greater needs for intelligence and cognitive approaches in wireless network design to handle complex radio environments and heterogeneous scenarios where both cooperation and competition exist. To carry out intricate methods such as artificial intelligence and machine learning, a comprehensive, consistent and systematic mathematical modeling of MAC designs and implementations is needed, which is also essential for developing a platform-independent MAC designing tool.

In this chapter, we focus on analyzing MAC protocols based on Hierarchical Finite State Machine (HFSM) [6] that fits most of the MAC design and implementation requirements. The HFSM model supports co-existence of various concurrency models of computation thus can handle intricate sequential control and address the concurrency which is an essential feature of MAC protocols, in particular the hardware based MAC implementations. In order to be able to fully utilize the characteristics of all these models, we categorized MAC implementations in both software simulators and hardware platforms according to the concurrency models and analyzed the performance of these models in developing MAC protocols while considering the aforementioned challenges. In the light of the analysis, we develop a MAC design tool named Decomposite MAC Description Language (DMDL) which is a new cross-platform architecture that enables flexible and reconfigurable MAC protocol design and implementation, in particular for fast prototyping MAC protocols on SDR platforms. DMDL defines a MAC protocol as a module based messaging driven reactive system. The atomic computational units in DMDL are the fine-decomposed MAC layer primitives, e.g., framing and MAC related functionalities such as Rx and Tx functions at MAC/PHY interfaces. This way, DMDL users can easily build MAC protocols by connecting these computation units

as a timed directed graph. The computational units are the vertices of the directed graph and the controlling signals are passed among MAC components via the directed path. DMDL is aimed at to provide a stable, complete and self-consistent development environment, which can be used directly as a library for designing MAC protocols in different SDR platforms.

This chapter is organized as follows. The state-of-the-art of existing works are shown and discussed in Section 4.2. Section 4.3 introduces the principle of HFSM and shows how to use HFSM to model MAC protocols. Section 4.4 shows the architecture of DMDL and its in-depth design details. To show its ease of use, we demonstrate some examples of using DMDL to build classic MAC protocols in Section 4.5.

4.2 Related Work

In the section, we analyzed several widely used MAC designing tools which share common points with DMDL but also have some fundamental differences. Then we introduce some popular SDR platforms which have the potential to be used for deploying DMDL.

4.2.1 MAC Design Tool

Due to the rapid evolution of MAC protocols, tools that people use to describe and design MACs are also changing over time. In earlier stages, MAC protocols are relatively simple thus easy to explain. For example, the MAC layer scheme of the first generation of ALOHA which people used in Hawaii area can be clear described by several lines of human language without ambiguity. However, the rapidly increasing demands on high speed data communications and reliability force the MAC protocols to be complex and miscellaneous. The last release of the IEEE 802.11 standard [45] is described by a document with more than 1000 pages, which is hard to implement or even understand all the contents straightforwardly. To handle such big projects, new models and methods are needed for both describing and developing.

Traditionally, MAC protocols were hardcoded in special-purpose hardware which did not have any re-programmability and flexibility. Due to the fast evolving communication technology and the significantly increasing demands, the standard maker, the radio equipment manufacturer and the system integrator need a more flexible approach to develop MAC protocols. This way, the development cost can be efficiently reduced and the risk of mistakes can be lowered. Moreover, the modern MAC communication systems are aimed at operating in a more complicated radio environments such as secondary usage (cf. Chapter 3) and directivity (cf. Chapter 2), where a flexible and re-programmable architecture is also needed to achieve run-time realization and optimization.

The general trend of achieving the aforementioned demands is the module principle [32,47,105,110], which first decomposes MAC protocol into independent components.

The complete MAC protocol can be implemented by connecting these already developed MAC modules. The module principle can enhance the flexibility and reconfigurability of the MAC protocol, while also enables fast design. In addition, it is able to further achieve on-the-fly realization and run-time protocol optimization [32, 110] by reuse the existing blocks which are already defined on the supported hardware platform. Furthermore, the module principle intuitively supports MAC protocol visualization thus may further enable to implement MAC graphically.

Following the modular design principle, there are two main branches to divide MAC protocols and define the components. The first approach models a MAC protocol as an FSM and the components are the logic state of the FSM. Actions taking place in state transitions are driven by events. For instance, Wireless MAC Processor (WMP) [105] defines its components as states of the extended FSM such as **IDLE** or **RX**. WMP focuses on providing a framework for the IEEE 802.11 commercial off-the-shelf chipset based development. In contrast, DMDL aims at developing platform independent, technology and protocol agnostic general tool, which is particularly suitable for SDRs. Other forms of FSM also widely used in modeling MACs [62, 78]. The FSM approach has several benefits. First, FSM schedules and executes sequential control logics better than other sequential control model. Second, it not only simplifies the representation of MACs because of its well developed visualization such as statechart [21], but also makes analysis and synthesis easy due to its strict mathematical definitions and finite nature.

The second approach follows the data flow model that first creates a function set including fundamental MAC functionalities by decomposing numerous MAC protocols and abstracting their similarities. After this tedious and complicated work, building new MAC protocols can be simply performed by reconnecting the well defined MAC components together. By well designed ports and connections between the elementary function blocks, the flexibility and reconfigurability of data flow model based systems are considerably higher than traditionally implemented systems. In some earlier works [47, 111], the authors have conducted the decomposition and then have built MAC protocols according to the data flow based model. Moreover, GNU Radio [1] also follows the data flow approach that allows data streaming to flow among its blocks. It is worth noting that its new added message passing mechanism can effectively support asynchronous data transfer among blocks, which has a huge potential on developing MAC protocols. While the data flow only approaches are well suited for many physical layer implementations, they are less ideal for modern MAC implementation work, where different complex branchings and timer related functionalities are regularly present.

4.3 MAC Models

Developing an efficient MAC design tool requires accurate modeling of MAC protocols. However, there is no optimal model fit for all platforms. For example, the models used to simulate MAC protocols in a software environment cannot be used for hardware based implementations. To better understand the pros and cons

among existing models and their targeted environments, we introduce HFSM and use its hierarchical architecture to model several popular MAC simulation softwares and SDRs platforms.

4.3.1 Reactive Systems and HFSM

The reactive system is the basic classification of all the embedded systems, real-time systems and messaging driven software system [74]. It performs interaction with the surrounding environment and neighboring agents through a message passing mechanism rather than simply generate or produce outputs, which matches the definition and demands of modern MAC protocols. Therefore, implementations of MAC protocols, including the software simulators and hardware based prototypes can be considered as reactive systems. Finite State Machine (FSM) has long been used for describing and analyzing the behaviors of reactive systems due to its efficiency and high performance in controlling sequence logic and reacting towards environmental changes which are the key performance indicators of modeling MAC protocols. However, drawbacks, also exist using simple FSM to model MAC protocols. The conventional definition of FSM does not satisfy the demands on the concurrency of a MAC protocol. It is also a lack of ability to perform large scale numerical calculations such as the baseband signal processing or coding.

To overcome the aforementioned drawback, combining FSM with other concurrency models is a promising approach. Several concurrency models have been already used for simulating and implementing MAC protocols. Discrete Event (DE) is a concurrency model which synchronizes its events loosely according to the their dependencies. The synchronization is done by scheduling events by assigning time stamps based on global time of the system. Events are scheduled to execute according to their time stamps chronologically. Definitions and examples of DE are given in [24, 109], and many MAC studies even some very popular tools and development environment such as ns-3 [76] are in this category. Synchronous Data Flow (SDF) is also widely used for modeling MAC protocols. SDF is a special case of the data flow model, which is a subclass of the process network model introduced several decades ago [57]. SDF follows the process networks model principle which is a network of current processing blocks that communicate through passing tokens via unbounded first-in-first-out queues. The model can be simplified to directed graphs where tokens are passed by the vertices via the directed arcs. The processing blocks are triggered by arrivals of the token, named *firing*, and then process the arriving token or other computation. All mentioned models have strength, but also disadvantages. The DE relies on the availability of the global time on all parts of the implementation which makes it hard to handle dynamic concurrency design. The SDF is normally implemented as the atomic blocks of computation thus hard to handle control logic with accurate timing.

Jointly using all models to describe MAC protocol can utilize the strengths of all models while avoid their drawbacks. The hybrid exploitation of different concurrency models has been studied for decades. In [6], the FSM semantics are decoupled

from the semantics of concurrency models, which enable developers to build their model by combining different concurrent models which include DE, SDF and FSM hierarchically. For example, a SDF block of a reactive system can be decomposed as a FSM, DE or SDF system, in which the element of the decomposed system can be further modeled as a hierarchical system. The proposed HFSM architecture enables to provide significant flexibility while considering the concurrency. The HFSM model has long been used to model reactive systems [6]. For example, the Ptolemy project [14] was using the FSM jointly with SDF and DE models to specify and simulate control functionality of real-time, embedded systems. The project has been developed over decades and its outcomes are still widely used in developing new semantics for block-diagram language [99], cyber-physical systems [27], and concurrency in software [25]. A recent example of using the hierarchical approach to develop communication system is the new domain-specific language designed by Klaus Havelund and Rajeev Joshi [59] that manages communications for the Curiosity Mars rover. However, to the best of the author's knowledge, the complete analysis and deployment of modeling MAC protocols using HFSM have not been carried out systemically.

4.3.2 Modeling MACs with HFSM

Before designing a MAC description language, the used MAC models are first listed and categorized into the hierarchical structures. The categorization is made roughly according to the depth of models, i.e., at which level the MAC protocol is decomposed and modularized. To show the general structure of the modularity level of MAC protocols, a complete classic CSMA/Collision Avoidance (CA) algorithm that used in the IEEE 802.11 Distributed Coordination Function (DCF) is modeled in different layers and shown in Figure 4.1. According to their distinct purposes, five levels of modeling, namely the protocol level, the node level, the state level, the function level, and the logic level, are categorized. The MAC behaviors in each layer are modeled differently.

At the protocol level, the interaction with higher layer protocols and the outward performances are the main focuses rather than the internal logic and signal processing. Therefore, an instance of a MAC protocol is considered as an atomic element or function of the system that maps environmental variations and inter-agents interactions to the performances, such as the SINR to PER or distance to throughput which are also mutually affected. This way, the implementation of a MAC ignores the intricate software/hardware implementations but the statistical properties which can be efficiently used for large scale network simulation and analysis. MAC protocols are normally modeled as a DE system at the protocol level.

DE is also widely used for modeling at the node level, where MAC protocols are not considered as an integrated concept but instances on individual agents. Therefore, the behaviors and interaction of the network participants are able to be modeled and analyzed. The modeling at node level focuses on the interaction inside a network

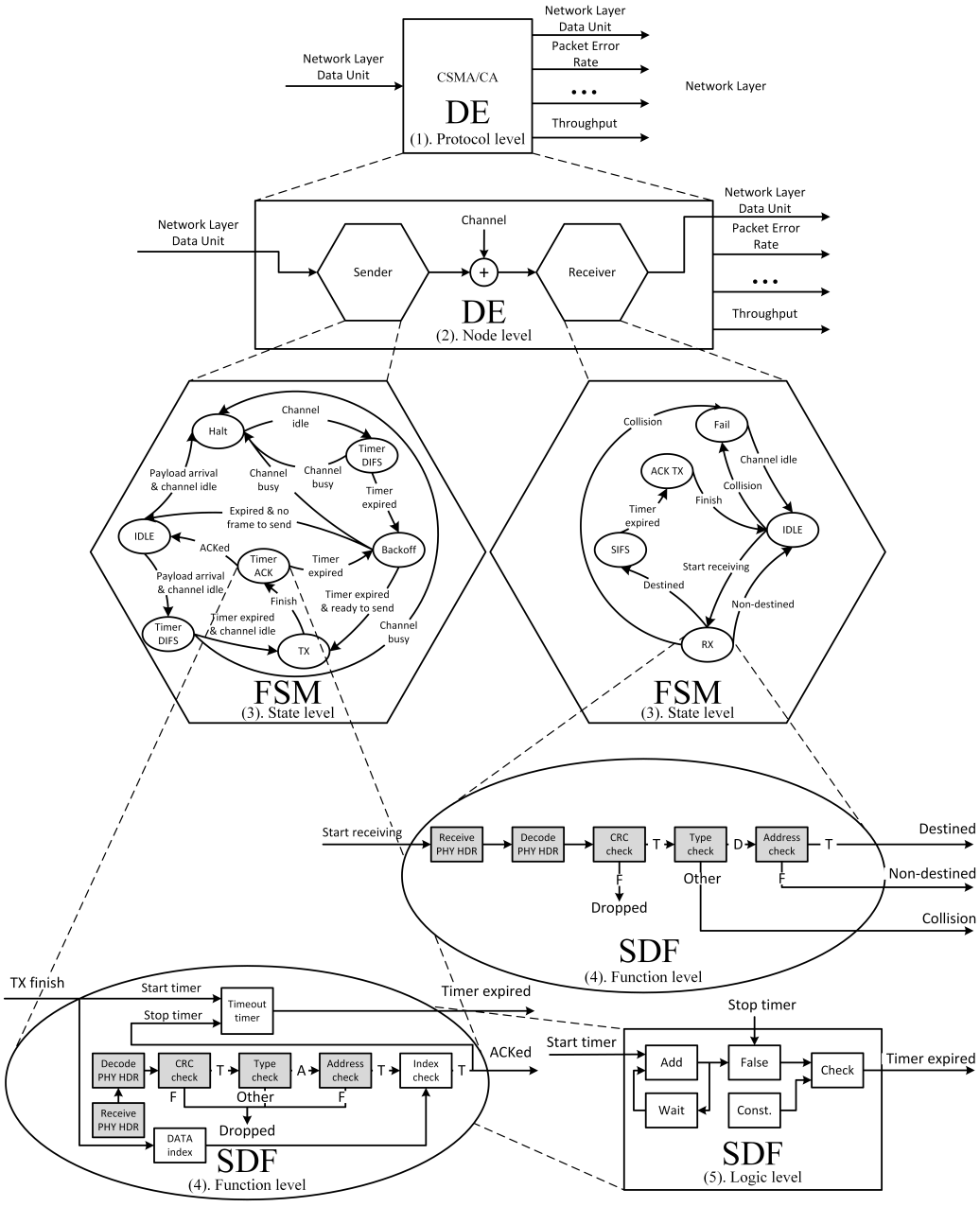


Figure 4.1: Hierarchical structure of models of the classic CSMA/CA used in the IEEE 802.11 DCF.

and the interplay of multiple agents. However, if a MAC protocol is only modeled at the node level, it is still aiming at offering services to network layer analysis which cannot be directly used for carrying out MAC implementations on hardware. It is worth noting that the majority of the software simulators, e.g., ns-3 [76], are operating at least partially in this level based on DE model.

The system operated at the state level can describe the logic control inside a node.

At this level, a MAC protocol is decomposed into states that fits the classic FSM definition. The state level contains the intricate internal logic which is executable and programmable. Therefore, it is the highest level that a real hardware based implementation can be applied. Due to the limitation on timed processing and concurrency, extensions of FSM are widely used for practical development. For example, an eXtended FSM (XFSM) is used in [105].

We name the next level function level which belows the state level. In general, states at the state level have to be first implemented as a sequence or other combinations of functions then to control the hardware. The module philosophy can also be applied at the function level so that the MAC functionalities are divided into independent blocks with fixed input and output ports. The SDF is an appropriate tool to schedule the execution order. The blocks at the function level still inherit the MAC characteristics from upper levels such as the blocks shown in Figure 4.1, named CRC check or Index check, where Cyclic Redundancy Check (CRC) and Index are typically MAC terminologies.

The blocks at the function level can be further decomposed by connected logic blocks at the logical level. The basic elements at the logic level are logic operators or atomic components such as registers. The logic level may be implemented by the FPGA in SDR platforms or directly on hardware circuit in the embedded system or other simple radio platforms.

It is worth noting that most of the existing works are not limited at one level. For example, ns-3 is defined as a DE based simulator, but many of its submodules such as the MAC high model in its 802.11 models are also implemented as a FSM, which is perfect in line with the HFSM architecture.

4.3.3 Requirements Analysis

A good MAC design tool that can be used for prototyping on hardware based platforms should have high decomposability and modularity which can enhance its flexibility and debuggability. Moreover, high decomposability and modularity is the presupposition to achieve high reusability that is also important. The protocol level and node level are barely able to use for implementing MAC functions on hardware based platforms. It is due to the fact that the DE model used at these levels mainly describes the performance and outputs of the system rather than describe the functions and detailed execution. The logic level is also not a good candidate for carrying out MAC implementation. Its elements are too general for developing a MAC protocol in a simple and straightforward manner. Developers working this level have to generate redundant code for the same functionalities in the same protocols, e.g., the different type of timers which share almost same structure but deployed in different places in a protocol.

Most of the MAC implementations are carried out at the state level [105] and the function level [47,111]. Although the implementation at these two levels share several similarities that make them sometimes indistinguishable, they still hold several key

differences. The state level focuses on what the protocol should behave, and the function level concerns more about how to execute. The reusability of the two levels are not the same. A complete 802.11 DCF FSM with both `Sender` and `Receiver` functions are shown in Figure 4.2. The FSM contains 11 states, but all of them are different from each other. Even it is obvious that some of them, e.g., the `ACK TX` and `TX` share considerable amount similarity, they are not able to be reused. On the contrary, the blocks at the function level has much higher reusability. For example, all blocks in the SDF model of the state `RX` in `Receiver` node are used by the SDF model of the state `Timer ACK` in `Sender` node.

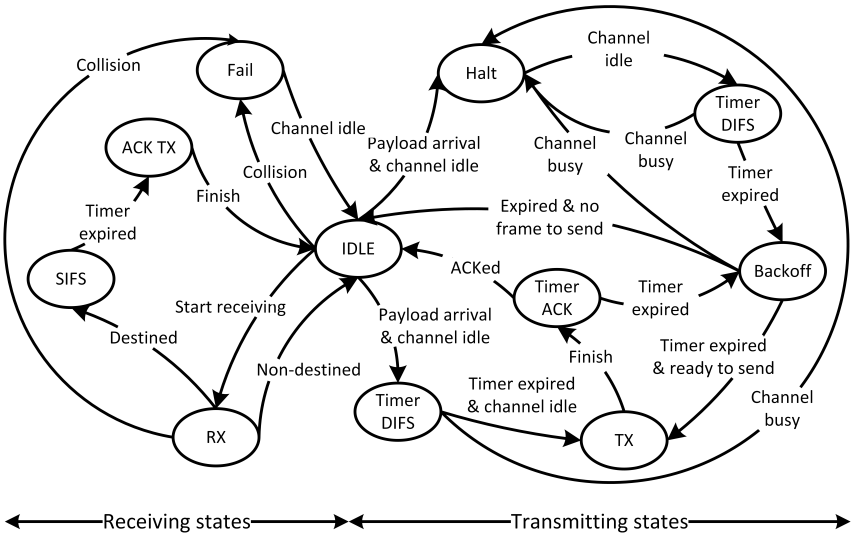


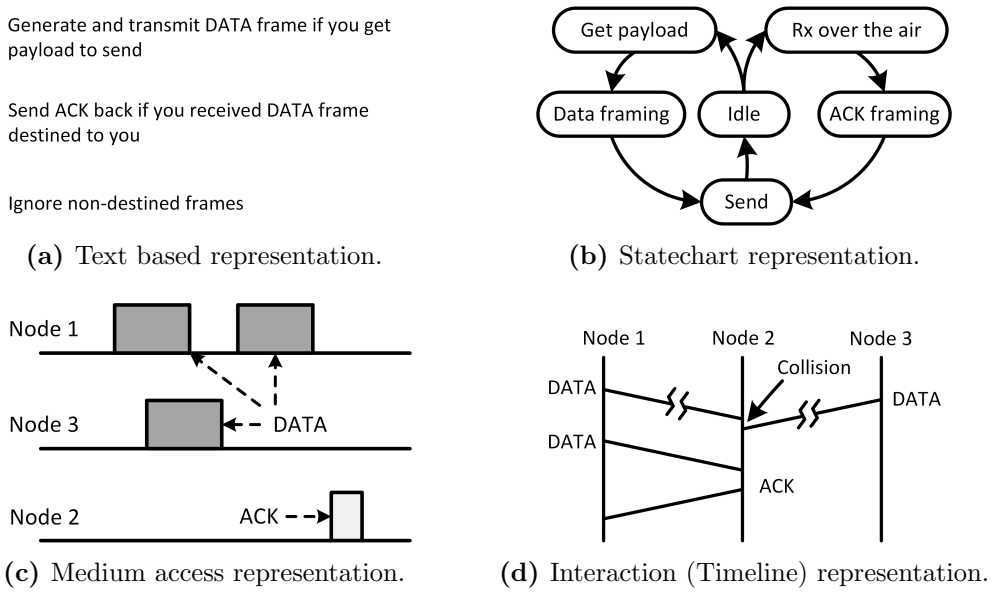
Figure 4.2: FSM of a complete CSMA MAC with both TX and RX functionalities.

Moreover, the flexibility of the MAC implementation at function level is also higher than the FSM level. For example, if the user wants to the `RX` state to also export the failed frame header, the developer must revise the source code of the state. However, in a function level realization, the user can simply read that value after the block `CRC check` and export the existing value from port `Dropped`. By carefully decomposing MAC functionalities and define interfaces, i.e., ports of blocks, the function level decomposition can achieve the highest reusability and balance flexibility and complexity. In addition, the functional level decomposition allows users to have direct access to the lower level functionalities and variables. For instance, several states at the state level of Figure 4.1 include direct PHY operations, namely `RX`, `TX`, `Timer ACK`, and `ACK TX`. Some of the states are triggered by the PHY parameters and events, e.g., `IDLE` is triggered by a detected collision in the channel or indicator from the PHY layer showing that the reception is started. The semantics of FSM cannot clearly show the access to the PHY and hardware which may lower the debuggability. Furthermore, this hidden connection may also cause a synchronization problem or signaling error which may require a complex signaling or internal communication mechanism to handle.

4.3.4 MAC Representation

A good MAC designing tool should have not only straightforward semantics but also readable representation. Moreover, if its semantics has very high readability, developers can program in a visualized way, which would be significantly improve the usability and efficiency. In other words, if the representation of a MAC protocol is executable, developing and designing can be merged into one activity which can make the whole development circle easier.

For different purposes, MAC protocols can be described and explained in various ways. Four major representations of the same ALOHA protocol with handshake are shown in Figure 4.3. The names in the caption except are given partially following the manner of Unified Modeling Language (UML) which is a language for modeling and visualizing software systems. Unlike the normal programming languages such as C++ and Python, UML is a well developed modeling standard for blueprint software systems, which is inherently capable of handling MAC protocols. The four representation are concluded from a large number of literatures and technical documents, in particular the studies regarding SDR implementation and the manuscripts of the IEEE 802.11 standards. Investigating all kind of MAC representations is not in the scope of this dissertation so only the heavily used ones are discussed.



(a) Text based representation.

(b) Statechart representation.

(c) Medium access representation.

(d) Interaction (Timeline) representation.

Figure 4.3: Major representations of MAC protocols.

The text based representation shown in Figure 4.3a is able to explain the key behaviors of a MAC protocols clearly. Moreover, it is the only way to accurately define terms such as states and events. To use them directly in other representations requires a certain level of defaultly or widely accepted definitions to avoid ambiguity. However, the text based representation show less about the connection of each part thus cannot provide an integrated picture of the system as a whole. It is also not

able to show the interactions of multiple participants of the network and the reaction of them to the environment. In general, the text based representation is more like an explanation not a description.

The statechart representation plots a MAC protocol as a directed graph. As shown in Figure 4.3b, the statechart representation composes several states and the connections among them. The detailed representation may also depend on the modeling. For example, a Petri Net based implementation [95] follows the style of visualization of Petri Net. The state chart representation can show the detailed system architecture which provides a straightforward impression of how the MAC is running. However, the state chart representation may suffer the state explosion problem [10] when used to demonstrate complex MAC protocols. The state explosion takes place due to fact that the increasing protocol variables requires exponentially increased number of states to describe the protocol.

The medium access representation in Figure 4.3c explicitly shows the channel status of a running network. It should contain some typical scenarios such as the collision or handshake to help understanding. Many classic MAC protocols, e.g., CSMA and TDMA series protocols are mainly operate in time domain, which make the medium access representation clear and effective. The drawback of the medium access representation is that it shows almost nothing of the inner logics and architectures. Therefore, this representation is more efficient in demonstrating a protocol rather than explaining and analyzing. Moreover, it is difficult to show the feasure of the protocols which are not operating in the time domain. For example, new structures have to be involved to correctly illustrate Code Division Multiple Access (CDMA) and Frequency-Division Multiple Access (FDMA) protocols.

The name of interaction representation is inspired by the sequence diagram of the Unified Modeling Language (UML) interaction diagram. As shown in Figure 4.3d, it weakens the concept of channel but show the relations among multiple participants with highlighted packet order. It is a light weighted representation of a protocol with a focus on packet exchange. It is used more in network layer protocols and logic link layer protocols but also used in MAC layer. Its shortcomings are similar to the medium access representation, plus the lack of channel interaction.

All representations have pros and cons. Text based representation makes the best of definition and highlighting key features, but is short of showing the implementation detail and inter-nodes interactions. The statechart representation is also short of showing inter-nodes interactions but opt at demonstrating the implementation detail and inner logics. The medium access representation focuses on the channel thus ideally for CSMA and TDMA types of time domain operated protocols. However, no information of implementation are given as the interaction representation. Therefore, mothods are always used together to provide a full picture of the targeted protocol in manuscripts.

Due to its module based feature, the statechart representation are widely used for showing the implementation details and algorithms of a MAC protocol. It is also convenient to visualize a MAC protocol thus it is used by many MAC implementations not only to describe but also programming a MAC protocol

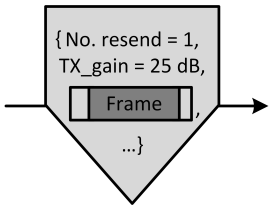


Figure 4.4: A sample of DMDL *command*.

[105, 110]. Considering that the statechart representation has been long used for describing directed graph which is the main tool for plotting the FSM and SDF, DMDL also follows the statechart representation.

4.4 Architecture

In the light of the analysis in Section 4.3, DMDL select SDF as the model to decompose MAC layer primitives at the functional level. Therefore, MAC protocols are defined as executable directed graphs with MAC components at the vertices. A MAC component represents an elementary function at the functional level of Figure 4.1 such address check or framing. The MAC component in DMDL is named block which is the semantic element of DMDL. To build a MAC protocol, users need to connect blocks with directed edges. Same as the Petri Net [55], tokens are used to convey control information among blocks via the directed edges to indicate the order of execution and deliver data which generated at the former blocks to latter ones. The token in DMDL is named *command*. In DMDL, *command* is only allowed to move following the direction of the edges. This way, the topology of the directed graph represents the algorithm of the MAC protocols explicitly.

4.4.1 *Command*

As introduced before, *command* is used to pass the information between blocks. A sample *command* is shown in Figure 4.4. The visualized *command* is used in this section to improve the readability. All used *command* in this chapter are shown in Figure 4.5. They are also introduced again when they first appear in the block descriptions in the next section. A typical *command* may have a MAC frame which is the main data object of MAC protocols, and several parameters related to the MAC frame, e.g., the address of the MAC frame and the PHY parameters in transmitting the frame. Blocks are supposed to generate or modify *command* according to its function. For example, the block to check the CRC of the received frame should add an item of the result of the CRC check into the input *command* and export the new *command* out so that the latter part of the MAC protocol may behave based on it. In DMDL implementation, the developer may implement *command* as the cluster structure in LabVIEW or the structure in C++.

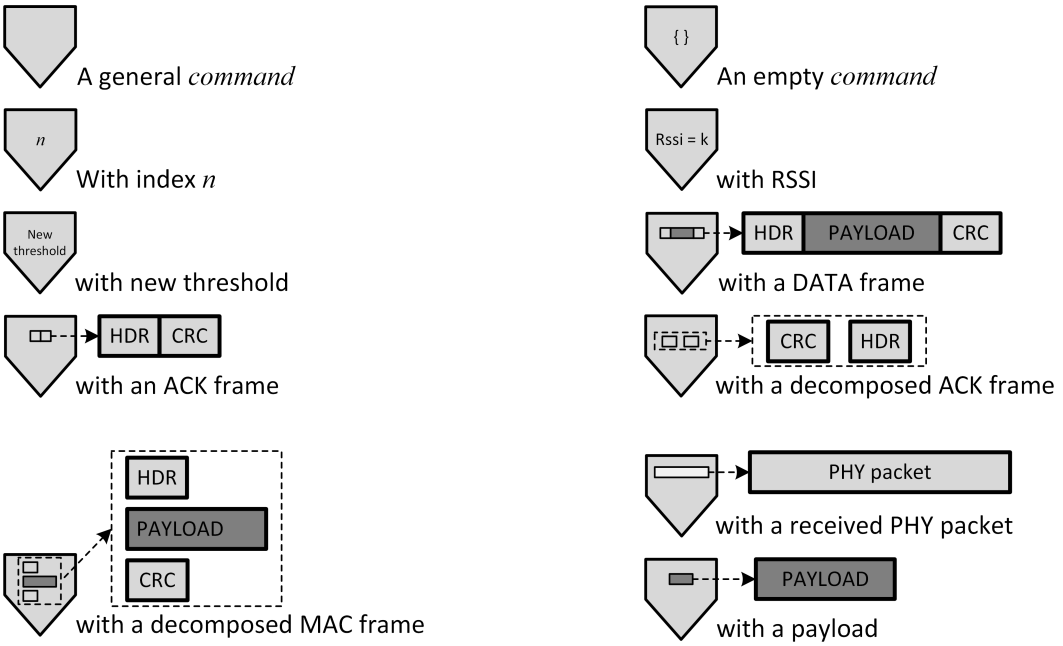


Figure 4.5: Detailed structures of *command* used in this section.

4.4.2 Blocks

Blocks are the elementary semantics of DMDL which are actions in MAC layer or responses to MAC related events. In DMDL, a typical block is shown in Figure 4.6. The block name is equivalent to the function name in conventional programming languages which briefly indicates the purpose and major usage of the block. The Properties are the configurations that users can set for the blocks before the execution. Two types of port are defined in DMDL, namely the input port and the output port. The input ports are the destination of directed edges while the output ports are the source. A block has to have at least one port regardless of their types so that it can be connected to the other parts of the protocol. *command* is always passed from output ports to input ports. An output port can connect to multiple input ports, and vice versa. Both the arrival and departure of *command* in input ports and output ports can be asynchronous, i.e., *command* can arrive at different input ports at different time. Exceptions are the time related blocks which will be elaborated later in the section. The *command* can also be passed from the output ports to the input ports of the same block. To prove the readability, all ports used in this chapter together with their full names are listed in Figure 4.7. The full name of a port is shown again when it first appears in the block descriptions.

Due to the *command* passing mechanism and the multi-ports feature, a system modeled and implemented by DMDL is inherently a concurrent system. At a specific moment, there can be any number of *command* exist and move in the system. If multiple *command* arrive at one port of a non-timed block, first-come, first-serve principle are applied. It is complicated for a timed block that new *command* arrives

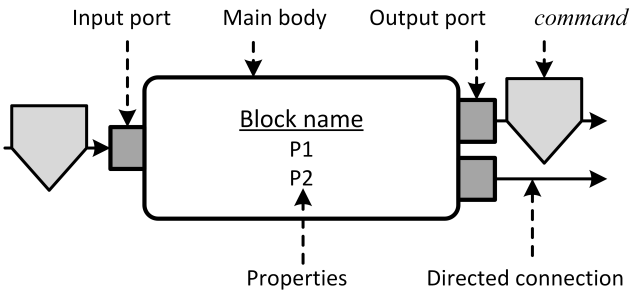


Figure 4.6: A DMDL block.

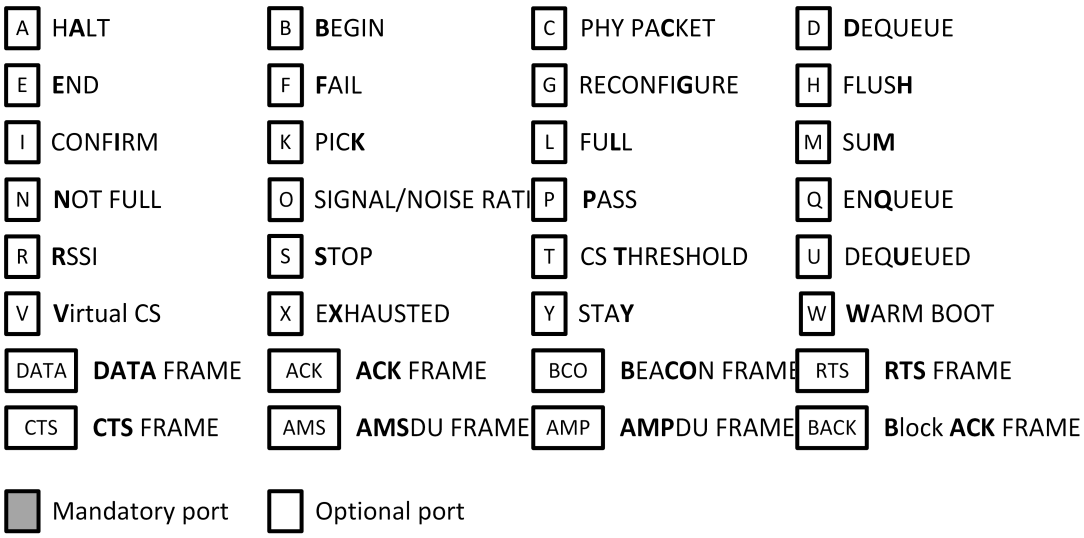


Figure 4.7: Ports and their names used in DMDL.

when the last one is still executing. In general, the timed block should report the event, so developers can improve the design. In general, blocks should have specific solutions to handle this situation. For example, if a *command* arrives at a timer when it is counting down, the *command* can be used to prolong the timer duration, or trigger an error report depending on the propose of the timer.

Due to different features and purposes, blocks defined in DMDL can be categorized into timed blocks, frame-related blocks, interface blocks and auxiliary blocks. It is worth noting that the DMDL block library is an open framework that supports and encourages the extension of the blocks list to fit future MAC protocol design. More ports are expected to be defined and included in DMDL in the future.

4.4.2.1 Timing Related Blocks

All blocks start executing after being activated by input *command*. Unlike the non-timed blocks which supposed to be executed instantaneously, timed blocks are designed to halt for a certain among of time then forward the input *command* from

its output port(s). Timer functions such as pending, stopping, or reset should be performed according to the state of the execution so they cannot be predefined in the properties. Therefore, DMDL implements these timer operations as input ports. The complete form of a timer is shown in Figure 4.8. The timer is supposed to perform the corresponding action if a *command* arrives at port B. More complex timing control algorithms can be developed based on combining different type of timers with the other blocks like counters.

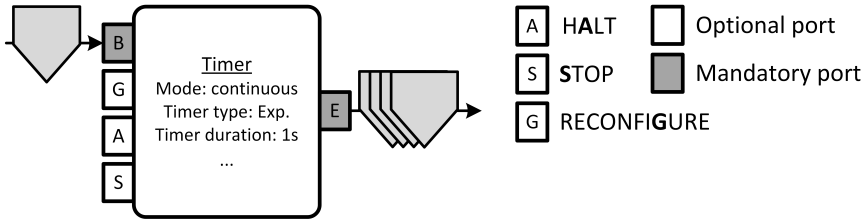
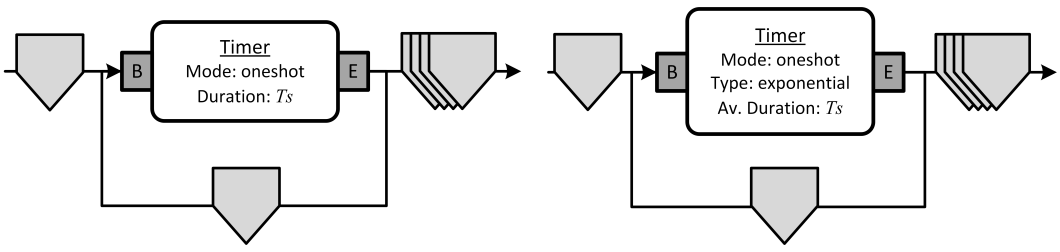


Figure 4.8: Complete form of DMDL timer.

It is common that a timer in a protocol does not require the full functionality, i.e., not all the ports are used. For example, the timer for counting down Short Interframe Space (SIFS) does not need port S. Therefore, the implementation of the timer block should be able to dynamically tailor its representation, at least be able to automatically remove the unused ports to improve the readability.

For example, a simple oneshot timer with only two ports is shown in Figure 4.9a. It waits for a preset duration T_s then forwards the input *command* without modification. Simply connecting the output port E and the input port B of the oneshot timer creates a periodic timer, which is also one instance of the simplest DMDL programming. As shown in Figure 4.9a, The exported *command* of the oneshot timer is driven to two destinations. One *command* is sent to the cascaded blocks for the future execution, another is connected to the oneshot timer again. This way, the oneshot timer is triggered by its own output thus becoming a periodic timer.

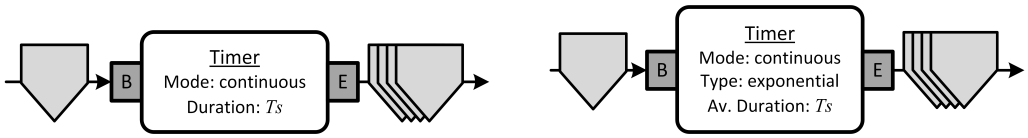
Randomization can be introduced into timers. In Figure 4.9b, a oneshot timer with exponentially distributed duration is shown. Applying the same feedback *command* path, the oneshot exponential timer is repeatedly activated so that its



(a) Periodic timer. **(b)** Poisson source.

Figure 4.9: Generate uniformly and Poisson distributed *command* using oneshot timer.

output *command* follow the Poisson distribution, which is one of the most widely used distribution in simulating packet arrival or generation.



(a) Simplified periodic timer. **(b)** Simplified Poisson source.

Figure 4.10: Syntactic sugar of uniformly and Poisson distributed *command* generation using oneshot timer.

Similar to the syntactic sugar in conventional programming languages, DMDL timer blocks are also able to provide similar simplifications. The syntactic sugar of the timer in Figure 4.9 is shown in Figure 4.10. The syntactic sugar of the timer is realized by adding the continuous mode to the timer blocks to break out a loop in the topology, which increases the readability and the flexibility. Nevertheless, the oneshot form is also useful in many situations, e.g., the timer to control the waiting time of the IFS before sending frames. Therefore, timers in DMDL have multiple modes for handling the requirements of the desired functions.

Backoff is a widely used timing operation in MAC protocols. Backoff is normally used in contention based MAC protocols which need all participants in the same shared medium to differentiate their transmissions to avoid collisions. One of the most classic usage of backoff is jointly used with CSMA/CA algorithms. Different backoff algorithms exist, e.g., constant backoff, random backoff, and binary exponential backoff. Moreover, the simple backoff acts like a pure timer that wait a certain among of time according to the inner logic and information in the input *command* such as the number of retransmissions. However, some advance backoff algorithms require the backoff to be aware of the channel status. For example, the backoff is supposed to halt when the channel is busy in the IEEE 802.11 DCF. In addition, the virtual carrier sensing should also be addressed by the backoff, because the backoff should be activated or prolonged according to the received RTS and CTS frames. It is possible using a timer and other logical operator blocks to implement backoff, however, considering such the various demands on the functionalities it would be much easier that providing a single backoff block. As shown in Figure 4.11, the backoff in DMDL has several input ports and one output port. Port B is a mandatory port which is used to trigger the countdown. Port R is an optional port which is designed to continuously acquire the Received Signal Strength Indicator (RSSI) value from the receiving/PHY related blocks. The carrier sensing threshold is able to be designed as a property, however, some dynamic carrier sensing algorithms may update the threshold in the run time, so another optional port T is assigned to it. The last port V is designed to get the timing information that the current backoff is needed to be prolonged. It is also able to trigger the backoff if the block is not triggered before.

In a MAC protocol, timeout is normally used when the protocols are expecting a response from another participant in a fixed time duration. The classic two-way

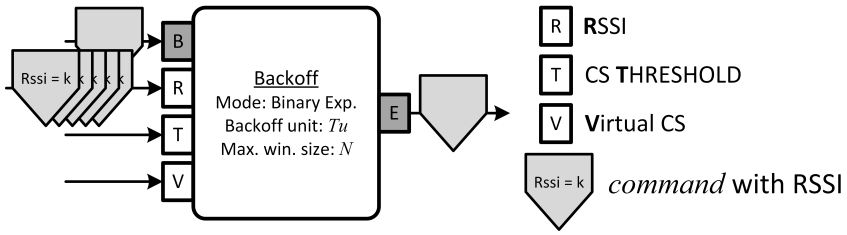


Figure 4.11: Backoff block.

DATA-ACK handshake is a good example. In the DATA-ACK handshake, the protocol should start a timeout after sending a DATA frame. The duration of the timeout is the maximal estimated time of the arrival of the ACK frame, i.e., the protocol tends to believe that ACK frame cannot be received after the timeout duration [1]. Therefore, the timeout block has to handle two types of input *command*, the one includes a DATA frame to trigger the countdown and the one includes a received ACK frame. The model of timeout in DMDL is shown in Figure 4.12. The block has two input ports, in which the first one triggers the timeout, e.g., the sent DATA frame or RTS frame, the second one clears the timeout, e.g., the received ACK frame or CTS frame, if it arrives inside the timeout duration.

To simplify the design and enhance the universal property, only one output of the is assigned to the timeout block. The output depends on to the states of the timeout, i.e., if the *command 1* successfully cleared the timeout, then the output command is the *command 2*. Otherwise, *command 2* is exported to indicate the failure of the timeout. This way, further blocks may need to distinguish the DATA frame based *command* and the ACK frame based *command*. The frame related blocks are introduced in the next subsection.

It is worth noting that the timeout block is able to not only used for handshaking schemes, but also for internal logic control where any type of trigger-confirmation mechanism is used.

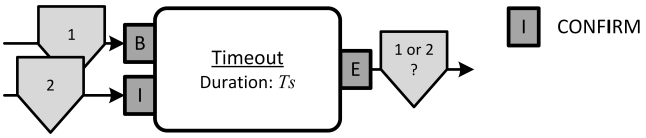


Figure 4.12: timeout block.

4.4.2.2 Frame-related Blocks

Frames are the core data element of a MAC system. In other words, a MAC protocol is a kind of data flow that connects the network layer interface and the PHY

[1] The value of the timeout depends on various aspects such as the round trip propagation delay, the processing delay at both the transmitter and receiver, and the MAC scheme related delay such as the IFS in the 802.11 DCF before sending the ACK frame, which is out of the scope of the dissertation.

layer interface. Therefore, the frame related functionalities are massively defined in DMDL.

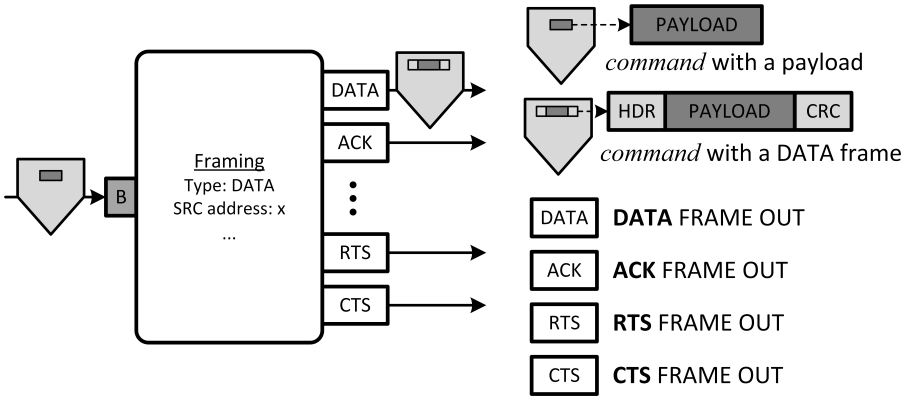


Figure 4.13: Using framing block to generate a DATA frame.

In DMDL, the series blocks for frame generation is called framing. The basic framing block has at least one input port B and one output port. To generate a DATA frame, the input port B receives the *command* with payload data while the output port DATA sends the *command* with generated frame. The process is shown in Figure 4.13, where the framing block removes the payload from the input *command* then inserts the new generated DATA frame.

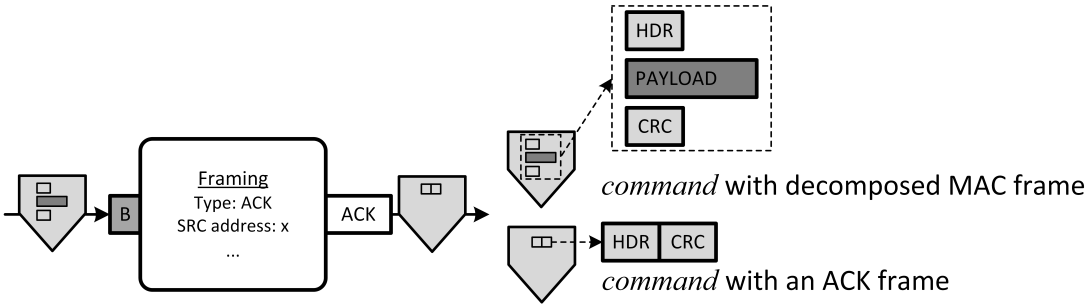
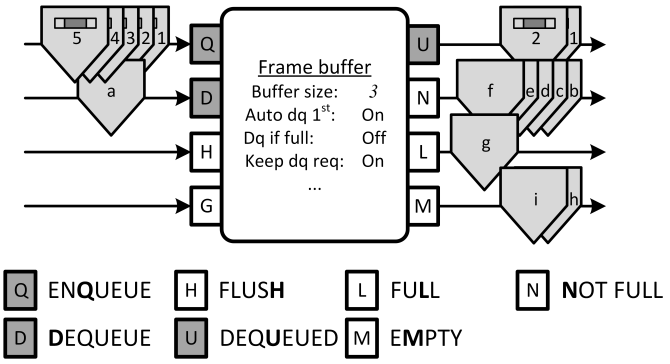


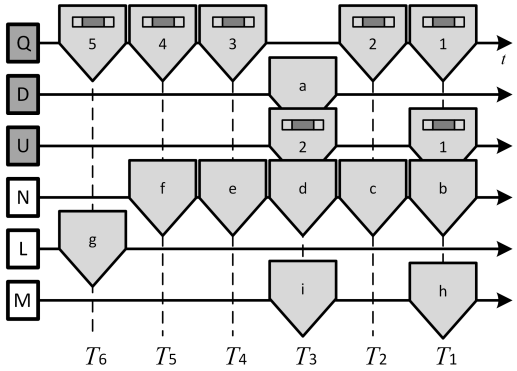
Figure 4.14: Using framing block to generating an ACK frame.

The generation of the ACK frame is shown in Figure 4.14. Similar to the DATA frame generation, the framing block removes the received DATA frame from the *command* and inserted an ACK frame instead. It is worth noting that different type of frames requires different input frame types which cannot be mismatched. Therefore, an input *command* check should be implemented before any framing operation.

In a MAC protocol, frames may not be sent immediately after generation. Therefore, the function of storing the generated frame should be considered. Such demands may exist in other places, e.g., the frame queue waiting for processing or composing a frame sequence that is common in some slide window protocols. Therefore, the frame buffer block is purposed in DMDL as shown in Figure 4.15a. The based



(a) Frame buffer block with an exemplar *command* sequence.



(b) Time line diagram of the example in Figure 4.15a.

Figure 4.15: Frame buffer block and its behaviors.

structure of DMDL is a queue whose element is *command*. Three mandatory ports are required, namely Q, D, and U, respectively. The Q port receives *command* and push the elements into the queue. If a *command* arrives at the D port, the queue pop one element up and export it through output port U. Several optional ports and properties are also defined. The input port H can be used for flushing the queue to empty the buffer. The reset size port G can be used for the buffer to reset the queue length according to the input *command*, which may be useful in some protocols that need to create a frame sequence with varying length. Output port N can be used to send out *command* to indicate the vacancy in the queue, and port M can be used to pass the information that the current queue is empty to the next block.

The *command* attached to the frame buffer block shows an example of how the *command* is joint work with *command* and properties. A clearer version of the example is shown in Figure 4.15b. The Auto dq 1st property enables the frame buffer to dequeue the first enqueued frame. This property is useful at the beginning phase of a protocol so that no other mechanism is needed to dequeue the first frame. At T_1, frame 1 arrives at the Enqueue port. Due to the Auto dq 1st property, frame 1 is immediately dequeued. In the meantime, *command* b is exported from port N because the buffer is not full after exporting frame 1. Moreover, *command* h is

also exported from port M because the queue is empty. It may be useful to pull the frame source for more frames. At T_2, frame 2 is arrived. Because frame 2 is not the first frame, therefore it is not dequeued. Similarly, *command* c is exported from port N. At T_3, *command* a arrives at the D port so that the frame buffer pops frame 2 from the U port. In the meanwhile, *command* d and *command* i are exported from port N and port M, respectively. At the moment, there is no frame is buffered in the queue. At T_4 and T_5, frame 3 and frame 4 arrive at the Q port, thus be pushed into the queue. Because the frame buffer is still not full after accepting them, therefore *command* e and *command* f are exported from port N. At time T_6, frame 5 arrives at port Q and is pushed into the queue. At this time, there are three frames, namely frame 3, frame 4, and frame 5 are buffered in the queue with size 3. Therefore, *command* g is exported from the port L to indicate the cascades block that the buffer is full.

Because the element of the frame buffer block is *command*, therefore the frame buffer block can store not only the MAC frames, but also payload, PHY packet, or even the MAC command or parameters if they are encapsulated in a *command*.

Many frame related blocks are on the receiving side. The MAC layer needs to convert the bit stream which provided by the PHY layer to the payload and other useful information. Moreover, the MAC layer also needs to validate the received frame and make further decisions. Therefore, the following frame processing blocks are mandatory for any DMDL implementation.

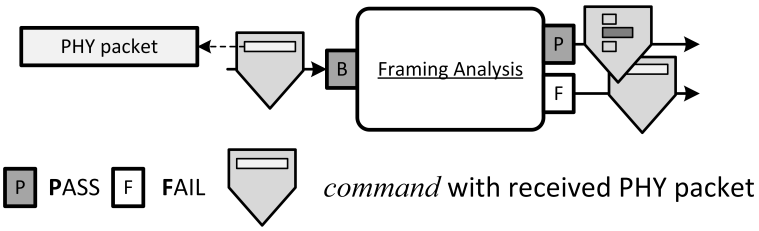


Figure 4.16: Frame analysis block.

The frame analysis block is supposed to decompose the received PHY packet and get the information in the MAC header. By using the frame length and frame type information in the frame header, the frame analysis block is able to cut the payload and the CRC out of the PHY packet. The frame analysis block replaces the PHY packet with the decomposed MAC header information, payload and CRC sequence. Failure may occur during the decomposition which normally means the corrupted MAC header information, e.g., unknown frame type, meaningless frame length, etc. One optional port F may add to forwards the input *command* to the next block if the frame header is not correct.

As always emphasized, DMDL is an open language which allows researchers to add new blocks in future. The currently defined frame processing blocks include,

- ▷ Frame Check block: checks the validation of a received frame by recalculating its CRC and compared with the received one (cf. Figure 4.17a).

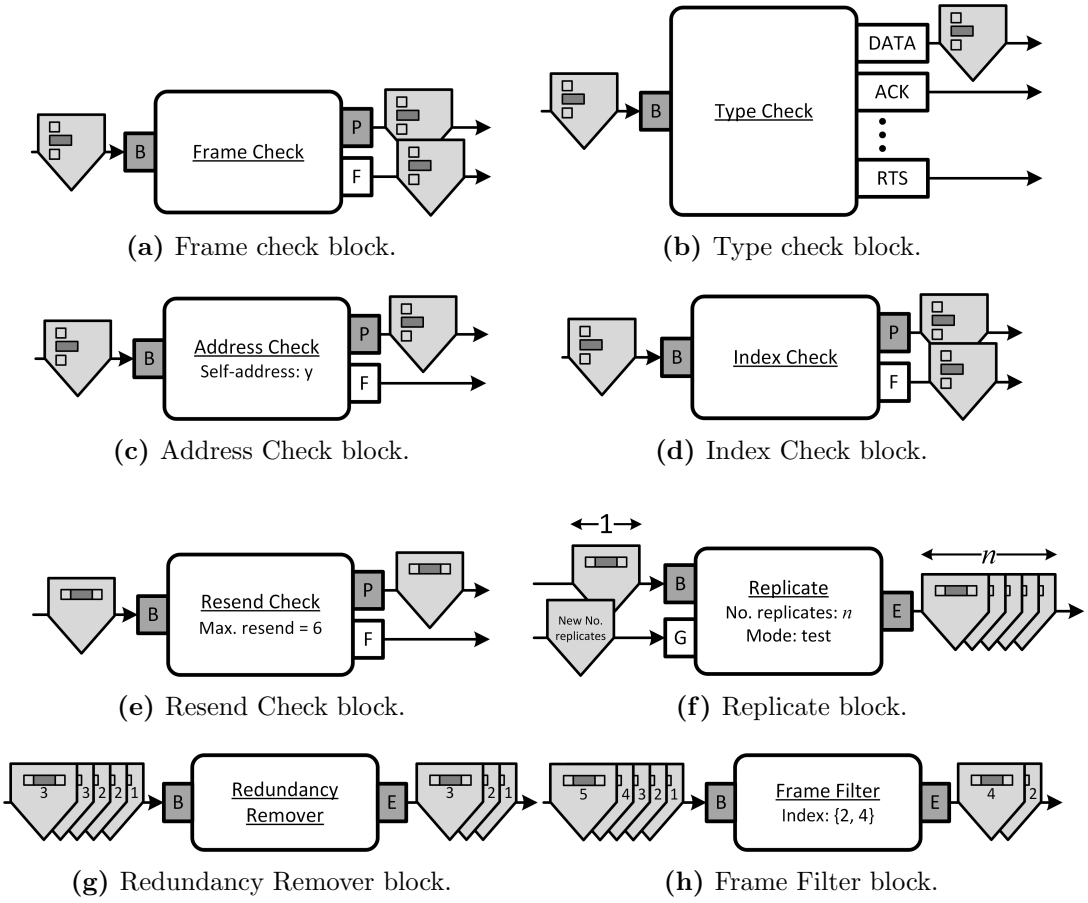
- ▷ Type Check block: distinguish the frame type and assign them a different output path (cf. Figure 4.17b).
- ▷ Address Check block: gets the destination address of the received frame and compared with the its own address (cf. Figure 4.17c).
- ▷ Index Check block: checks the received frame index with the last received frame index to ensure the seamless reception (cf. Figure 4.17d).
- ▷ Resend Check block: checks the number of retransmissions of the input frame with the maximal number of retransmissions (cf. Figure 4.17e).
- ▷ Replicate block: replicates the input *command* according to the predefined property. The number of replicates should be able to be reset in the run time (cf. Figure 4.17f).
- ▷ Redundancy Remover block: removes the repeatedly received frames. It buffers the previous received frame index and compares it with new arrived frames (cf. Figure 4.17g).
- ▷ Frame Filter block: Only allows frames with pointed frame index to pass (cf. Figure 4.17h).

4.4.2.3 Interface Blocks

Currently, three types of interface blocks, i.e., the Sending block, the Receiving block, and the Carrier Sensing block, are defined. Both the Sending block and the Receiving block include the corresponding PHY functionalities. In other words, the PHY of the system is hidden behind the two blocks. The Sending block sends the frame arrives at its port B. One optional port G may be necessary if the MAC protocol has a cross layer design. The PHY and radio configurations can be changed by using this port. The selected optional output port out can be used to pass the sent frame to the following block such as the timeout or slide window control. In the MAC protocol implementation, the developer should export the *command* only after the real physical transmission ends so that the following logics can be triggered, e.g., the timeout.

The Receiving block is supposed to have the full functionality of the receiving PHY. If DMDL is implemented on a hardware based platform or SDR, it should be able to get the PHY packet from the radio waveforms and at least provide the RSSI and SNR signal so that other MAC blocks are kept being aware of the radio environment and channel conditions. The Receiving block should be synchronized with the Sending blocks in terms of the PHY and radio properties to ensure valid transmissions.

The RSSI value provided by the Receiving block can be used by the Carrier Sensing block to determine the channel condition. As shown in Figure 4.19, the sensing is triggered by any arrived *command* at port B. In carrier sensing, the detected signal strength from port R is compared with the carrier sensing threshold to decide the



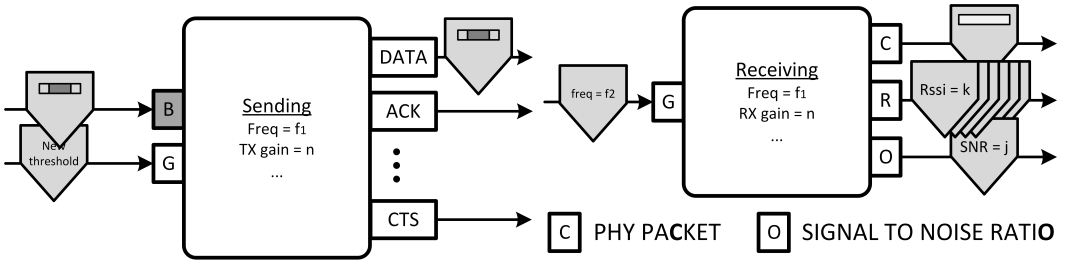
(a) Frame check block. **(b)** Type check block.

(c) Address Check block. **(d)** Index Check block.

(e) Resend Check block. **(f)** Replicate block.

(g) Redundancy Remover block. **(h)** Frame Filter block.

Figure 4.17: Major frame processing blocks.



(a) Sending block. **(b)** Receiving block.

Figure 4.18: The interfacing blocks connecting with PHY layer and the hardware.

channel condition. As shown in the Figure, the carrier sensing block is selected in fixed duration mode. Therefore, the block starts to comparing the input RSSI after triggered by receiving a *command* for a fixed duration n [ms]. In the sensing duration, any RSSI higher than the threshold leads to the failure of the carrier sensing. After remaining in idle for the whole duration, the input *command* which normally with a frame to send is exported from the port P. Otherwise, the *command* is exported from the port F. Port T outputs the carrier sensing threshold in case any adaptive threshold algorithm is applied inside the Carrier Sensing block.

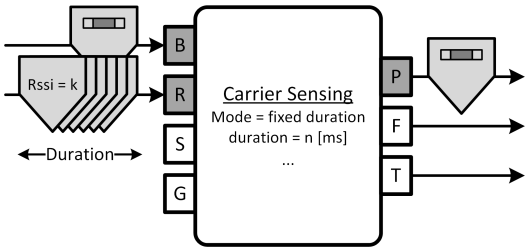


Figure 4.19: Carrier sensing block.

Different carrier sensing algorithm may required by the different MAC protocols. Several widely used carrier sensing modes addressed by the DMDL carrier sensing blocks are listed below.

- ▷ Fixed duration: it requires the Carrier Sensing block to stay in sensing mode for a fixed duration. In the duration, any channel busy event terminates the sensing and output the input *command* immediately from the port F.
- ▷ Oneshot: it only requires the Carrier Sensing block to read the next coming RSSI. The comparison between it and the threshold determines the status of carrier sensing.
- ▷ Unlimited sensing: it forces the Carrier Sensing block to stay in the carrier sensing mode until the channel turns to idle.

4.4.2.4 Auxiliary Blocks

Many auxiliary blocks are also designed in DMDL to enhance the capability of DMDL in modeling different types of MAC protocols.

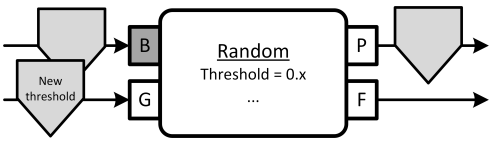


Figure 4.20: Randomization block.

Randomization is the key function in opportunistic MAC protocols. DMDL address the randomization in different blocks. As mentioned in the timer blocks and backoff

blocks, randomization is used there to generate random time durations. In general, the randomization is performed by the random block shown in Figure 4.20. The block has a property named threshold which is a pure number between 0 and 1. If the threshold is x, for any input *command*, it is sent to output port P with probability x and to output port F with probability $1 - x$. The function can be jointly used with timer block in p-persistent CSMA to perform the randomized transmission after detecting the channel turns to idle.

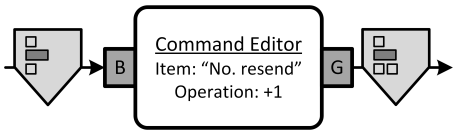


Figure 4.21: Command Editor block.

In DMDL, *command* is the token kind of signaling element used to convey data and states among blocks. In principle, *command* should be generated, modified and removed from the system in MAC blocks. However, in some special cases, it would be much easier to directly operate on the *command*. To fit the demand, DMDL also includes the Command Editor block. As shown in Figure 4.21, user should be able to specify one item in the *command* and modify it. In a real implementation, the developer should realize this block according to the software environment.

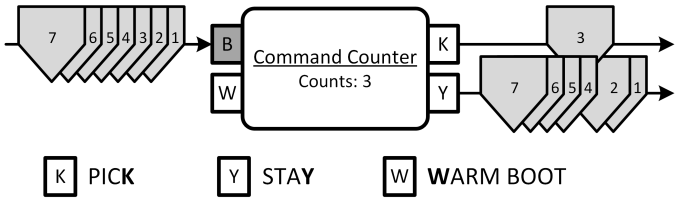


Figure 4.22: Command Counter block.

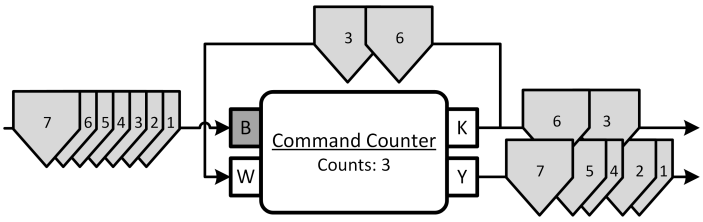


Figure 4.23: Using Command Counter block to pick up frames whose index is the multiple of 3.

The Command Counter block is designed to pick up the specific *command* in the arriving *command* sequence. As shown in Figure 4.22, the property Counts is set to 3, so the third arrival is picked up and exported from port K. All other *command* are exported from port Y. To enable iteratively selecting the *command*, the Command Counter block support to reset the counts, i.e., when a *command* arrives at the port W, the block clears the current counting and recount from zero. Therefore, by simply connect the port K and W, the Command Counter block is able to pick up

all *command* whose positions in the input *command* as the multiple of the Counts.

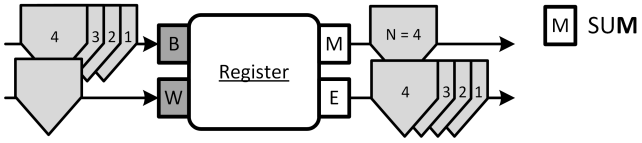


Figure 4.24: Register block.

The Register block is used to record the number of *command* has arrived at port B between the arrivals of the *command* at the port W. As shown in Figure 4.24, the Register block may export one *command* with the information of how many *command* has arrived in the duration and may also export all received *command* from port E in the duration successively.

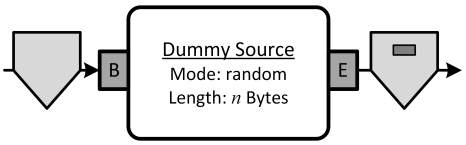


Figure 4.25: Dummy Source block.

The payload of a MAC frame is the upper layer data units, e.g., the IP packet. However, in some MAC researches, the source of the payload is not the main focus. Moreover, the demands on testing different payload size or traffic pattern may arise in evaluating the performance of new designed MAC protocols. Therefore, DMDL has the Dummy Source block which can generate user specified payload.

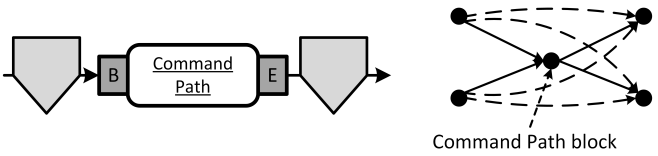


Figure 4.26: Command Path block.

In a MAC coded by DMDL, it may occur that multiple blocks are supposed to have a one-to-one connection to every block in another group of blocks. In this case the number of connections should be equal to $N \times M$, where N and M are the number of blocks in the two groups. Command Path block (cf. Figure 4.26) is designed to alleviate this problem by letting all source blocks first connect to it, then forwards all input *command* to all sink blocks. The command Path block itself has no effect on the *command*. By applying the Command Path block, the number of connections is reduced to $N + M$ with a much clearer topology.

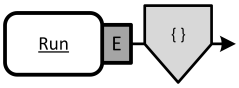


Figure 4.27: Run block as the first block in a MAC which drives the whole implementation.

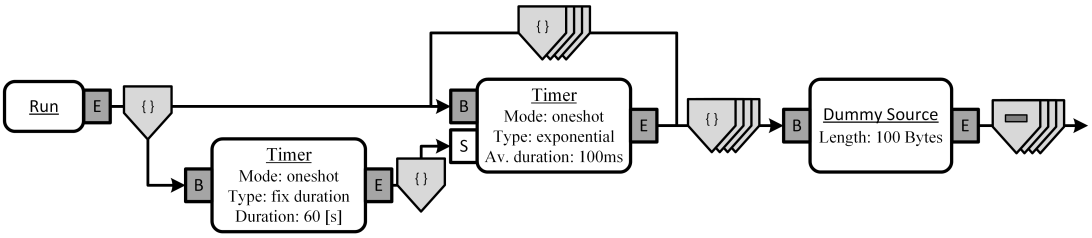


Figure 4.28: Using DMDL to design a simple Poisson source with packet arrival rate 10/s and the payload length is 100 Bytes. The source only generates payload for 60 seconds.

In order to keep the topology and execution order clear and manageable, normal DMDL blocks start executing only when it is activated by input *command*. It can be found in the block representation that almost all blocks have at least one mandatory input port expect the Receiving block which is supposed to be continuously fed by the PHY layer. The intuitive question may arise here is that how is the first *command* generated. To solve the problem and start running the MAC, the Run block is defined in DMDL. As shown in Figure 4.27, it supposed to generate only one empty *command* at the time when the MAC starts to run.

4.5 Examples

In this section, several DMDL designed MAC schemes are demonstrated and explained. The examples are either a complete MAC protocol, or some key components which may used by many MAC schemes.

4.5.1 Time limited Poisson Source

Poisson source is one of the most used type of source. Due to its clear and straightforward mathematical representation and properties, it is widely used in both theoretical analysis and practical experiments. In Figure 4.28, the generation of payload is controlled by the exponential one shot timer. The timer also feedback its output to its port B to enable unstoppable generation. Moreover, the example uses another timer with duration 60 s to control the Poisson source. Therefore, the Poisson source starts generating payload after the program starts, and stop generating after 60 s.

4.5.2 p-persistent CSMA

1-persistent CSMA includes more blocks and has a more complex structure. Similarly, the protocol is driven by the Run block and then generates a dummy payload. After that, a MAC DATA frame is then generated using the payload, and sent to perform carrier sensing. The Carrier Sensing block is set to unlimited mode

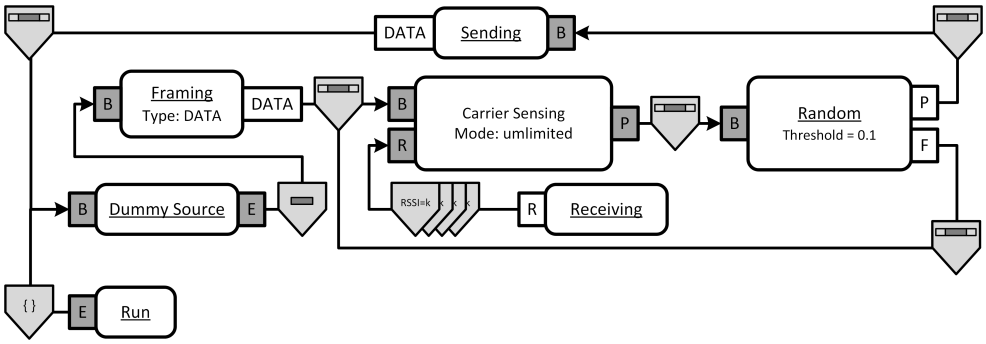


Figure 4.29: Using DMDL to design the p-persistent CSMA.

so that it holds the coming *command* until the channel is free. After detecting a free channel, the *command* is passed to Random block which has 0.1 probability to pass and 0.9 to fail. The passed *command* is sent to the Sending block and sent out. After sending, it is passed back to the Dummy Source block to generate the next dummy payload, then everything repeats again. If the *command* does not pass the Random Block, it is sent back to the Carrier Sensing block and retake carrier sensing.

The hybrid control mechanism of the DMDL is also shown in the example. On one hand, items in the *command* is able to be used as the variables for blocks to make a decision. For example, port R of the Carrier Sensing block reads the RSSI value from the imported *command*. If the imported frame does not contain the RSSI information, the carrier sensing is therefore not able to be performed. It should be avoided in the protocol design because it arises confusion. In the hardware based DMDL implementation, the developer should address this missing/wrong *command* problem by reporting an error to the protocol designer. On the other hand, the arrival of *command* is also defined as an event and can be used for the block to make a decision. For example, the *command* exported by the Sending block is passed to the Dummy Source block to generate the next payload. It is obvious that the payload is independent from the last sent DATA frame, so the Dummy source requires nothing from the imported *command*. In contrast, the arrival of the *command* at the port B itself is a trigger for generating the new payload. This hybrid control mechanism guarantees the precision of the states and parameters transfer among blocks but also simplifies the representation of a protocol.

4.5.3 IEEE 802.11 DCF

To show how to use DMDL to design and represent a complex MAC protocol, we build a IEEE 802.11 DCF liked protocol [2] as shown in Figure 4.30.

[2] Due to the limits of SDR and software development environment, we cannot implement the exact IEEE 802.11 compliance protocol using DMDL. The protocol we demonstrate in this section follows the 802.11 DCF principle with all of the major functionalities such as timeout, exponential backoff, interframe intervals, and two way handshake.

The design uses 19 DMDL blocks to build a complete IEEE 802.11 DCF liked protocol, including the whole receiving functionalities DATA-ACK two-way handshake. To better demonstrate the protocol, *command* with key items in them is also added into the figure. Similar to the *p*-persistent CSMA, the protocol is supposed to be activated by the empty *command* generated by the Run block. The *command* is filled with a dummy payload in the Dummy Source block. After framing, the *command* is passed to the Carrier Sensing block and the Timer which waits for DIFS. The Carrier Sensing block is set to unlimited mode, which means that the carrier sensing can only be terminated by detecting a busy channel or the arrival of a *command* at the port S. After Carrier Sensing block, the *command* is exported from the port P to the Sensing block if the channel keeps idle during the whole DIFS before the Timer block expires, or exported from the port F if channel turns out to be busy during the DIFS. The first route leads to the Sending block which transmits the frame in the coming *command*, and the latter route leads to the Backoff block which postpone the transmission to avoid the collision. After being sent, the Sensing block receives export the *command* from its port DATA if the *command* contains a DATA frame. The *command* with the DATA frame is buffered by the Timerout block until the corresponding ACK frame is arrived or the timeout is expired.

If the ACK frame is received during the timeout duration, the DATA frame is acknowledged. The Timeout block exports the new arrived ACK frame to the Frame Type Check block. Otherwise, the previously buffered DATA frame is passed. The Type Check block delivers the ACK frame to the Backoff block with only the minimal backoff window size to enhance the fairness among nodes.

If the expected ACK frame is not received on time, the Timeout block passes the *command* with the DATA frame to the Type Check block. The route from the port DATA of the Type Check block is connected to the Backoff block.

Besides the non-ACKed DATA frame, the DATA frame that fails in the carrier sensing has to also perform backoff. During the backoff duration, the Backoff block also reads the RSSI and halts the countdown if any channel activities are detected. After backoff, the *command* is passed to the Resend Check block to check the number of resend. If the number of resend is lower than the predefined threshold, the *command* is further passed to the Command Editor, where the item 'No. resend' is added one. Then the updated *command* is sent to the Carrier Sensing block to repeat the carrier sensing. If the DATA frame fails in the Resend Check block, then it is passed to the Dummy Source block to generate the new DATA frame, i.e., it is dropped.

We also implemented it using the GNU Radio and performed extensive testing using USRP devices. The implementation and experimental results are shown in Chapter 5.

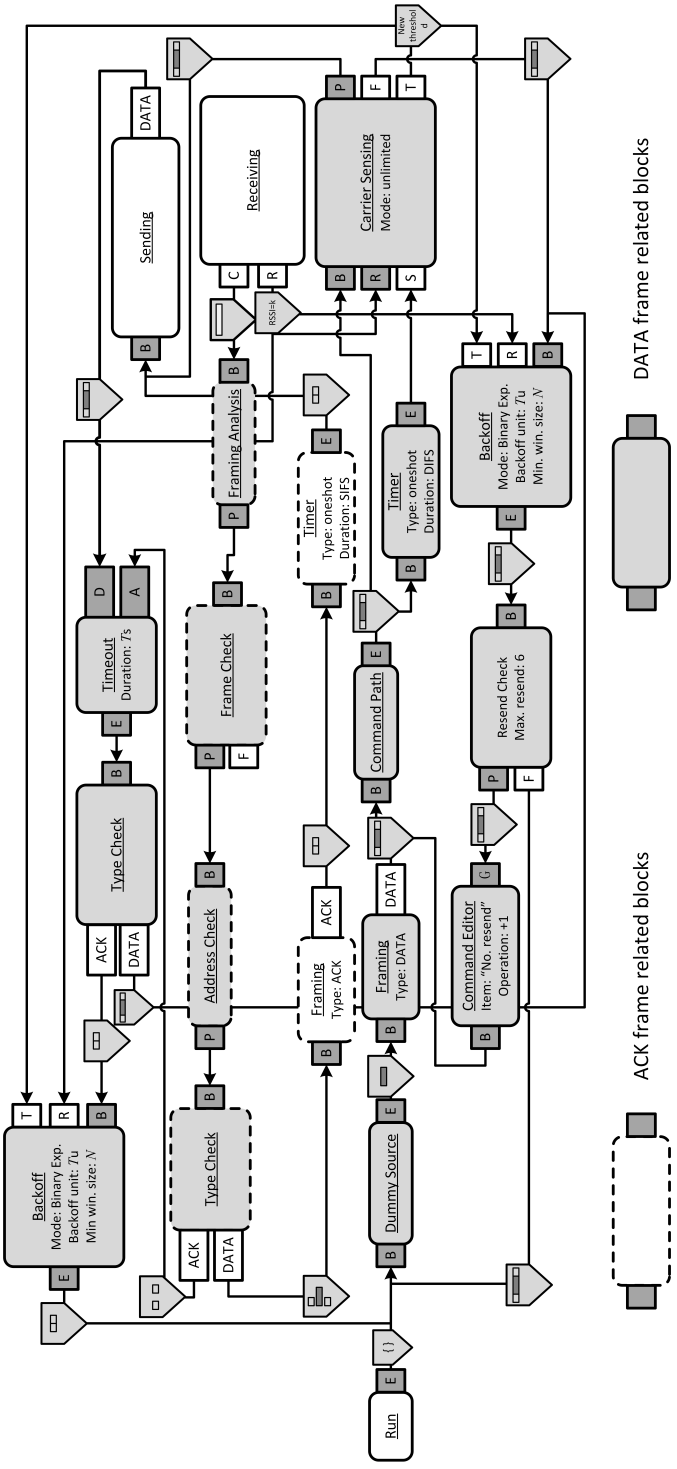


Figure 4.30: DMDL implementation of the classic IEEE 802.11 DCF.

4.5.4 Discussion on FSM and SDF

In Section 4.3, the reasons of implementing MAC components at the function level not the state level are issued. The conclusion can be further validated using the above examples. The Poisson source shown in Figure 4.28 is a typical dataflow structure which cannot be easily modeled by the general state machine models. It is difficult to insert this piece of functions into the CSMA/CA shown in Figure 4.2. The most similar place could be that the code for state transition which appears between `IDLE` state and `Timer DIFS`. Nevertheless, the Poisson source is hardly to be visualized and implemented in a pure MAC FSM. It is owing to the fact that The Poisson source is not exactly part of the MAC logic because it emulates the upper layer behaviors. However, it is important functions for MAC researchers to test their protocols so it definitely should be addressed in a MAC tool.

The hierarchical architecture and the hidden FSM structure can be seen from complex protocols. The classic CSMA/CA protocol is both represented by the SDF model in Figure 4.30 and the FSM model in Figure 4.2. The first thing can be observed is that some of the distinct states share the same MAC functions. In Figure 4.30, dashed border means the block is an ACK frame related block, which belongs to the left side of the state machine in Figure 4.2. And the gray blocks are related to DATA frame transmissions, which means that they are located in the right part of the state machine in Figure 4.2. However, many blocks in Figure 4.30 such as the Frame Analysis block, the Frame Check block and the Address Check block have both identities. Therefore, they are marked by both dashed border and gray filling. The overlapping shows that the states in FSM cannot correspond to the actual functionalities which are normally hardware based processing blocks, such as the FPGA logic or on-board registers. Therefore, the FSM based MAC language may cause overprogramming or resource wastage.

The second observation is that the mapping between the FSM states and the MAC functions are still representing. For example, the `Timer ACK` state is can be mapped to the `Timeout` block, and the following `Type Check` block is the transition which decides which state to go after the waiting. Therefore, the hierarchical architecture is still able to be seen from the SDF based implementation that the behaviors of the MAC protocol follow the control sequence of the upper level FSM which is hidden.

4.6 Summary

In this chapter, we demonstrate the DMDL implementation. The elements of DMDL, the MAC function blocks are defined and elaborated in detail. The implementation of several classic protocols are demonstrated by connecting the predefined MAC blocks. The implementations themselves demonstrate that DMDL is able to realize complex MAC protocol in an ultra clear and straightforward way. Despite of the different architectures, most of the SDR platforms have at least certain amount of computational capacity either on board or in host PC to support complex programming. Therefore, it would be very realistic to say that DMDL

is capable of being developed on many SDR platforms using various programming languages. To show the feasibility of developing DMDL on commercially available SDR platforms, we developed it in GNU Radio. Based on the GNU Radio version of DMDL, we show the implementation of the IEEE 802.11 DCF and CogMAC+ protocol with performance of large scale measurements including multiple wireless nodes. The results show that DMDL is able to trace the execution of a MAC protocol comprehensively, which is ideal for MAC prototyping and research.

5 DMDL Implementation in GNU Radio

In order to verify the feasibility of implementing DMDL on commercially available SDR platforms, we developed GR-DMDL, the DMDL implementation in GNU Radio which can be used to drive multiple SDR platforms including USRPs. We release the the source code of GR-DMDL in GitHub with instruction manual and examples [80]. GNU Radio is one of the most popular SDR development environment that can setup and configure signal processing in a graphical way. It has a powerful set of built-in signal processing blocks, which provides very high PHY layer reconfigurability. In combination with USRP and many other SDR platforms, users can easily test and prototype new ideas on real hardware and realistic radio environments. GNU Radio allows users to add blocks into GNU Radio as out-of-tree modules, which provides a very friendly environment for realizing DMDL. Our implementation is fully compatible with all versions of GNU Radio after version 3.7.9 and supports various NI USRPs including USRP2 and USRP-2953R.

5.1 Blocks

In this section, we show the GNU Radio implementations of some important DMDL blocks including their GUI, interfaces and functionalities. Thanks to the support of GNU Radio, the architecture of the implemented blocks is identical to the definition in DMDL and even their appearances are quite similar. Please note that the development of DMDL library is on going, so the online library may contain more blocks than what is described in this dissertation.

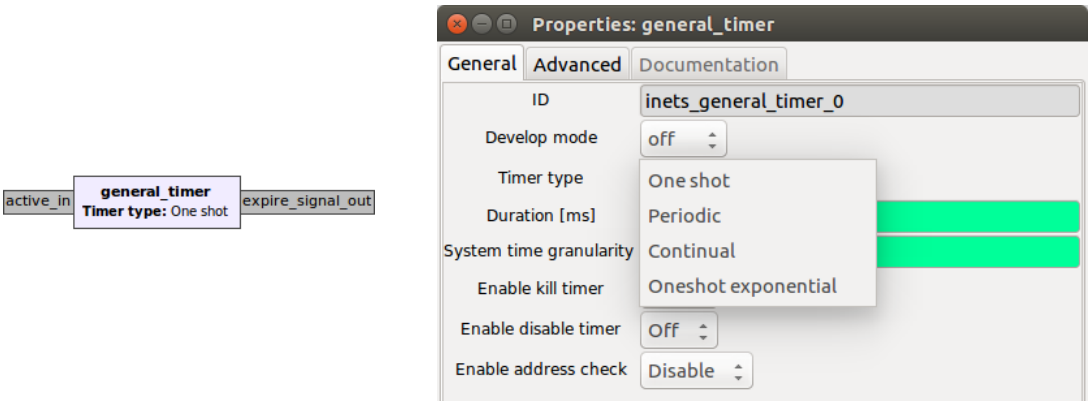
5.1.1 Timing related blocks

Timing related blocks are developed according to the DMDL design. The waiting functionality is implemented in C++ so that the accuracy can be lowered down to the level of 10^{-5} seconds.

5.1.1.1 The general_timer block

The **general_timer** block[1] is shown in Figure 5.1a and its properties are shown in Figure 5.1b. Several optional modes are designed to ease the complexity in the further protocol design. Currently the **general_timer** block has four modes, namely

[1] In order to distinguish the block name used in the definition of DMDL and GR-DMDL, the block names used in GR-DMDL are lowercase and in bold.



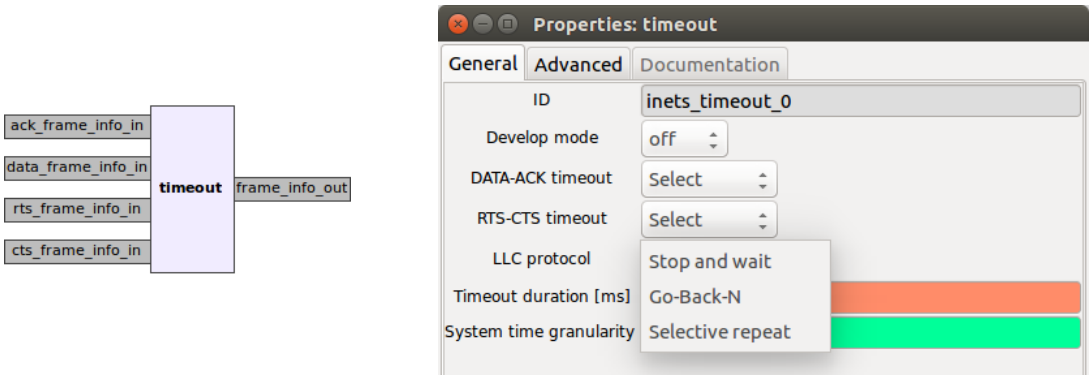
(a) The **general_timer** block. **(b)** Properties of the **general_timer** block.

Figure 5.1: The GUI of the **general_timer** block in GRC.

Oneshot, *Periodic*, *Continual*, and *Oneshot exponential*. The waiting time is set by the parameter *Duration [ms]*. In the *Oneshot* mode, the **general_timer** block buffers the input *command* and sent it out after the *Duration [ms]* milliseconds. *command* in GR-DMDL is implemented by the Polymorphic Types (PMTs) of GNU Radio so it can contain unlimited amount of information. For more information about the *command* and signaling system in GR-DMDL, please refer to Subsection 5.2. To set the timer duration in run-time, user need to send a *command* which contains a field *timer duration [ms]* to the **general_timer** block. Then the waiting duration is set to the value in the coming *command* in millisecond instead of the predefined duration. In the *periodic* mode, the timer sends the input *command* periodically until a *command* is received from port *kill_timer_in* which is hidden in Figure 5.1. Users need to set the property *Enable kill timer* to **On** to activate the port. The *Continual* mode is designed for the slide window protocols that every received frame may prolong the timer duration to the preset duration in parameter *Duration [ms]*. The last mode, i.e., the *Oneshot exponential*, does not wait for a constant duration as in *Oneshot* mode. In this mode, the waiting time is an exponentially distributed variable with the value in property *Duration [ms]* as the rate parameter. As shown in Figure 5.36a, the **general_timer** block is used for generating Poisson distributed traffic.

5.1.1.2 The timeout block

In general, the *timeout* block is used to implement handshake schemes. Currently, GR-DMDL supports two classic applications, i.e., the DATA-ACK handshake and the RTS-CTS handshake. If these two are jointly used, users are able to design 802.11 like four way handshake protocols. For example, as shown in Figure 5.2, if the *command* containing a DATA frame arrives at the *data_frame_info_in* port, the **timeout** block is activated, i.e., the **timeout** block starts counting down. If the corresponding ACK frame arrives before the counting down finishes, the DATA frame is considered successfully acknowledged. The arrived ACK frame is exported



(a) The **timeout** block. **(b)** Properties of the **timeout** block.

Figure 5.2: The GUI of **timeout** block in GRC.

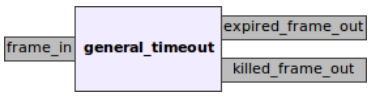


Figure 5.3: The **general_timeout** block.

from the *frame_info_out* port. However, if the ACK frame does not come back on time or the wrong ACK frame is received, the **timeout** block export the buffered DATA frame to indicate that the two way handshake is not ended successfully.

5.1.1.3 The general_timeout block

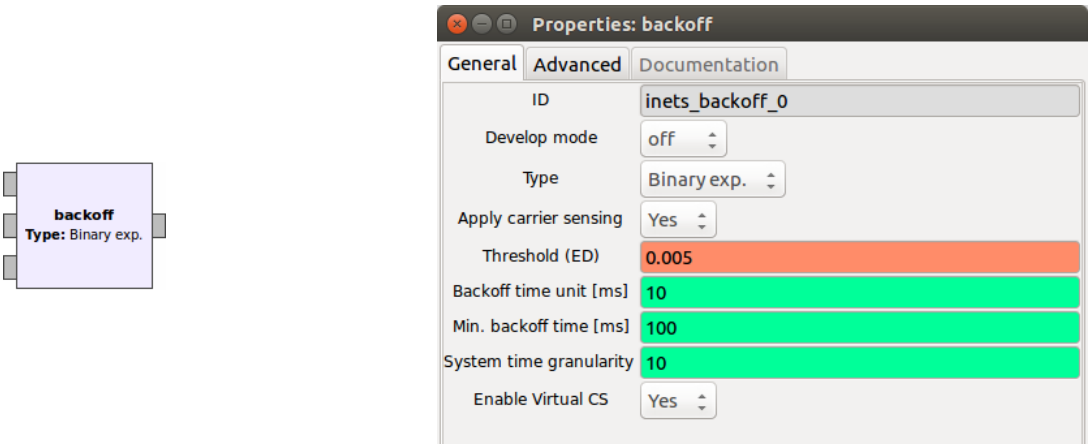
Considering the demands of other handshake mechanisms, a more generalized timeout timer is developed. The **general_timeout** block has one input port which is described in Figure 5.3. The rule of the **general_timeout** block is very simple, i.e., it always uses the latter received *command* to acknowledge the former.

5.1.1.4 The backoff block

The **backoff** block is also implemented in GR-DMDL (cf. Figure 5.4). It is implemented according to the backoff module defined in Figure 4.11 with multiple backoff modes. We also developed carrier sensing as an optional function which can be configured via the properties. It allows the **backoff** block to continuously check the channel status, which is required in some protocols such as the IEEE 802.11 standard. The *Threshold (ED)* can be preset but also able to be reconfigured using import *command*.

5.1.2 Frame related blocks

In this section, we first show the MAC and PHY packet structures. Then we introduce the frame related blocks which generate and process MAC frames in



(a) The **backoff** block. **(b)** Properties of the **backoff** block.

Figure 5.4: The GUI of **backoff** block in GRC.

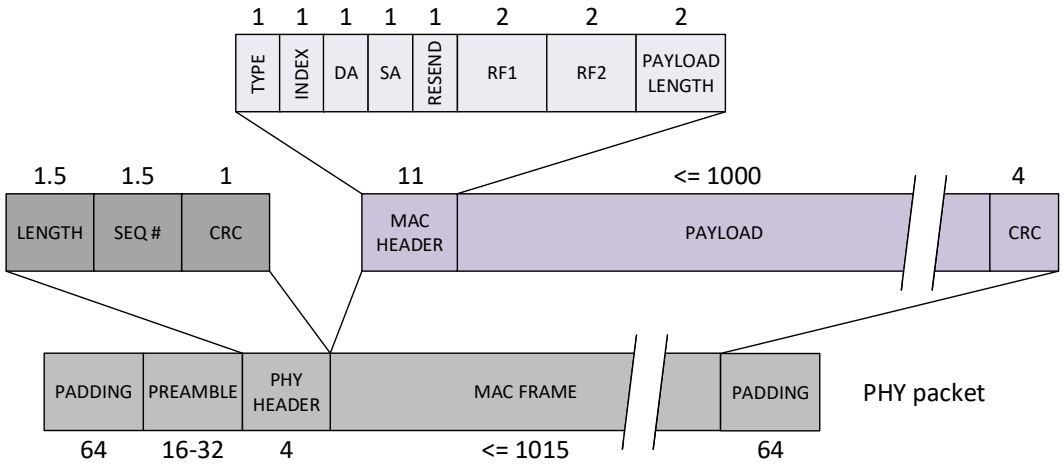


Figure 5.5: Structure of MAC frame and PHY packet.

GR-DMDL.

5.1.2.1 The framing block

GR-DMDL is targeted to conduct MAC layer protocol design and prototyping. It accepts the upper layer data units and encapsulate them into MAC frames. The encapsulation procedure is called framing, therefore the block is named as **framing** block. The frame structure used in GR-DMDL is shown in Figure 5.5, in which the PHY packet structure is obtained from [49]. Please refer to Subsection 5.1.3 and [49] for detailed explanation on PHY implementation and PHY packet operations. The lengths of each field in MAC frame and PHY packet are shown in bytes attached to blocks in Figure 5.5. Without loss of generality, GR-DMDL deploys a general frame structure which contains a MAC frame header, payload, and the CRC sequence in the end. The frame header contains multiple fields which stores the general information

such as address and sequence number to support MAC layer operations. The default lengths of all fields are shown in Figure 5.6. Users can customize them according to their own research purposes. We note that the lengths of the fields also have to be changed on the receiving side to ensure correct frame header decoding. All fields in the MAC frame header are listed below.

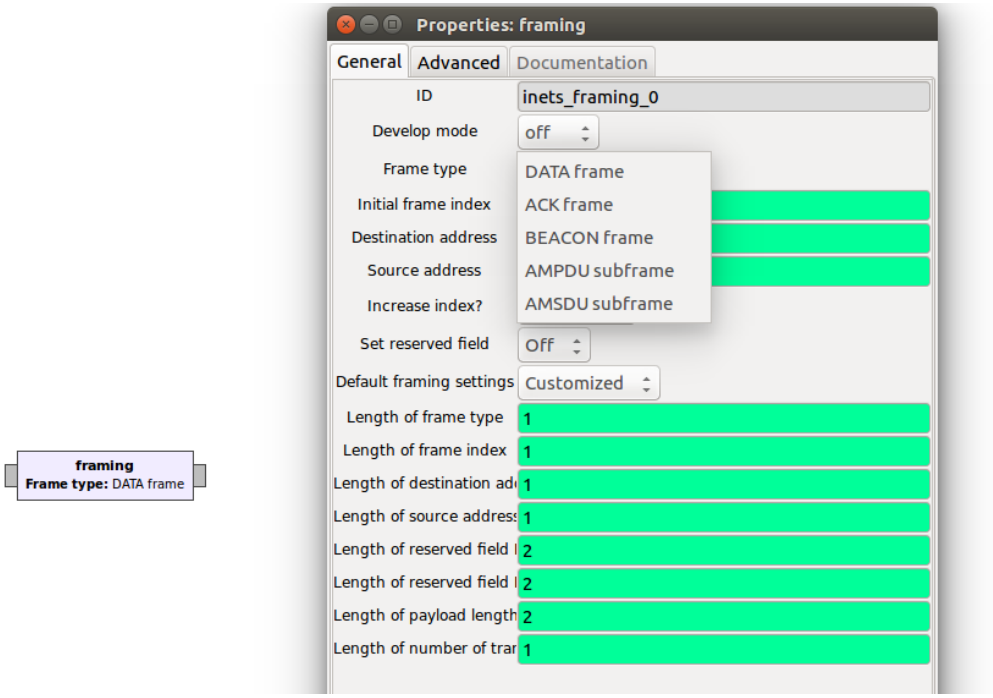


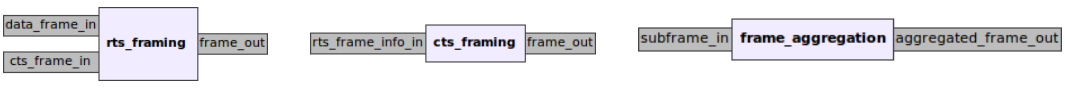
Figure 5.6: The **framing** block and its properties.

- ▷ Frame type (TYPE) field. The field is used to specify the type of the frame.
- ▷ Frame index (INDEX) field. In order to distinguish the term 'sequence number' which is widely used in all layers of communication systems, the term INDEX is used in GR-DMDL for identifying the order of the frame in the frame generation procedure. As shown in Figure 5.6, the assignment of frame index can also be controlled in **framing** block. Three frame index assignments have been developed, namely increasing, decreasing and constant. Users may also set the initial frame index from the properties of the **framing** block in GRC. The assigned frame index is also used as the constant frame index in the constant frame index assignment.
- ▷ Destination Address (DA) field. The field is used for the receiving node to understand the destination of the frame, or the addressing method that the transmitter is using. In GR-DMDL, setting the destination address field to all 1s means that the frame is a broadcasting frame. Otherwise, all specific addresses mean the frame is an unicasting frame.
- ▷ Source Address (SA) field. This field is used to store the sender's address.

- ▷ Number of resending (RESEND) field. This field is reserved for all backoff or retransmission related functionalities. Its functionality and usage are given in detail with the introduction of the **resend_check** block.
- ▷ Reserved Field 1 (RF1) and Reserved Field 2(RF2). These two fields are left blank for future user demands. Any protocol developer is able to easily insert up to 4 bytes information in these two fields.
- ▷ Payload length field. The payload length field stores the length of the payload so that the **frame_analysis** block knows how to cut the payload off the received bit sequence. The default field length is 2 bytes; therefore it is supporting the payload length up to 65535 bytes that much more than 1000 bytes which is the maximal supported frame size of the used USRP devices.
- ▷ CRC field. Even though the CRC does not lie in the front the frame, it behaves like part of the frame header. It is in the end of frame after the payload to check the validity of the frame after receiving.

The reverse operation of framing is to analyze the received frame and get all the information from the frame header. This functionality is done by the **frame_analysis** block which is shown in Subsection 5.1.2.2.

In GR-DMDL, DATA frame, ACK frame, BEACON frame, AMSDU subframe, and AMPDU subframe are generated in **framing** block as shown in Figure 5.6. If the *Frame type* field of the **framing** block is selected to *DATA frame*, the framing block generates a DATA frame using the input bit sequence as the payload. ACK frame is generated if the *Frame type* field is selected as *ACK frame* and the input *command* contains a received DATA frame. Besides the aforementioned types, RTS and CTS frames are generated in **rts_framing** block and **cts_framing** block, respectively. RTS frame is generated if the input *command* contains a DATA frame. The generated RTS frame is passed to the next blocks immediately after generation, while the DATA frame is stored in the build-in buffer. Once the corresponding CTS frame is received from the *cts_frame_in* port, the DATA frame is passed to **sending** for transmission. If the pending RTS frame is not correctly acknowledged by a CTS frame, the current DATA frame is dropped. The AMSDU subframe and AMPDU subframe are aggregated in **frame_aggregation** block.



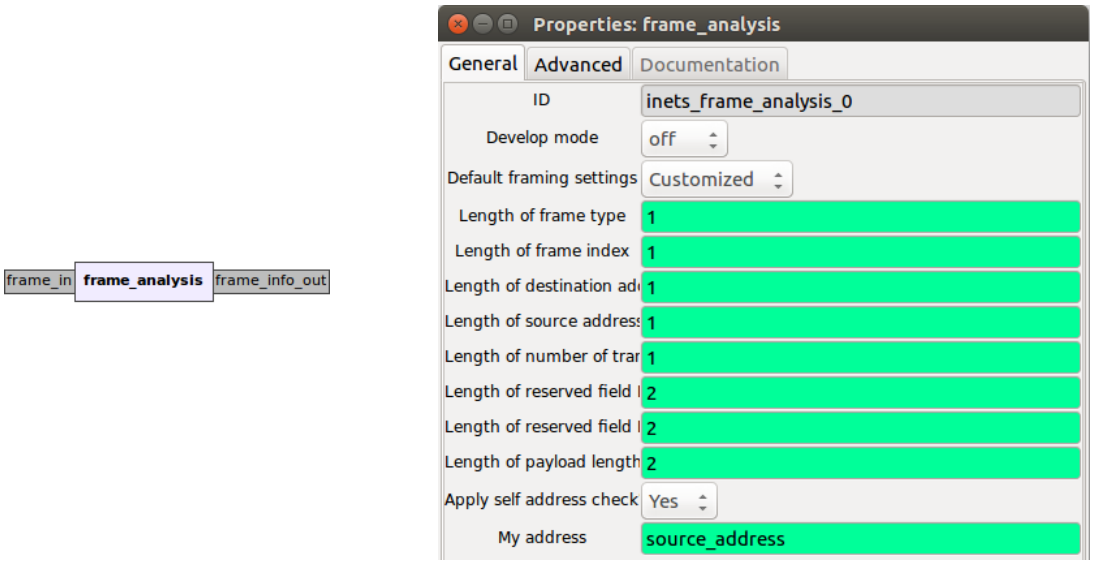
(a) rts_framing. **(b) cts_framing.** **(c) frame_aggregation.**

Figure 5.7: Other blocks used for framing.

5.1.2.2 The frame_analysis block

After receiving, the information in the received MAC header should be decoded and compared. The validity of the received frame should also be checked. Therefore,

several blocks are designed to analyze the information of the received MAC frame.



(a) The **frame_analysis** block. **(b)** Properties of the **frame_analysis** block.

Figure 5.8: The GUI of **frame_analysis** block in GRC.

The **frame_analysis** block is used for decoding the received PHY payload, i.e., the MAC frame. It first cuts the MAC frame header out according to the framing settings. As shown in Figure 5.8b, a group of lengths of all fields in the MAC frame header is already set as default. All default field lengths are equal to the default settings in the **framing** block and other framing related blocks. The decoded MAC frame header information, e.g., the source address, the destination address, and the frame type, including the MAC layer payload which is cut according to the value in the payload length field, is saved in a *command* and passed out from the *frame_info_out* message port for later blocks to use.

5.1.2.3 The frame_check block



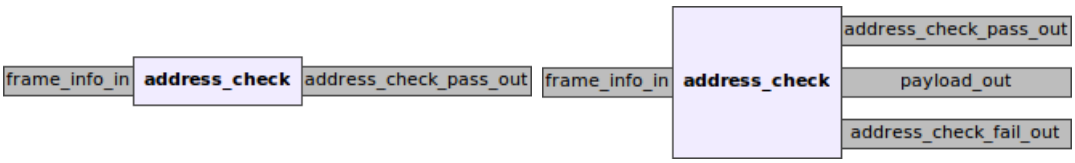
(a) **frame_check** block. **(b)** Complete form of **frame_check** block.

Figure 5.9: The GUI of **frame_check** block in GRC.

The CRC of the received MAC frame is checked in the **frame_check** block. In this block, a new CRC sequence is recalculated according to the received MAC frame by the same polynomial in the **framing** block. The new CRC is compared with the received CRC. If these two CRC sequences are the same, then the *command*

is exported through the *good_frame_info_out* port. Otherwise, it is exported from the *bad_frame_info_out* port which need to be manually activate. Moreover, if the frame is correctly received, the **frame_check** block also enables user to read it MAC payload from port *payload_out*.

5.1.2.4 The address_check block



(a) **address_check** block. **(b)** Complete form of **address_check** block.

Figure 5.10: The GUI of **address_check** block in GRC.

In MAC protocols, especially the protocol addresses unicast and multicast traffic, addressing may determine the state transition of the MAC state machine when frames are received. Therefore, **address_check** block is designed to separate the path of the MAC flowchart according to the received frame addresses. As shown in Figure 5.10a, the basic function of the **address_check** block is to filter the frames. If the destination address of the received MAC frame is equal to the address of the receiving node, then the frame is considered as a successfully delivered frame and the input *command* is exported from the *address_check_pass_out* port with a new created parameter good_frame with value 1. Otherwise, the input *command* is exported from the optional port *address_check_fail_out* with *good_frame* set to 0. Similar to the **frame_check** block, **address_check** block also enables user to read payload from its *payload_out* port.

5.1.2.5 The frame_index_selector block



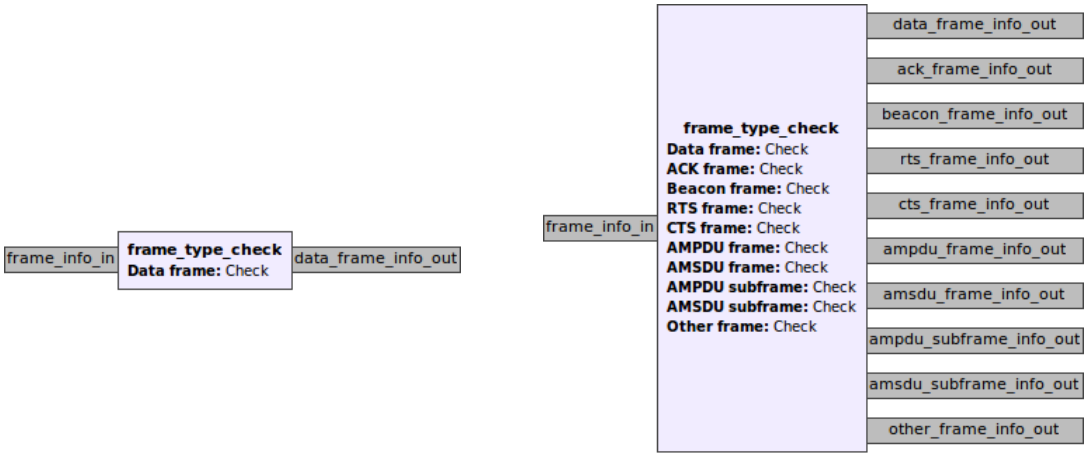
(a) **frame_index_selector** block. **(b)** Complete form of the **frame_index_selector** block.

Figure 5.11: The GUI of **frame_index_selector** block in GRC.

Frame index is the frame identification in the MAC layer. Nodes use the frame index field in the frame header to mark the frame before transmission, so that the receiving nodes know the order of the coming frame and how to acknowledge it. Besides using it in the handshake schemes, frame index can be also used in other protocols. For example, the frame index in CogMAC+ protocol [83] are used for counting down the back-to-back transmission. Therefore, the **frame_index_selector** block

is developed to filter the frame with a selected frame index out. As shown in Figure 5.11, the **frame_index_selector** block is able to pick the frame with a predefined frame index out of all received frames. Its complete form shown in Figure 5.11b has two ports which can transmit the state of the whole protocol differently according to the received frame index.

5.1.2.6 The frame_type_check block



(a) The **frame_type_check** block.

(b) Complete form of the **frame_type_check** block.

Figure 5.12: The GUI of **frame_type_check** block in GRC.

MAC protocols should have distinguished strategies for handling different types of frames. Therefore, a block which can categorize imported frames and export them in different ports is necessary in GR-DMDL. The **frame_type_check** block is developed to fit the demand. It gets the frame type information from the imported *command* and exports the frame from the corresponding output ports. The **frame_type_check** block currently supports all existing frame types which are generated in the **framing** block, the **rts_framing** block and the **cts_framing** block. Users should modify the **frame_type_check** block if they defined new type of frame or modified existing ones.

5.1.2.7 The beacon_interpreter block



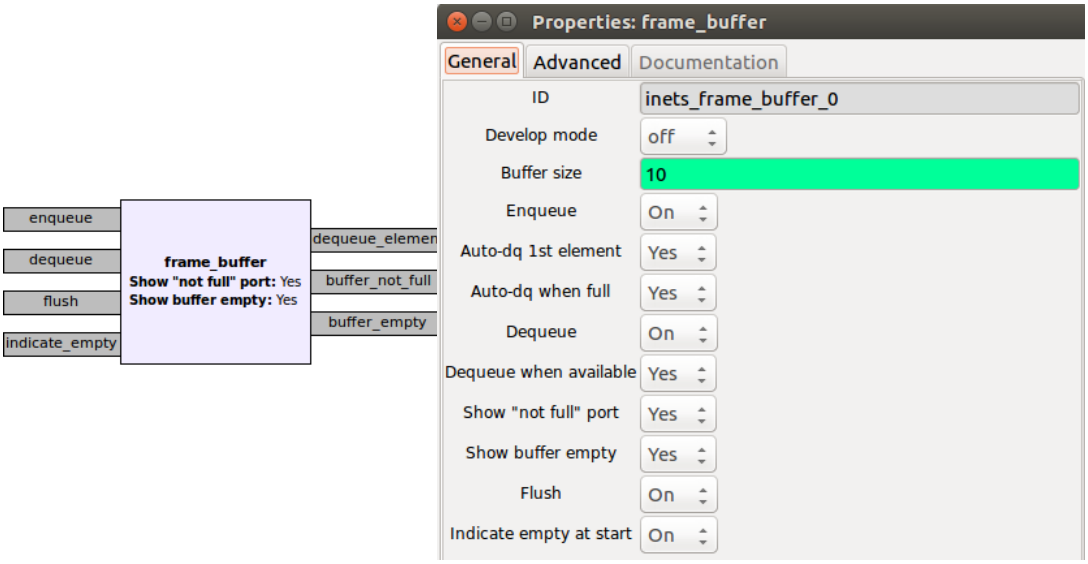
Figure 5.13: The **beacon_interpreter** block.

Unlike the RTS frame and CTS frame that have a specific design purpose in the IEEE 802.11 types of protocols to execute four-way handshake, BEACON frame is a more general frame which can be used for carrying any general information.

In GR-DMDL, the BEACON frame is used for the classic TDMA protocol time scheduling as an example. In the design, the payload of the BEACON frame is used for containing the time slot information which can be found in **framing**. The **beacon_interpreter** block decomposes the payload of received BEACON frame and gets the scheduling information. If the receiving node finds its own address is scheduled, the **beacon_interpreter** block exports a special type of *command*, namely *tdma_info* which describes the slot duration and the starting time after the reception of the BEACON frame. The other part of the protocol may make strategies for different TDMA protocols by using this information.

5.1.2.8 The Buffer block

Due to the scheduling and other time domain strategies, MAC frames are not always sent immediately after being generated. In addition, some MAC layer algorithms, e.g., the frame aggregation, require the joint information from multiple frames. In order to fit the demand, the **frame_buffer** block is developed for buffering frames. Its GUI and properties are shown in Figure 5.14.



(a) The **frame_buffer** block. **(b)** Properties of the **frame_buffer** block.

Figure 5.14: The GUI of **frame_buffer** block in GRC.

The core data structure used in the **frame_buffer** block is a queue. The the operation of input and output are simple and follows the strict rules of the queue, i.e., First in First Out (FIFO). In order to achieve the basic enqueue and dequeue operations, the input message port *enqueue* and output message port *dequeue* are developed. The port *enqueue* pushes received dictionary message into the queue if it arrives at the port. It is worth noting that the buffer does not distinguish whether the input message has a bit sequence or not. On one hand, it offers high freedom in protocol design that the buffer can even be used as a *command* buffer which

can used for scheduling the protocol execution. On the other hand, it may easily cause programming error if users connect the blocks carelessly. The port *dequeue* actually has two modes. If the input message is an integer number, then it exports the number of frames continuously from the *dequeue_element* if there are greater than or equal to this number of frames stay in the buffer. If the input message ports receive a message in any other forms, the buffer simply exports one buffered element from the *dequeue_element*.

In general, the buffer operations in a MAC protocol could be very complex. Therefore, we developed several properties and ports to control the **frame_buffer** block. As shown in Figure 5.14a, the **frame_buffer** block flushes all the elements in its queue immediately. The *indicate_empty* property lets the **frame_buffer** block export a message from the *buffer_empty* port so the following blocks know that the buffer is empty. The *buffer_not_full* port exports an empty *command* if the number of elements in the buffer is lower than the property *Buffer size* and the property *show "not full" port* is set to *Yes*.

The *auto dq 1st element* property automatically exports the first received elements from the *dequeue_element* for the entire execution of the protocol. This function is designed to be the 'first drive' of the protocol. That is, when the protocol starts running, the **frame_buffer** block can dequeue the first arrived frames without the dequeue *command* which is normally designed as the end of transmitting last frame, e.g., the successfully acknowledged or retransmitting frames. The property *Auto-dq when full* is designed to enable the buffer to abandon the long waiting frames, if the newer frame is available and the buffer is already full. The property *dequeue when available* enables the buffer to store the dequeue request when the buffer is empty. The next enqueue element will be exported directly if the dequeue request is already in the buffer. The property *Indicate empty at start* enables the **frame_buffer** block to automatically export an empty *command* from the *buffer_empty* port to ask for input frames.

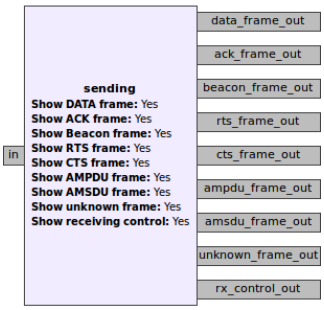
It is worth noting that, except the **frame_buffer** block, some other blocks may also store frames. For example, the **frame_aggregation** block stores the generated subframes until the number of frames for aggregation are reached. All timeout related blocks that store the frame waiting for acknowledgment also save at least one frame.

5.1.3 Interface blocks

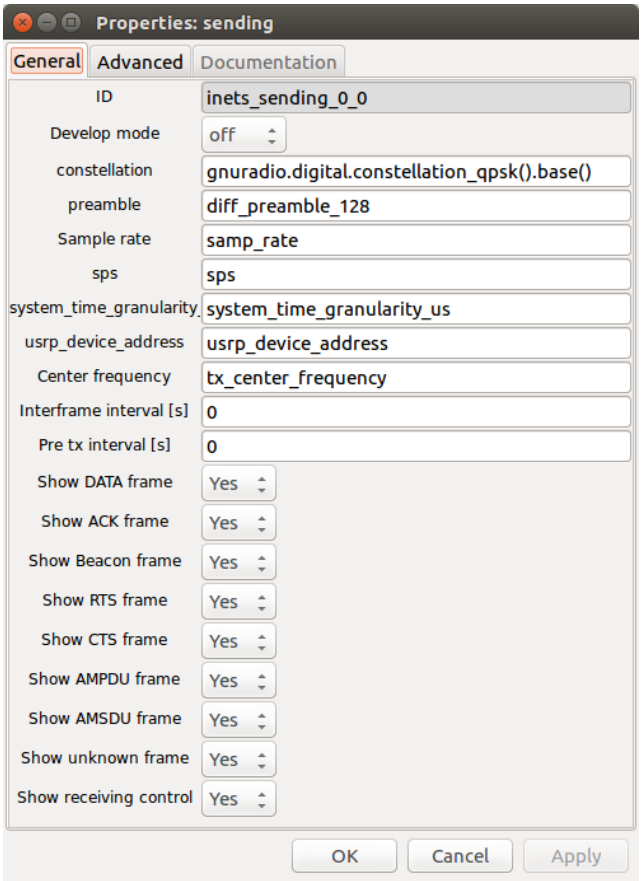
The interface blocks defined in DMDL are also implemented in GNU Radio. The **sending** block and the **receiving** block are implemented as hierarchical blocks which contain the complete PHY block chain which is an updated version of the PHY developed in [49] with timing control. The **carrier sensing** block has no PHY components, thus it relies on the information fed by the **receiving** block.

5.1.3.1 The sending block

The functionality of sending a frame in GR-DMDL is done by **sending** block. Figure 5.15 shows the complete form of **sending** block and its properties, respectively. Users are able to customize ports according to the protocol and usage. For example, as shown in in the classic CSMA/CA protocol with two way handshake, only the *data_frame_out* and *ack_frame_out* port are needed. **sending** block is a hierarchical block, thus its design can also be viewed in GRC. The **sending** block sends frames and indicate the remaining part of the program when it finishes transmitting. As shown in Figure 5.15a, it has two input message ports and nine output message ports. Frames which are supposed to be transmitted are passed to the *data_in* input message port. As shown in Figure 5.17, the received *commands* are separated after **decouple_frame_info** block. The input *command* actually are divided into two parts, i.e., the pure frame bit sequence which is sent to the **packetizer** block for the whole PHY processing, and the other frame information is passed to the **pending_tx_finish** block which is shown in Figure 5.16b. The frame information waits in the **pending_tx_finish** block until both the PHY processing



(a) Complete form of the **sending** block.

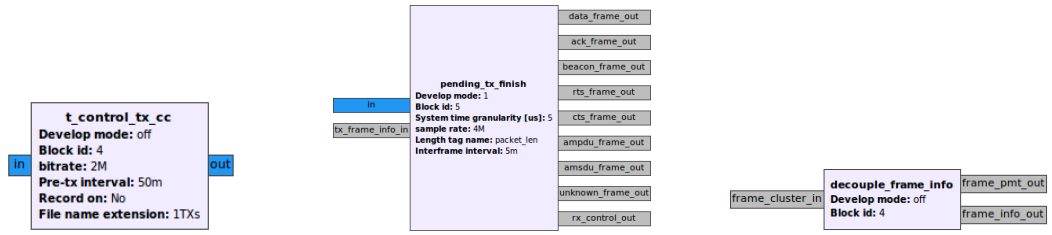


(b) Properties of the **sending** block.

Figure 5.15: The GUI of **sending** block in GRC.

and USRP transmission finish.

The separated frame bit sequence is passed to the **packetizer** block to form the PHY layer packet. Please refer to [49] for the whole PHY layer design and performance evaluation. In the end of the PHY processing, the generated I/Q sample stream is passed to **t_control_tx_cc** block to add the time stamp for transmission and measure the actual start time of transmission. Then the transmitted I/Q samples are passed to two blocks. The I/Q samples are sent out at **UHD: USRP sink** block which is the interface to USRP hardware. And the I/Q samples sent to **pend_tx_finish** block trigger the countdown of transmitting time. After counting down the transmission time, **pend_tx_finish** block generates the new *command* and output it according to the transmitted data type.



(a) t_control_tx_cc **(b) pending_tx_finish** **(c) decouple_frame_info**

Figure 5.16: sub-blocks in **sending** block

Holding the generated *command* before finishing sending the whole frame is important because it also holds the state transition of the state machine at the top level before the transmission ends. Otherwise, the following state or action may be triggered before the transmitting and receiving PHY/hardware are ready. However, the real transmission time, i.e., the time between the reception of the first bit of the transmitting I/Q sample stream at **UHD: USRP sink** block and the disappearance of the detectable energy in the medium is impossible to calculate. It is due to the fact that after passing the I/Q sample stream to **UHD: USRP sink** block, GNU Radio loses the control of the timing, i.e., and hardware related delay in USRP is unknown. To understand the real hardware performance, several measurements were designed and carried out. Please refer to Subsection 5.3.3 for the detailed measurement results. In **sending** block, to guarantee the *command* is passed after the radio front end is ceased, two timing parameters are inserted in sub-blocks of **sending** block. *pre-tx interval* is added to **t_control_tx_cc** block to enable user to assign an added delay before transmitting a frame. *Interframe_interval* is added to **pending_tx_finish** block to enable user to hold in the **sending** state with additive time. This way, the user may configure these two parameters according to the measurement results in their USRP and PC setup to ensure that the **sending** state holds the whole system duration the overall transmission time.

In order to enable on the fly reconfiguration on the USRP hardware, e.g., the center frequency and gain, a message port which is used for changing the USRP hardware settings is implemented and connected to **UHD: USRP sink** block. All *commands* follow the rule of the UHD Interface [36] can be passed to reconfigure the USRP at

any time.

5.1.3.2 The receiving block

The **receiving** block receives signals over the air and outputs the raw bit stream for later analysis. Received raw bits can be used by other blocks from the *rx_frame_out* output Message port. As shown in Figure 5.19, **receiving** block is also able to output the received signal strength which is normally used for carrier sensing from the *rx_power_out* output message port, the SNR of received signal from *snr_out* output message port, and the transmitting-receiving switch control signal which is explained in detail in Subsection 5.3.2.

Two input message ports are implemented in **receiving** block. The *rx_switch_in* works together with the corresponding output message port in **sending** block to avoid self receiving which is explained in detail in Subsection 5.3.2. Similar to the **sending** block, the on-the-fly reconfigurability of the USRP hardware is also supported by the **receiving** block. The *reconfig_in* input message port is developed for changing the USRP hardware settings in the run time. All *commands* follow the rule of the UHD Interface can be passed to reconfigure the USRP at any time.

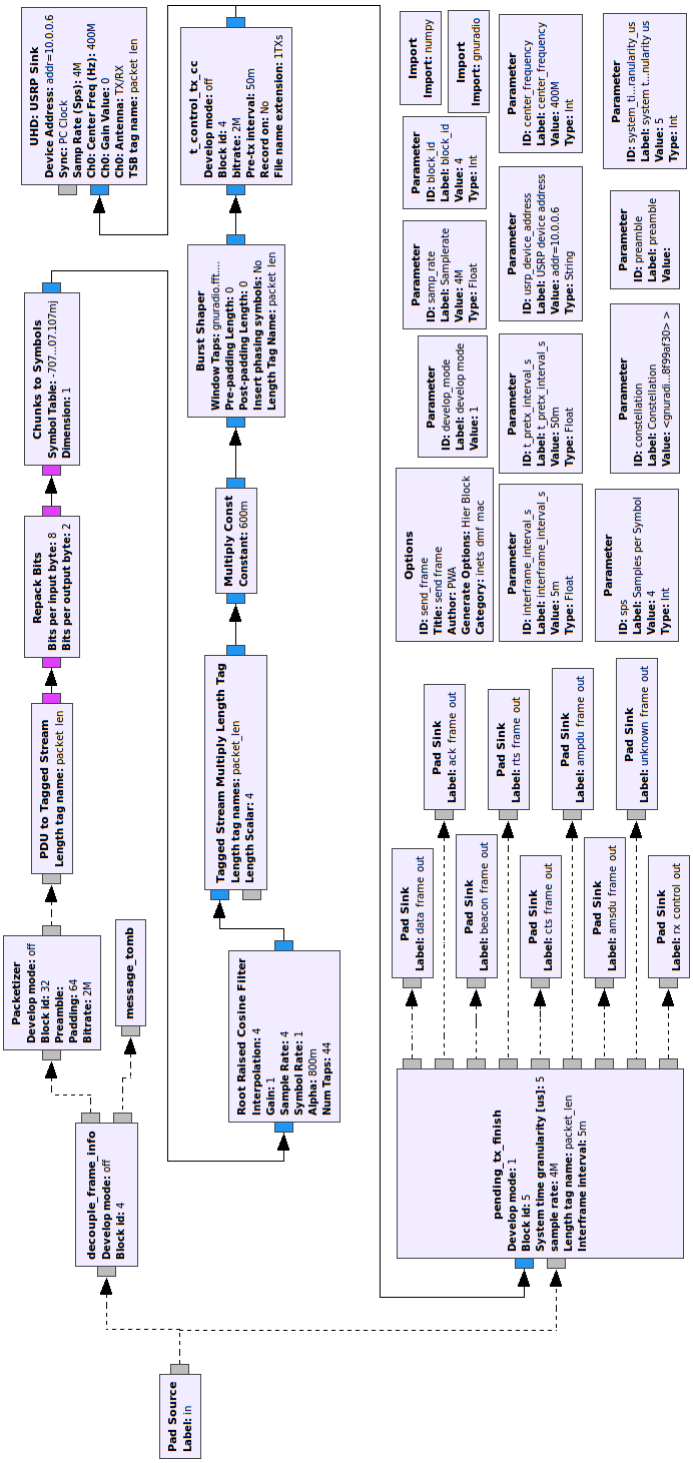


Figure 5.17: Details of the **sending** block.

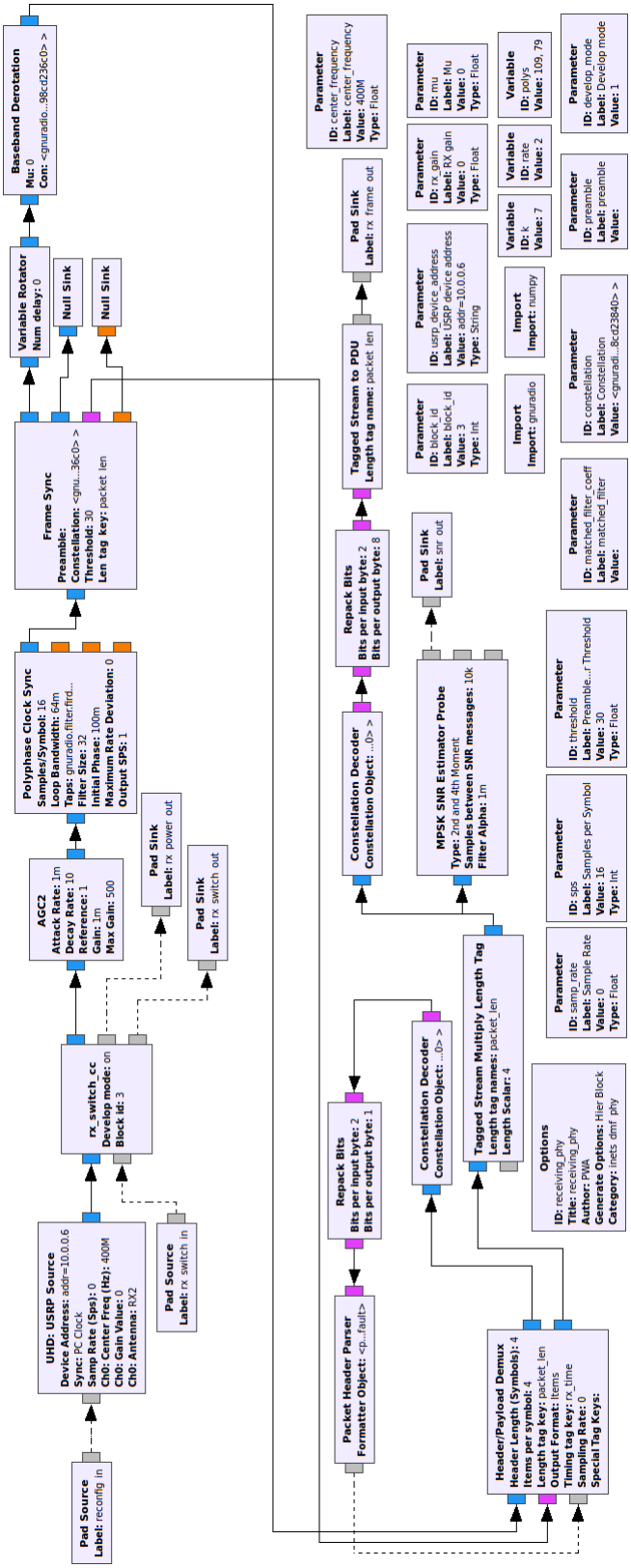
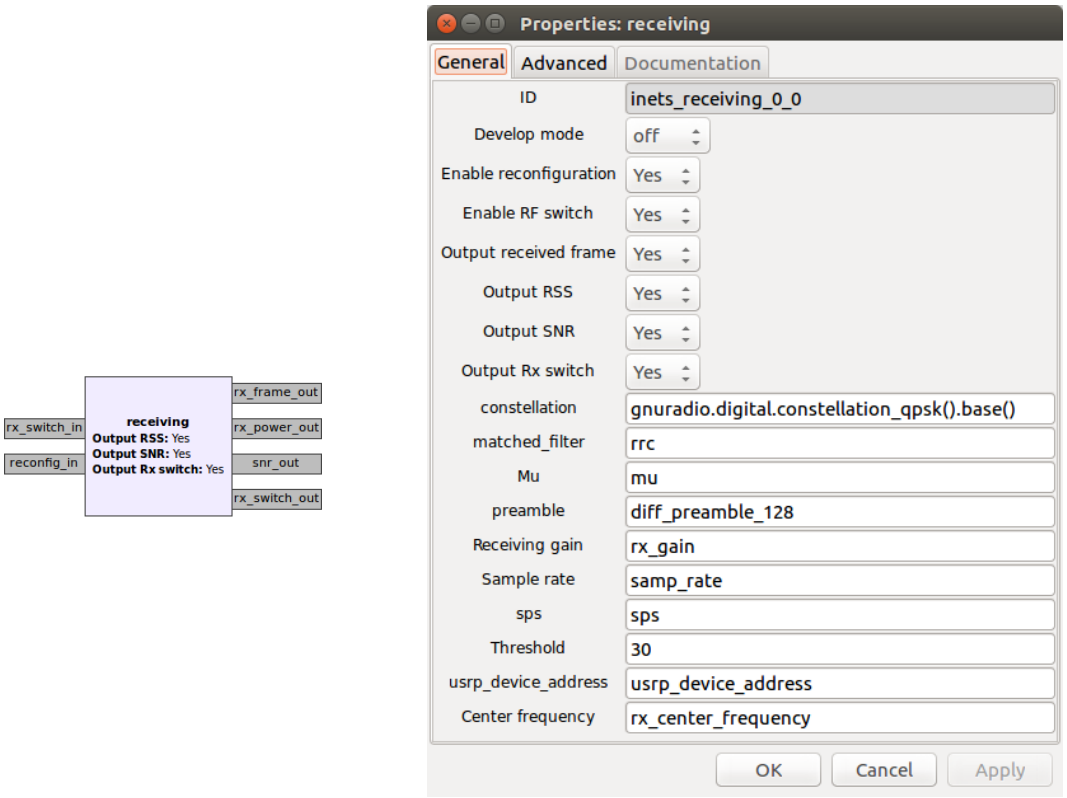


Figure 5.18: Details of the **receiving** block.

The **sending** block and **receiving** block are implemented based on the GNU Radio hierarchical block, i.e., they are purely graphical represented blocks. However, the PHY layer components cannot be changed and reordered as MAC blocks due to the fact that the GR-DMDL MAC is designed as an executable state machine but the PHY is a data-flow in which all blocks are fixed into their own positions for dedicated purposes. In addition, appropriate MAC transmission and reception rely on the accurate timing control which is designed and implemented based on the empirical results. Any modification on the PHY, i.e., inside the **sending** block and the **receiving** block, especially the timing control part may lead to significant performance degradation or malfunctions in MAC and even the systematic level. Therefore, users should be very careful in changing the **sending** block and the **receiving** block. Moreover, in order to achieve a dynamic GUI design for hierarchical blocks, the **sending** block and the **receiving** block are actually based on the **send_frame** block and **receiving_phy** block. After downloading the code, users should always generate the python code of **send_frame** block and **receiving_phy** block so that other blocks may connect to them.



(a) The **receiving** block. **(b)** Properties of the **receiving** block.

Figure 5.19: The GUI of **receiving** block in GRC.

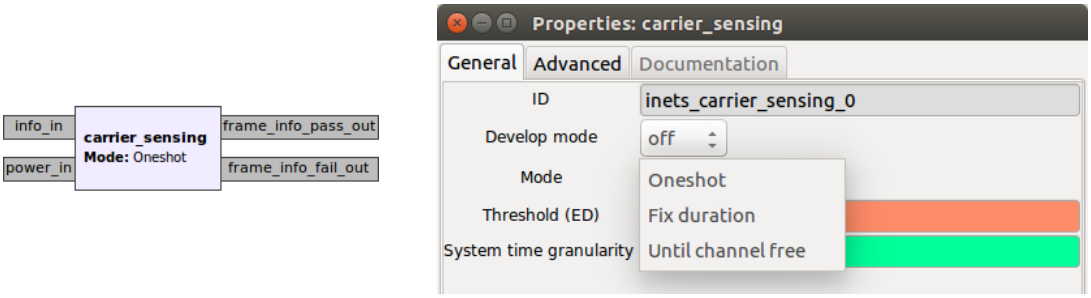
5.1.3.3 The carrier_sensing block

Carrier sensing is used in MAC layer for nodes to check the channel status. By comparing the received signal strength to the noise floor, carrier sensing enables the nodes to be aware of the real-time channel status. There are many carrier sensing technologies exists [67]. In order to achieve a fast carrier sensing, GR-DMDL adopts the energy detection as the default carrier sensing method. As shown in Figure 5.45, the **carrier_sensing** block gets the received power signal from the **rx_switch_in** block which is a sub-block of **receiving** block. The *rx_power_out* port continuously exports the received energy level at the maximum rate that the system can provide which is around 1 ms level. Depends on the purposes, nodes may sense the channel differently in time domain. As shown in Figure 5.20b, three types of carrier sensing are implemented. The *Oneshot* carrier sensing only compares the received energy level with the threshold once. If the received energy level is lower than the threshold, the channel is detected as idle. The *Fix duration* option keeps the **carrier_sensing** block listening to the channel for a fixed period. If the channel is detected as busy at any time during the carrier sensing duration, the channel is marked busy and the carrier sensing ends immediately. The *Until channel free* mode forces the **carrier_sensing** block to keep on sensing the channel until the channel back to idle if channel is detected busy at the beginning.

According to the carrier sensing results, the **carrier_sensing** block exports the input *command* which normally carries a MAC frame from different ports, namely *frame_info_pass_out* if the channel is detected idle, and *frame_info_fail_out* if the channel is detected busy.

5.1.3.4 Virtual Carrier Sensing

Virtual carrier sensing is one very specific MAC scheme used almost only in the IEEE 802.11 standards. However, due to the popularity of the IEEE 802.11 standard, we also implemented it in GR-DMDL. Unlike the energy detection based carrier sensing introduced above, the virtual carrier sensing purely relies on shared information by exchanging management frames, i.e., the RTS frame and the CTS frame. As



(a) The **carrier_sensing** block. **(b)** Properties of the **carrier_sensing** block.

Figure 5.20: The GUI of **carrier_sensing** block in GRC.

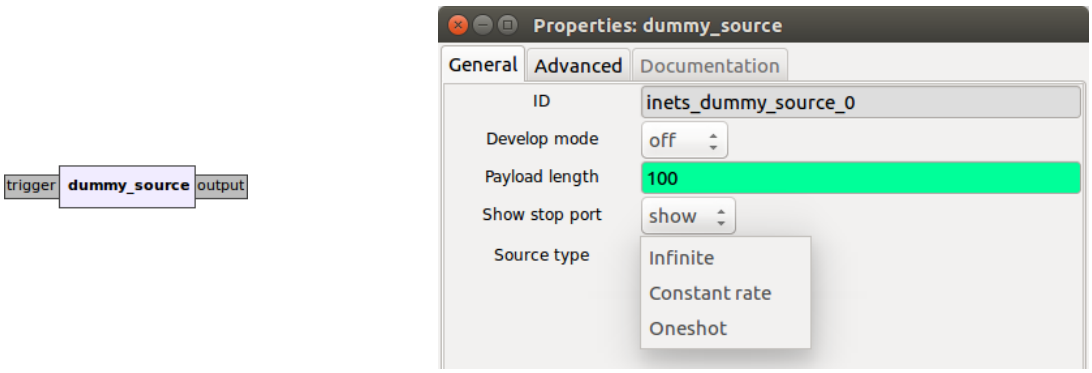
shown in Figure 5.4, the **backoff** block has an option to switch on its virtual carrier sensing functions by enabling the port for *command* containing RTS/CTS frames. After getting the timing information from the received frames, a countdown timer is activated. After the expiration of the countdown timer, the backoff is still to backoff, if it was in backoff before receiving the RTS/CTS frame, or stop by exporting the *command* so that the other part of the MAC is able to execute.

5.1.4 Auxiliary blocks

According to the definition in GR-DMDL, auxiliary blocks are also developed in GNU Radio to collect the empirical data and generate different types of traffic.

5.1.4.1 The dummy_source block

GNU Radio offers several sources to provide DATA from upper layer. For example, the User Datagram Protocol (UDP) source class [35] offers the interface to receive the UDP traffic in the host PC as a source. The corresponding UDP Sink class enables the GNU Radio to upload UDP packet to the host PC. By using these two, GR-DMDL is able to carry out real time video streaming.

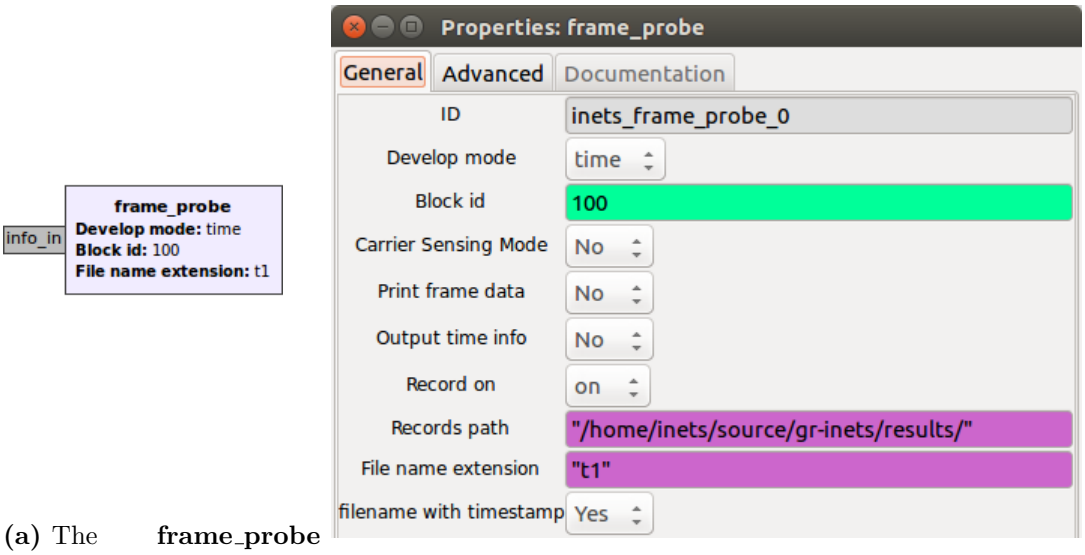


(a) The **dummy_source** block. **(b)** Properties of the **dummy_source** block.

Figure 5.21: The GUI of **dummy_source** block in GRC.

In order to test more types of traffics and enable testing theoretical models in GR-DMDL, we developed the **dummy_source** block. As shown in Figure 5.21, the **dummy_source** block starts generating random bytes to form the dummy payload with customized length. Several generating methods are realized in the **dummy_source** block which can be selected by users as shown in Figure 5.21. The infinite dummy source generates payload as fast as it can until receiving a *command* from its stop port. The constant rate dummy source generates payload with the given frequency. An instance of the **dummy_source** block in the Oneshot mode only generates one payload with one input *command* from its *trigger* port. Combining this mode with the **general_timer** block can generate different types of traffic.

5.1.4.2 Recording and Monitoring



(a) The **frame_probe** block.

(b) Properties of the **frame_probe** block.

Figure 5.22: The GUI of **frame_probe** block in GRC.

To enable users to understand the details of the execution inside blocks, three levels of develop modes, namely *on*, *off*, and *time*, are embedded in all blocks. If the property *develop_mode* is *off*, no debugging information is displayed in the Terminal at the bottom of the GNU Radio window. This is also the normal working mode which hides the execution details so that users may be able to focus on other more important information. If the property *develop_mode* is selected as *on*, the block will show the frame information such as the received frame index in the **frame_index_check** block and the actual timeout duration when ACK frames are coming on time. The third option, the *time* shows the key time points inside a block, especially a timing consuming block, which cannot be measured from the outside.

The original **message_debug** can only display very few types of message information, thus new blocks are designed and implemented. To have a closer look at the protocol running and record the data from the experiments, several auxiliary blocks are designed for recording the experiments results and monitoring the protocol execution in real time. Inspired by the name of the **Probe** tool in LabVIEW [75], the **frame_probe** block is designed for showing the intermediate values of GR-DMDL *commands* and other PMT messages as shown in Figure 5.22. This block has a log system which can save all recorded data into a text file for later use. It is worth noting that in order to ensure the appropriate display of all *commands*, newly added *commands* should also be included in the **frame_probe** block.

To record the number of frames at a specific location of the flowchart, we design the **counter** block. It simply records the message received from its input port and

display the records in the run-time at the terminal of GNU Radio. A protocol may deploy multiple **counter** blocks as shown in Figure 5.36b. In this example, **counter 8**, **counter 9**, **counter 10**, **counter 11**, **counter 12**, and **counter 13** record the number of received energy signal blocks, the number of frames which at least have a complete MAC frame header, the number of frames which pass the CRC check, the frames which have a complete MAC frame header but fail in the CRC check, the number of correctly received destined frames, and the number of correctly received frame with other destination address. This way, users are able to check any intermediate values of the running protocols. In addition, it also enables fast and straightforward debugging and optimization.

5.2 Signaling System

According to the *command* defined in GR-DMDL, we developed a message passing based signaling system in GNU Radio to connect blocks and convey information between blocks. The controlling signal is implemented by the dictionary container [34] in GNU Radio and passed among blocks by the message passing system [33]. In GR-DMDL, the controlling signal is named as *command*, which carries all necessary information related to the state transition. For example, the **framing** block generates the *command* in the transmitting side because it is the block which encapsulates the upper layer data unit into MAC frame. The *command* generated in **framing** contains the following fields,

- ▷ *frame_type* field, records the type of the generated frame.
- ▷ *frame_index* field, records the frame index of the generated frame.
- ▷ *destination_address* field, records the destination address of the generated frame.
- ▷ *source_address* field, records the source address of the generated frame.
- ▷ *num_transmission* field, records the number of transmissions of the generated frame, which may be increased when retransmission takes place.
- ▷ *reserved_field_I* field, records the information of the first received field of the generated frame.
- ▷ *reserved_field_II* field, records the information of the second received field of the generated frame.
- ▷ *payload_length* field, records the length of the payload of the generated frame.
- ▷ *header_length* field, records the length of the frame header of the generated frame.

It is worth noting that these fields in the *command* are quite similar to the fields introduced in Subsection 5.1.2.1. However, they are not the same. First, all the fields in the MAC frame header are byte or byte sequence which is about to be further processed by the PHY layer, while the field in *command* are just information which is easy to be read and handle. Second, all the MAC header fields are sent to the destined nodes and used there, while the fields in *command* are used locally. For example, as shown in the classic CSMA in Figure 5.45, the *command* with the generated frame is passed to the **time_out** block after being sent out and wait there until the expiration of the timeout timer. If the sent DATA frame is successfully acknowledged, the corresponding ACK frame is also passed to the **timeout** block. In the block, the field *frame_index* of the *command* containing the waiting DATA frame and the same field in the *command* containing the received ACK frame are compared. If the result shows that the DATA frame is successfully acknowledged, then the ACK frame is exported via the output port of the **block** for the future processing. Otherwise the *command* containing the DATA frame is sent out.

There are other transmitting related blocks that also generate *command* as the **framing** block does. The **rts_framing** block generates the *command* according to the information of the input *command* which contains the DATA frame, and the **cts_framing** block almost generates the same fields according to the received information from the RTS frame.

Another important block which generates *command* is the **frame_analysis** block because it is the block which decode the received MAC frame header. The fields which are added in the *command* generated by the **frame_analysis** include:

- ▷ *frame_type* field, records the type of the received frame.
- ▷ *frame_index* field, records the frame index of the received frame.
- ▷ *destination_address* field, records the destination address of the received frame.
- ▷ *source_address* field, records the source address of the received frame.
- ▷ *num_transmission* field, records the number of transmissions of the received frame.
- ▷ *reserved_field_I* field, records the information of the first received field of the received frame.
- ▷ *reserved_field_II* field, records the information of the second received field of the received frame.
- ▷ *payload_length* field, records the length of the payload of the received frame.
- ▷ *header_length* field, records the length of the frame header of the received frame.
- ▷ *address_check* field, create for recording the results of address check of the received frame. The field is set to 0 as default before being passed to the **address_check** block.

- ▷ *good_frame* field, records the frame index of the generated frame. The field is set to 0 as default before being passed to the **frame_check** block.

Thanks to the flexibility of the PMT dictionary container, a *command* can contain an arbitrary number of fields according to the blocks it passes, i.e., all blocks are able to generate, update and consume fields according to its own functionalities and demands. For example, the **frame_analysis** block generates a special field named *nav_time* if the coming frame is an RTS frame or a CTS frame. This field is used by the **bakcoff** block to wait.

After receiving a *command*, blocks make decisions or perform data processing according to the information in the received *command*. blocks may also create and update the fields in the *command* and forward it to the next stage.

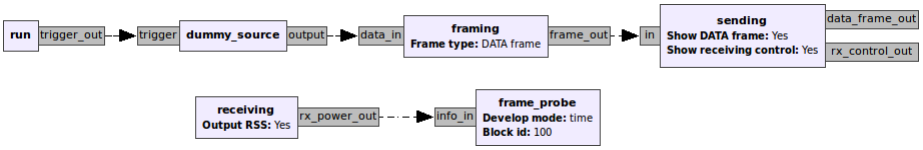
5.3 Performance Evaluation on Single Frame Transmission

Timing performance is one of the most important benchmark to evaluate the implementation of MAC protocols. Unlike the theoretical models and pure software based simulations, hardware developers have to address practical problems and physical limitations. For example, in the theoretical studies [30], frames are sent over the medium immediately after the protocol decides to transmit it. However, in hardware platforms, there is always an overhead when states are transited and transmission is carried out. Delays also exist in the receiving side due to the speed limitation on sensors and internal signal transmission. Moreover, due to the limited computational capacity, the processing and scheduling at processors may also cause unexpected and unstable delay especially when the computational load is heavy. These practical issues definitely bring the discrepancies between analytical expectations and measured data. Although the hardware based prototyping intuitively addresses the practical problems which make it more real, the system with smaller overhead and discrepancies are still considered as a better platform. Some overheads and limitations are platform dependent, i.e., the problem may only take place in one platform but not others. Therefore, the measurements carried out from a platform with low hardware dependent overhead can be considered more reliable. Besides, the results taken from the platforms with less discrepancies are more understandable. Moreover, it is also convenient for users to fast develop their protocols and the scheme because users may focus on the core implementation but not the compatibility or other platform related issues.

Therefore, two main targets should be addressed in developing a MAC framework on SDR platforms. First, it should be effective and efficient. Black box like unstable components should be reduced as much as possible. Second, all the hardware caused overhead should be measured and analyzed so both the theoretical protocol designer and the practical developer are able to fully understand the system and its outcomes.

We carried out a number measurements to investigate the performance of GR-DMDL with USRP2 devices in terms of the mentioned targets. The testbed includes up to 16 USRP2 devices and PCs with 16 GB memory and Intel i7 3770 CPU. The results are demonstrated in this section as the baseline for future usage. We first carried out measurements at the frame level, in which we focus on the signal frame propagation characteristics and the timing specifications inside a node and between nodes. The energy strength of received MAC frames at the receiving port of the same USRP2 and at other USRP2 devices are measured and compared, where we observed a strong tail effect from the self-received signal. The tail effect and the corresponding impacts on protocol design are illustrated and discussed in Subsection 5.3.1 in detail. In order to avoid the tail effect, a Transmission and Reception Switch (TRS) system is developed and evaluated in Subsection 5.3.2. All critical time points in transmitting a single DATA frame are measured in Subsection 5.3.3. All time points in a two way handshake are measured and analyzed in Subsection 5.3.4. The results are used for determine the MAC layer parameters, such as the interframe intervals or the system time unit which are used in the next section where MAC protocols are implemented and measured.

5.3.1 Frame Waveform



(a) The GRC code of the transmitter in frame waveform measurement.



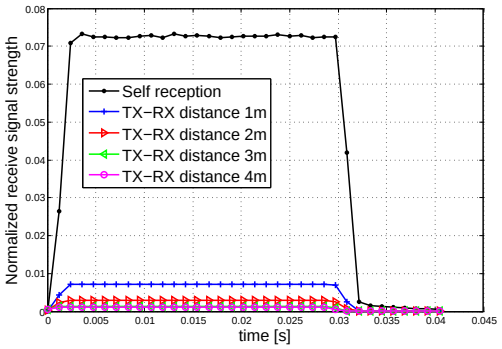
(b) The GRC code of the receiver in frame waveform measurement.

Figure 5.23: The GRC code used in frame waveform measurement.

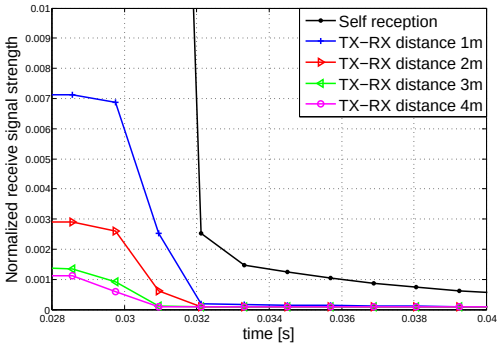
To start using GR-DMDL to design and implement MAC protocols, the transmission of a single frame should be measured and studied. The GRC code used in the measurement is shown in Figure 5.23. In this measurement, the waveform of frames over the air with different distances are plotted. Due to the full duplex feature of USRP, simultaneous transmission and reception are enabled. Therefore, the self-received frame is also plotted as the baseline. As shown in Figure 5.23b, the *rx_power_out* port of **receiving** block is activated and a **frame_probe** block is connected to it to record the measurement data. In the measurement, the distance between the transmitter and the receiver are changed to get RSSI values with distances. In order to test the self-receiving feature, the RSSI of the self-received signal is also recorded as shown in the lower half of Figure 5.23a. The measured frame waveforms are shown in Figure 5.24. In the measurement, the frame length is fixed to 700 bytes with 15 bytes of MAC header and 147 bytes of PHY overhead. QPSK modulation is used, and the sample per symbol is 4. The sampling rate is

fixed to 400 KHz which is the double of the minimal supported sampling rate of USRPs.

It is obvious that the self-received frame has the highest signal strength. It is due to the fact that the distances between the two antenna ports of one USRP is about 2 cm which is much shorter than the normal distance between any two USRP devices. Moreover, as shown in Figure 5.24b, the RSS of self-received signal still remains about at a certain high level after the end of the frame transmission, namely tail effect. Even though different USRPs show different RSS characteristics, it is observed on all USRPs in the measurements. The tail effect has some negative effects on the protocol design and execution. It may lead to false positive error in carrier sensing involved MAC protocols after sending a frame. At the reception side, the remaining energy may interfere the coming frame from other nodes. In order to eliminate the tail effect, several algorithms are deployed in GR-DMDL. In order to avoid the self-interference, a control scheme to coordinate transmission and reception blocks are applied. It is explained in detail in Subsection 5.3.2. Besides, a time interval is inserted after transmissions in **sending** block which is shown in Subsection 5.1.3.1. With the help of this practical improvement, GR-DMDL is able to achieve error free reception and energy detection based carrier sensing.

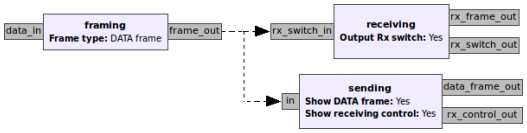


(a) Waveform of GR-DMDL frame transmission with different receiving distance.

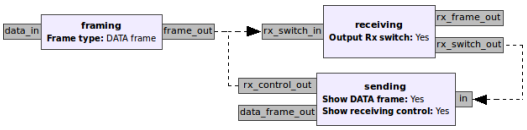


(b) Tail effect of self received signal. Zoom in from the waveform figure.

Figure 5.24: Waveforms of GR-DMDL frame transmission. Frame length is 700 bytes. Modulation is QPSK and Sample per symbol is 4. Sampling rate is 400 KHz.



(a) Simple TX-RX control mechanism.



(b) TX-RX switch system.

Figure 5.25: Mechanisms for avoiding self receiving.

5.3.2 Transmission and Reception Switch System

Due to the size limits and costs, many mobile devices are only equipped with half-duplex radio. Therefore, classic MAC protocols such as CSMA and TDMA are also targeted for half-duplex system. In contrast, most of the widely used SDR platforms are able to perform simultaneous transmission and reception. In some platforms such as USRPs, staying in the full duplex mode enables fast switching between transmitting and receiving modes. Therefore, an SDR platform may receive frames sent by itself. Self-receiving may cause severe problems to both the hardware and the logic of protocols. For instance, it may trigger carrier sensing or other energy detection functions which may drive the state machine of the system into wrong state.

In order to avoid self-receiving, the **sending** and **receiving** blocks are needed to be coordinated. A simple design of **sending/receiving** coordination is shown in Figure 5.25a which sends independent control messages to both the Sending and Receiving blocks. In order to guarantee no self-receiving, the system must own the knowledge of the transmission time of the sent frame and also the processing delay of the **Sending** block before it performs baseband processing. However, due to the instability of the GNU Radio implementation which is based on the operating system scheduling, all timing parameters may vary with the configuration of the PC, the version of the operating system, and even the other running tasks in the same PC except GNU Radio. Therefore, a Transmission/Reception Switch (TRS) system is developed to avoid such synchronization problem. As shown in Figure 5.25b, the transmitting frame is passed to **receiving** block before passed to **sending** block to ensure that the reception is disabled. After the transmission, **sending** block passes a message to **receiving** block to switch on reception. This way, concurrent transmission and reception are avoided.

In **receiving** block, the **rx_switch_cc** block is designed as the first block of the whole receiving block chain for controlling the ON/OFF of the remaining receiving functionalities (cf. Figure 5.18). If a *command* with a valid frame sequence is received, it stops passing received I/Q sample stream to the receiving PHY blocks which literally switches off the receiver in the digital domain. When a controlling message for switching on is received, it switches back to the receiving mode.

It is worth noting that the TRS system is not an essential part of GR-DMDL protocol design. For example, in all handshake based 802.11 liked protocols, the carrier sensing also takes place before the transmission and there are always waiting blocks after transmission, therefore the tail effect can never disturb the carrier sensing and transmitting functionalities. Users may choose to use it or not according to the transmission schemes and even the experimental results.

5.3.3 Timing in Single Frame Transmission

In order to achieve effective and accurate MAC protocol design and implementation, we measured the timing characteristics of sending frames. In these measurements,

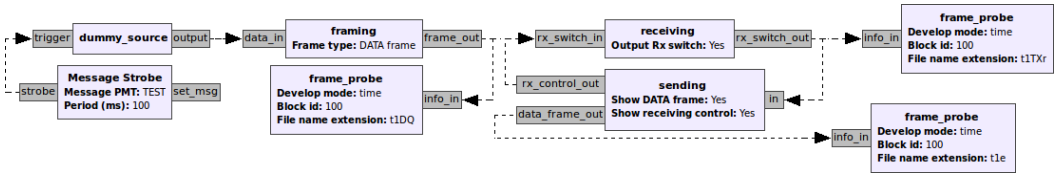
the transmitting node continuously sends frames to the receiving node with a fixed frequency as described in Figure 5.26a and Figure 5.26b, respectively. In this scheme, the **framing** block generates DATA frame, and the **sending** block performs all PHY layer signal processing and send the frame over the air using USRP. As introduced in Subsection 5.3.2, the **receiving** block is also involved in the whole transmission procedure. Connecting all mentioned blocks, a single frame transmission test is able to be carried out to test the timing performance of transmission, reception and carrier sensing using the schemes shown in Figure 5.26a and its corresponding receiving scheme shown in Figure 5.26b. Some of the key parameters used in the measurements are listed in Table 5.1.

Parameters	Value
Frame generation speed	5/s
Sampling rate	400 k/s
t_{switch}	4 ms
Normalized carrier sensing threshold η	0.003

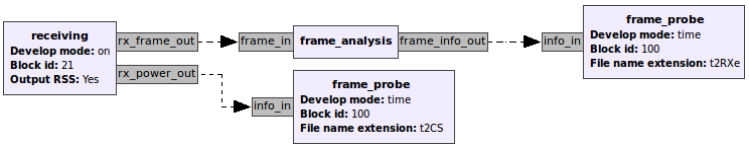
Table 5.1: Key parameters used in single frame transmission measurements.

All key time points take place in a single frame transmission from node 1 to node 2 are marked and shown in Figure 5.27. The frame is decided to be transmitted at $t_{1\mathrm{DQ}}$ at the transmitter, which is normally the time point that the frame is dequeued from the transmission frame buffer, or in general the time point which the frame leaves the previous block before the **sending**. [2] Due to the TRS system deployed in the GRC implementation, $t_{1\mathrm{TXr}}$ and $t_{1\mathrm{TXs}}$ are the time points when the receiver is switched off and the frame starts being sent to USRP hardware, respectively. The frame finishes being sent at $t_{1\mathrm{TXe}}$ and the whole transmission procedure is delayed to $t_{1\mathrm{e}}$ to avoid self-receiving caused by the tail effect. On the receiver side, the carrier

[2] The numbers within the subscript of the time parameters shows which node it belongs to and the following letters refer to the key actions. For example, DQ mean dequeue, CSs means start carrier sensing, and REe represents receiving ends.



(a) Transmitter design for single frame transmission test.



(b) Receiver design for single frame transmission test.

Figure 5.26: Design and implementation of single frame transmission test in GRC.

sensing is first triggered at $t_{2\mathrm{CSs}}$, and ends at $t_{2\mathrm{CSe}}$. All these time points except $t_{1\mathrm{TXs}}$ are measured by the **frame_probe** blocks as shown in Figure 5.26. To achieve a more accurate measuring for $t_{1\mathrm{TXs}}$, it is measured by the **t_control_tx_cc** block which is the last block in GNU Radio before passing the PHY packets to USRP. Due to the delay caused by the whole PHY layer processing and MAC layer processing at **frame_analysis** block, the end of the reception, i.e., the moment that other blocks are able to access the decoded information, is $t_{2\mathrm{REe}}$. In the measurements, the payload length is changed from 0 bytes to 1000 bytes. By carrying out at least 10000 times of transmissions for each payload length, the duration between the key time points are measured and illustrated in Figure 5.28, Figure 5.29, and Figure 5.30.

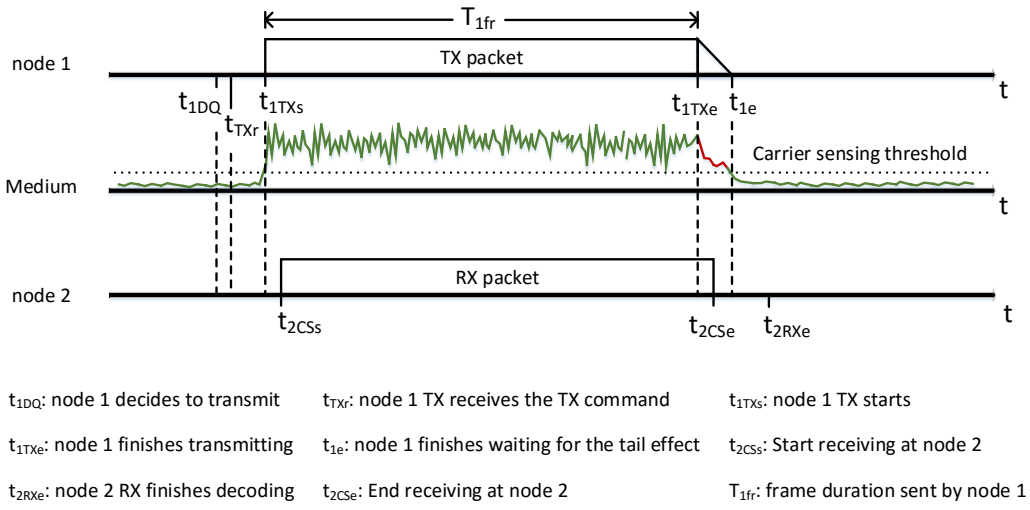


Figure 5.27: Key timing parameters in block level synchronization.

The timing characteristics related to the frame and the delay between the transmitter and the receiver are shown in Figure 5.28. In total six timing parameters are shown for comparison. In this figure, four results are quite similar. $T_{1\mathrm{DQ\text{-}1e}}$ is the overall time consumed in the transmission side. $T_{1\mathrm{TXs\text{-}1e}}$ is the duration between the starting point of real transmission and the end of the whole transmission. $T_{2\mathrm{CSs\text{-}2CSe}}$ records the channel busy status. Therefore, $T_{2\mathrm{CSs\text{-}2CSe}}$ can represent the channel conditions in all other nodes when one is transmitting. We note that $T_{2\mathrm{CSs\text{-}2CSe}}$ depends on the carrier sensing threshold. However, as shown in Figure 5.24, the channel occupation time has no significant difference for different carrier sensing thresholds in a large scale. T_{frame} is the theoretically calculated frame transmission time. With the parameters used in the selected platform, the transmission time of a frame with N bytes payload is,

$$T_{\mathrm{frame}} = \frac{8L(N + N_{\mathrm{MAC}} + N_{\mathrm{PHY}})/\log_2 M}{f_s}, \tag{5.1}$$

where N_{MAC} and N_{PHY} are the overhead in MAC and PHY layer. M is the number of distinct messages that could be sent in one symbol, and L is the number of samples

per symbol. In the end, f_s is the sample rate. According to the parameters set in Table 5.1 and the frame structure in Figure 5.5, N_{MAC} and N_{PHY} are 15 bytes and 148 bytes, M is 4, L is 4, and f_s is 400000 samples/s, respectively.

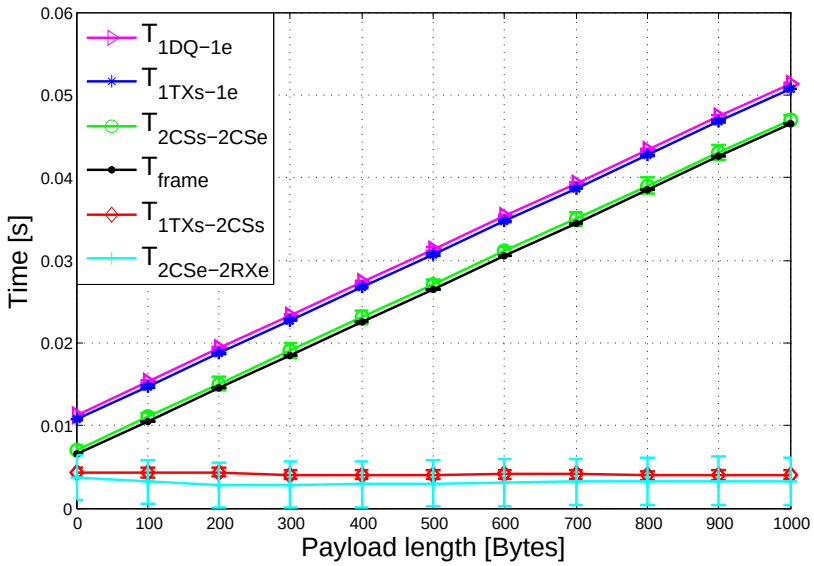


Figure 5.28: Frame level timing characteristics in single frame transmission measurements.

As shown in Figure 5.28, $T_{\text{1DQ-1e}}$, $T_{\text{1TXs-1e}}$, and $T_{\text{2CSs-2CSe}}$ grow linearly with the payload as T_{frame} which is the theoretical value. Their standard deviation is small, which shows the high stability of GR-DMDL. Moreover, the difference between the T_{frame} and $T_{\text{2CSs-2CSe}}$ is also very small, which shows that the actual reception is quite accurate. More on these two durations, $T_{\text{2CSs-2CSe}}$ is always slightly greater than T_{frame}. That is due to the fact that the transmitter cannot switch off energy emission immediately after finishing sending the packets, which is also shown in Figure 5.24. Nevertheless, the very small difference between them still enables accurate estimation on channel occupation time, which is widely used in MAC protocols, i.e., in virtual carrier sensing. The difference between $T_{\text{1DQ-1e}}/T_{\text{1TXs-1e}}$ and $T_{\text{2CSs-2CSe}}/T_{\text{frame}}$ caused by two major overheads, i.e., the delay between making the decision of transmitting to real transmission and the forced waiting time for TRS system. As shown in Figure 5.28, the difference is stable, which enables the protocol designers and developers to consider the gap as a constant in their protocols.

The two curves in the bottom show the delay in the receiving side. The first one, $T_{\text{1TXs-2CSs}}$, represents the delay between the transmitting node and receiving node. As shown in Figure 5.27, $T_{\text{1TXs-2CSs}}$ keeps constant with all payload lengths and has very small deviation. It is usually ignored in theory when analyzing the CSMA based protocols. However, due to the imperfection of the hardware implementation, that gap exists in the real hardware platform. It can only be alleviated but not eliminated. It is the additional overhead which may lower down the overall performance. More importantly, $T_{\text{1TXs-2CSs}}$ may raise the risk that in the duration one node may not able to detect the ongoing transmission and start transmitting its own frame. Users need

to address it in protocol design and implementation. The second one, $T_{\text{2CSe-2RXe}}$, only takes place in the receiving side. It is the duration between the time point that the signal disappears at the receiving energy detection function and the the time point that the frame first appears in MAC blocks, i.e., after decoding all MAC header information after the **frame_analysis** block. Due to the high complexity and load in digital processing, the cost time is around 4 ms with a high standard variation on a PC with 16 GB memory and Intel i7 3770 CPU. Deploying high performance single core processor may shorten the delay. However, due to the pipelining principle of GNU Radio, this delay would not impact the maximal receiving ability.

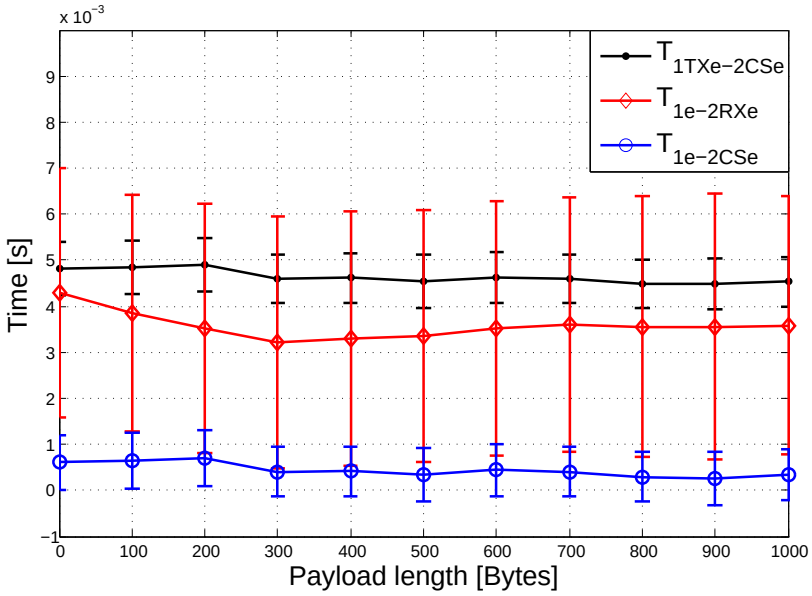


Figure 5.29: Delay between nodes.

Some delays between nodes are also measured and plotted in Figure 5.29. $T_{\text{1DQ-2CSs}}$ represents the duration from the time point that a frame is exported by the **framing** block until the time point that it is detected by other nodes. Like $T_{\text{1TXs-2CSs}}$, this duration is also not addressed in theoretical studies but it is very critical in practical implementation. In all CSMA based protocols, transmitting nodes are not able to cancel the transmission after t_{1DQ}. However, its signal energy starts being detectable in the medium after t_{2CSs}. Therefore, $T_{\text{1DQ-2CSs}}$ is the time window in which collision avoidance is not able to be performed. This effect should be addressed when comparing the analytical results and the experimental data. Similar to $T_{\text{1TXs-2CSs}}$, $T_{\text{1TXe-2CSe}}$ represents the delay of frame transmission and carrier sensing in other nodes. $T_{\text{1TXe-2CSe}}$ has the same mean value to $T_{\text{1TXs-2CSs}}$ but a greater standard deviation. That is due to the fact that receiving a frame requires much more processing resources than sending, which may cause various lengths of delay. The red curve represents the difference between the end of the transmitting node and the receiving node. For all frame lengths, the transmitting node finishes around 3 ms to 4 ms prior to the receiving node including the duration taken by the TRS system. One conclusion can be drawn from this result is that the receiving node

may start transmitting immediately after decoding the receiving frame information which is a normal transmission strategy for the handshake based protocols because the transmitting node is always finished earlier and ready for receiving. The last curve in Figure 5.28, $T_{\text{1e-2CSe}}$ represents the time difference between the end of transmission in the transmitting node and the end of the detectable energy signal in other receiving nodes. Its mean value for all payload lengths are lower than 1 ms.

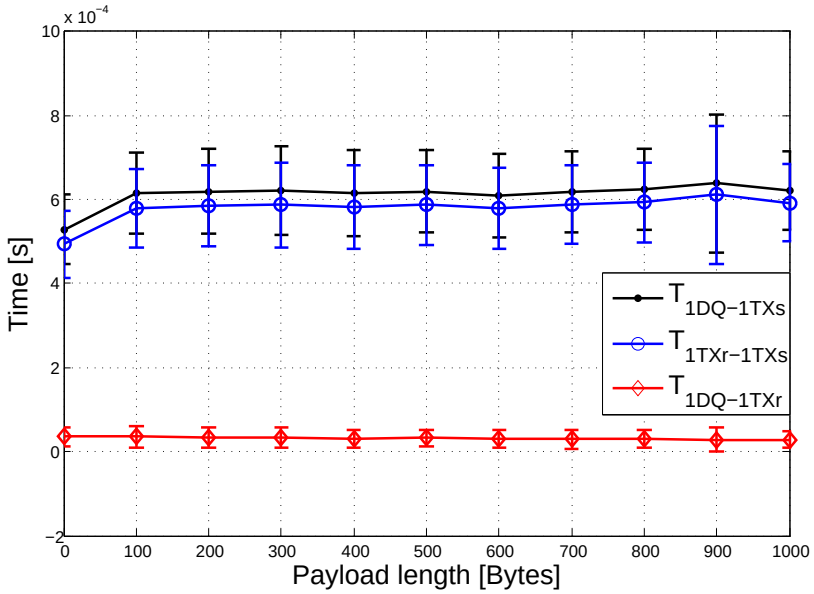


Figure 5.30: Delay between blocks in transmitting node.

In the end, the delays in the transmitting node are measured. In Figure 5.30, $T_{\text{1DQ-1TXs}}$ is one part of the $T_{\text{1DQ-2CSs}}$ at the transmitting side. $T_{\text{1TXr-1TXs}}$ represents the duration from when a frame is received by the **sending** block from the **framing** block until it is sent. Actually, this duration measures the PHY processing delay. It has only a small gap between $T_{\text{1DQ-1TXs}}$, which is $T_{\text{1DQ-1TXr}}$, the pure software overhead of passing messages from block to block. As shown in Figure 5.30, the pure software delay is around 100 us which is much smaller than the frame transmission time and other overhead. Therefore, the software processing and GNU Radio message passing are not the bottlenecks of the system.

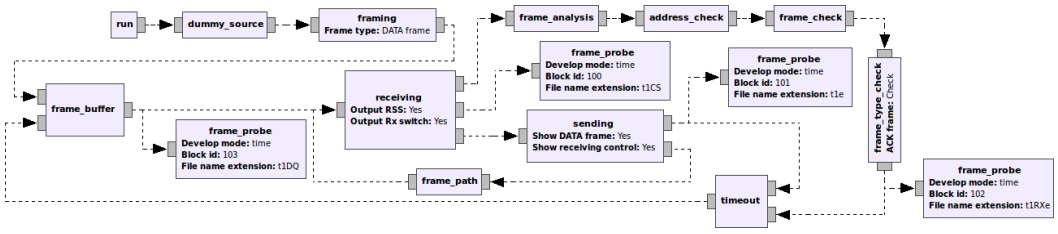
5.3.4 Timing in Handshake

After testing the timing performance of single frame transmission, the timing specifications of handshake is also tested. Handshake is the most widely used transmission scheme in MAC layer which performs fast exchanging of information between nodes, e.g., receiving nodes send an ACK frame after receiving a DATA frame in 802.11 liked protocols. The most important metric to evaluate the handshake transmission is the response time which is the time needed for the transmitter to wait until the response comes back from the destination node. It

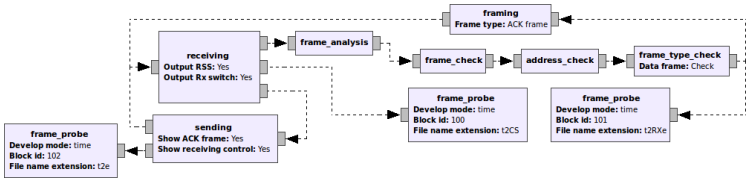
includes the propagation delay for the round trip and the processing delay at both nodes. Obviously, the shorter the response time, the better performance the system can achieve. In many theoretical studies, the response time is ignored due to its complexity in modeling, however, it should be addressed in all hardware based implementations. In practice, the response time includes many aspects from both the transmitting side and receiving side. For example, the receiving PHY processing delay at the receiving node between $t_{2\text{CSe}}$ and $t_{2\text{RXe}}$ in Figure 5.32, the transmitting PHY processing delay which lays between $t_{2\text{RXe}}$ and $t_{2\text{TXs}}$ at the receiving node, and the receiving PHY processing delay at the transmitting node from $t_{1\text{CSe}}$ and $t_{1\text{RXe}}$. In order to measure the timing performance in handshake, a testing protocol is designed in GR-DMDL as shown in Figure 5.31. Similar to the previous measurements, many **frame_probe** blocks are inserted to record the results. The results are shown in Figure 5.33 to Figure 5.35. In order to achieve statistically stable results, at least 10000 two way handshake communications are carried out.

Similarly to the results shown in Figure 5.28, frame transmission duration related timing performance are quite stable and linear with the lengths of the payload in the two-way handshake measurements. As shown in Figure 5.33, the duration $T_{1\text{TXs-1e}}$, and $T_{2\text{CSs-2CSe}}$ show a good linearity with $T_{1\text{fr}}$ which is the frame transmission time for all sizes of payload. At the receiving node, the frame length of the ACK frame does not change with the length of the received DATA frame, so both the $T_{2\text{TXs-2TXe}}$, and $T_{1\text{CSs-1CSe}}$ are constant and equal to the corresponding durations at the transmitting node with 0 payload length. It is due to the fact that the ACK frame shares the same frame structure with DATA frame but without payload.

Some results of the transmitting-receiving transition time are shown in Figure 5.34. All curves in this figure describes the durations take place at one node. $T_{1\text{e-1CSs}}$ shows the minimal gap at the transmitting node between finishing sending a DATA frame and starting being able to sensing the replied ACK frame. The length of this duration depends on the processing speed at the receiving node side and the



(a) Transmitter design for two way handshake test.



(b) Receiver design for two way handshake test.

Figure 5.31: Design and implementation of two way handshake test in GRC.

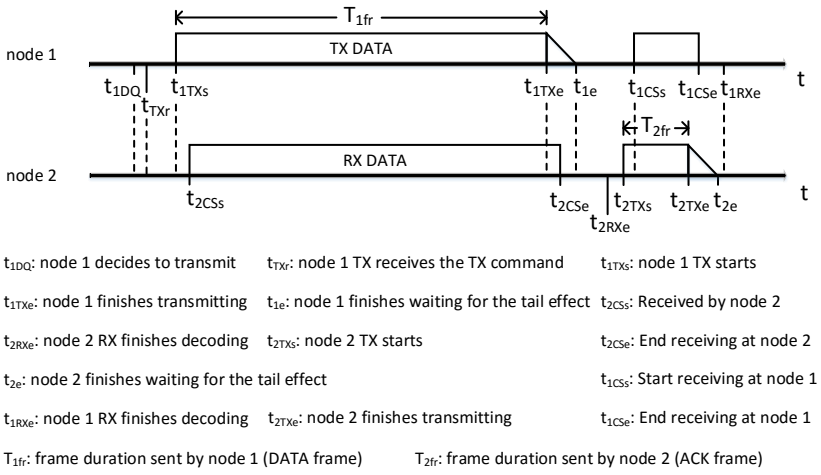


Figure 5.32: Key timing parameters in two way handshake.

length of the tail effect. Obviously, $T_{\text{1e-1CSs}}$ should be strictly kept positive to ensure the reception of the ACK frame. As marked by the blue curve in Figure 5.34, the reception of the ACK frame is guaranteed that $T_{\text{1e-1CSs}}$ is always greater than 0 for all payload lengths.

$T_{\text{1CSe-1RXe}}$ illustrated the receiving processing delay at the transmitting node for handling the ACK frame. In comparison to $T_{\text{2CSe-2RXe}}$, it is slightly greater but with the similar standard variation. Considering the transmission tasks before receiving, the exceeded time may caused by extra scheduling. Nevertheless, the processing delay is still not significant. In a continuously transmission system such as streaming, the delay can be compensated at MAC layer by forcing inserting an inter-frame

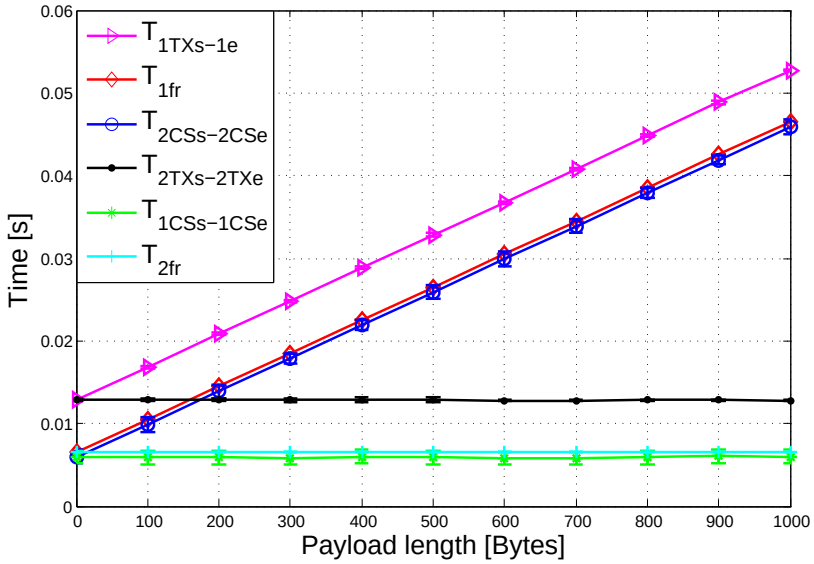


Figure 5.33: Frame level timing characteristics in two-way handshake measurements.

interval before next transmission, and in a handshake based system such as the 802.11 liked networks, nodes are not supposed to send immediately after finishing the last round handshake to ensure the fairness among nodes. Therefore, the negative impacts of $T_{\text{1CSe-1RXe}}$ and $T_{\text{2CSe-2RXe}}$ can be eliminated.

$T_{\text{2CSe-2RXe}}$ illustrates almost the same result as $T_{\text{2CSe-2RXe}}$ in the single frame transmission which shows that the performance of the receiving node keeps constant in spite of following transmission tasks. The last curve in Figure 5.34 shows that the switching time between finishing receiving and start transmitting is relatively small and not to be a negative factor of the system performance.

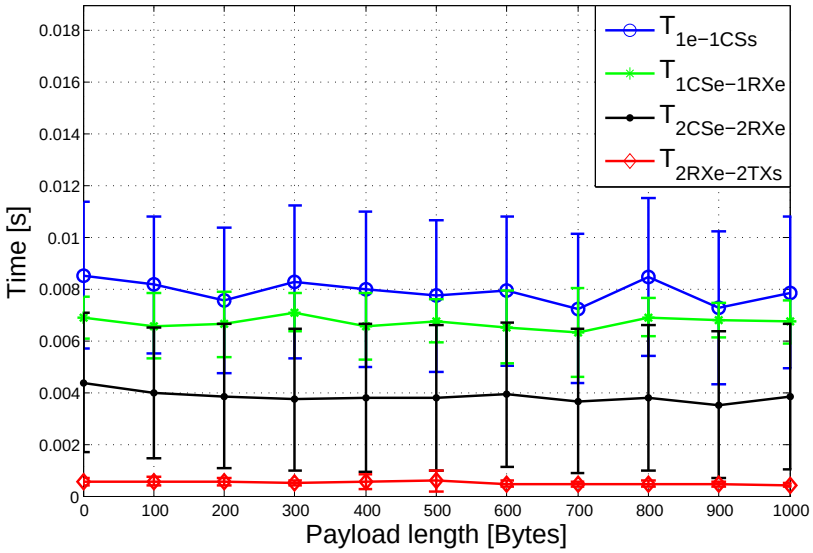


Figure 5.34: Delay between blocks in the same nodes.

The delays between different nodes are shown in Figure 5.35. $T_{\text{2CSe-1CSe}}$ shows the channel idle time between handshake. From the handshake nodes point of view, the delay represents the inefficiency of the handshake caused by processing delay. In handshake based systems, timeout is a frequently used functionality to check whether the acknowledgment is coming back on time. A longer timeout may let the system collect all sent acknowledgments, but may be inefficient in networks which have frame loss. In general, the timeout should be set according to the processing delay and the propagation delay. In indoor laboratory environments, in contrast to the distances caused propagation delay, the processing delay on board and host PC are dominating. Therefore, the timeout duration should be set according to it. From other nodes' point of view, $T_{\text{2CSe-1CSe}}$ is the duration in which there is no ongoing transmission but is also not idle. That is, all other nodes has to sense the channel longer than this delay to enable detecting the potential ACK frame to ensure that the channel is real idle.

$T_{\text{2TXs-1CSs}}$ shows the delay between the time point that the receiving node makes decision of sending the ACK frame and the time point that the transmitting node starts detecting energy signal on the channel. Its value and stability are similar to

$T_{\text{1TXs-2CSs}}$ in Figure 5.28. Combining with other results which show the linearity of GR-DMDL performance in the time domain, the timing performance of GR-DMDL realization on GNU Radio has proven to be both linear and stable. The results in this section can be further used in later protocol designs to accurately predict the performance and set the parameters.

$T_{\text{1CSe-2e}}$ records the delay between the disappearance of the energy signal on the channel and the logical ends of transmission on the receiving node. The curve shows that this delay is always positive, which guarantees that even a node is back-to-back transmitting frames, other nodes in its range is still able to receiving all of the frames because the receiving node is always ready before the transmitting node finishing the last round of transmission.

The last delay $T_{\text{2e-1RXe}}$ measured here is the time difference between the end time point of the whole procedure in transmitting node, t_{1RXe}, and the end time point of the whole procedure in receiving node, t_{2e}. As marked by the blue curve in Figure 5.35, the difference is very small. It shows that the GR-DMDL realization in GNU Radio is well tuned so that the transmitting node and the receiving node in handshake are able to finish in almost the same time for all payload lengths. This way, the transmitting node and the receiving node are able to be synchronized again after the transmission, thus the following timing operations like IFSs and backup can be performed correctly and precisely.

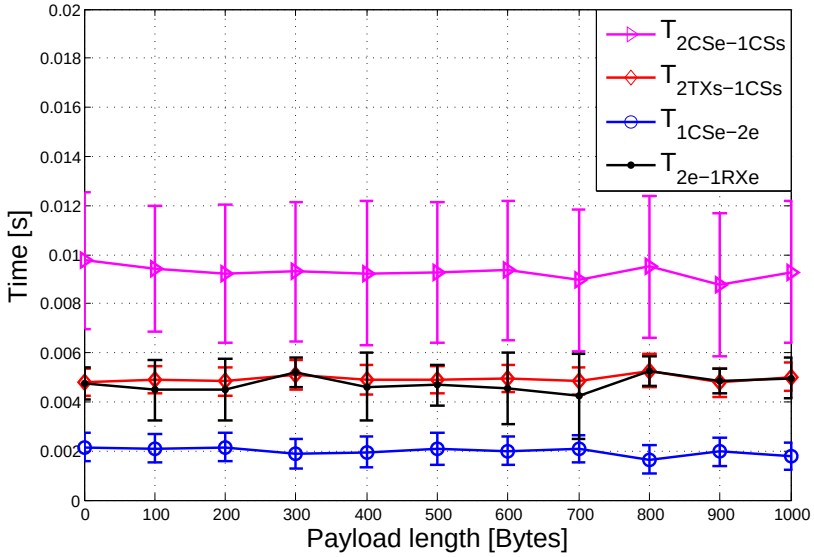


Figure 5.35: Delay between blocks in different nodes.

5.4 Performance Evaluation on MAC Protocols implemented with GR-DMDL

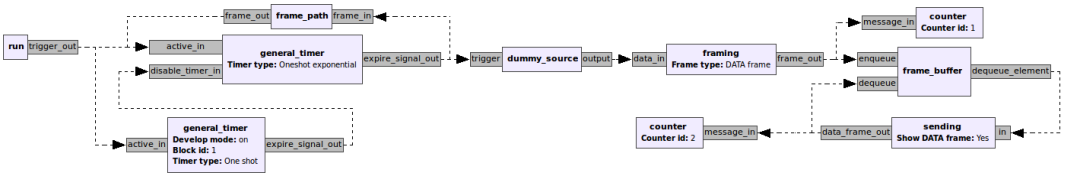
We implemented well known MAC protocols using GR-DMDL and evaluated their performance through long-term measurements. The actual performance of the tested protocols is either compared with theoretical results or previous implementation of the same protocol on older platforms. The comparisons show that GR-DMDL achieves a significantly high level of accordance with the theoretical expectations which makes it a powerful tool for future experimental verification, protocol design and prototyping.

Parameters	Value
Payload size	837 bytes
Frame size	1000 bytes
Frame transmission duration	40 ms
Sampling rate	400 k/s
Carrier frequency	430 MHz

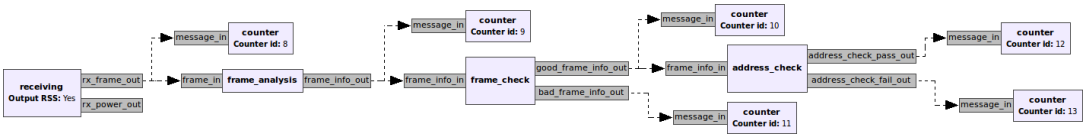
Table 5.2: Key parameters used classic MAC protocols evaluation.

We list several key configurations of the experiments in Table 5.2. The protocol specific parameters are defined in the corresponding subsections.

5.4.1 Pure ALOHA



(a) The GRC code of the transmitter in pure ALOHA measurement.



(b) The GRC code of the receiver in pure ALOHA measurement.

Figure 5.36: The GRC code used in pure ALOHA measurement in which two TX nodes and one RX node are used.

Pure ALOHA is one of the simplest MAC protocols. It was first developed at the University of Hawaii to provide packetized wireless communication services at Hawaii area where cable connecting all islands is uneconomic [72]. According to the types of synchronization, ALOHA protocols can be further categorized into pure ALOHA

and slotted ALOHA. In pure ALOHA, the rules are quite simple that nodes send data whenever they have to send.

In order to evaluate the performance of GR-DMDL, a measurement with three nodes is designed as shown in Figure 5.36. In the measurement, two transmitting nodes are equipped with a Poisson traffic source. Unlike the well know theoretical analysis in [9] where queueing is not considered, the ALOHA nodes in this measurement are equipped with a transmission buffer as shown in Figure 5.36a. In contrast to the theoretical model, the new model with buffer is more similar to the commercial off the shelf devices. Therefore, the transmitting node can be modeled as a queueing system. According to the [101], the transmitting node and the channel compose a classic M/D/1 system. Therefore, the arrive rate λ and the service rate μ of the queueing system are the average number of generated frames in time unit and the reciprocal of the frame transmission duration $1/T_{\text{fr}}$, respectively. The probability of a successful transmission is,

$$P_{\text{success}} = \left(1 - \frac{\lambda}{\mu}\right) e^{-\frac{\lambda}{\mu}}. \tag{5.2}$$

In the experiment shown in Figure 5.36, the generated payload length is fixed to 837 bytes. Adding the MAC header and PHY overhead, the overall length is 1000 bytes. The sampling rate is set to 400000 samples/s and the modulation is QPSK. Therefore, the transmission time of one frame is 40 ms which is the reciprocal of μ. The mean waiting time of the Poisson source is set from 800 ms to 42.1 ms so that the channel load in this measurement is tested from 0.05 to 0.95. On the transmitter side, two counters are used to record the number of generated frames and actually sent frames. When one round of measurement finishes, the differences between these two are the number of frames which are stocked in the queue. However, this measurement only considers the lightly loaded scenario which the channel load is less than one, so the frames should not wait in the queue frequently. This is also observed in the experiments. On the receiver side, five counters are used to record the reception, as shown in Figure 5.36b, namely counter 8, counter 9, counter 10, counter 11, and counter 12. Each counter records a step in the reception. Counter 8 records the received PHY packet, more precisely, the records the number of detected PHY preamble. Counter 9 records a received MAC frame header. We note that the packet or frame which is counted at counter 8 and counter 9 are not verified by checking the CRC sequence which lies in the end of the frame. Counter 10 records the number of bad frames that fail to pass the CRC check. Counter 11 and counter 12 record the received frames according to its destination address. In order to distinguish the source of the received frames, the two transmitting nodes are supposed to transmit frames with different destination addresses, and only one of them are destined to the receiver.

The throughput of the pure ALOHA is shown in Figure 5.37. The black curve represents the theoretical throughput which is drawn according to (5.2). The red line is the measured throughput. Even though the correlation coefficients of measured results and the theoretical results is 0.9985 and the Mean Square Error (MSE) of

the measured data is 0.1454, there is a visible gap between the two curves especially for the middle range channel load.

To further explain the gap, the probability analysis is retaken by considering the PHY packet structure in Figure 5.5 where all zeros padding is attached. The length of the padding is surely one part of the packet when it is generated and transmitted because the **pending_tx_finish** block calculates the pending time according to the whole length of the packet. However, the padding parts have no effect on the channel, i.e., if another node starts to transmit a frame when the second padding field of the previous frame is being transmitted, both of the frames are able to be received correctly. Therefore, the improved probability of a successful transmission is,

$$P_{\text{success}} = \left(1 - \frac{\lambda}{\mu}\right) e^{-\frac{\lambda L_{\text{packet}}}{\mu(L_{\text{packet}} - L_{\text{padding}})}}, \tag{5.3}$$

which is the blue curve in Figure 5.37. In contrast to the original one, the improved curve achieves 0.9992 correlation coefficients with the measured data, and the MSE is reduced to 0.022 which is so small as to be negligible in normal situations.

Figure 5.38 shows the normalized results of the same measurements with more results from intermediate steps, which is due to the benefits of using GR-DMDL probe and counter tools. All measured data are normalized by the number of transmitted frames at both transmitting nodes. Similar to Figure 5.37, the improved theoretical results almost overlap with the measured data which shows the accuracy of the GR-DMDL implementation.

The curve *PHY headers* records the total number of correctly received PHY headers, and the curve *MAC headers* records the number of correctly received MAC headers.

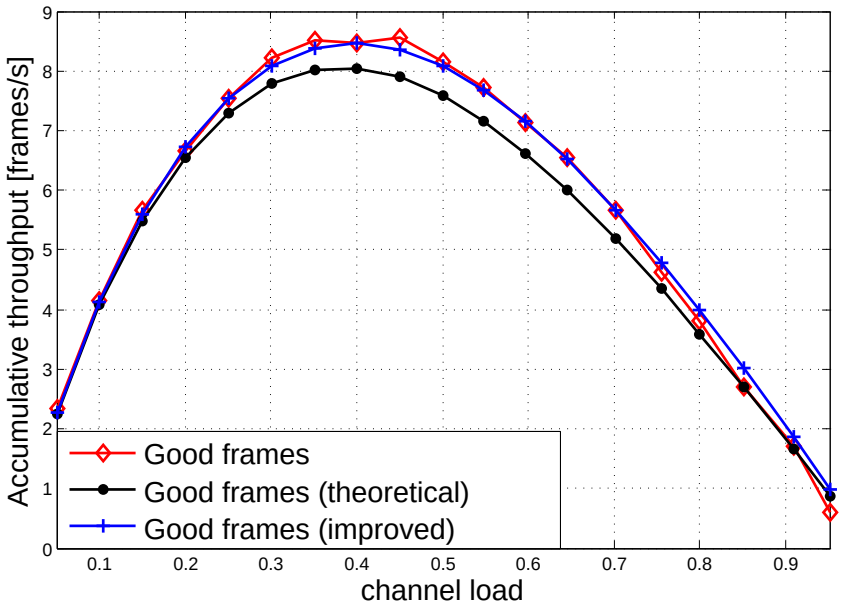


Figure 5.37: Throughput in pure ALOHA network with two transmitting nodes.

The difference between them is the case that when one node is transmitting its MAC header, another frame is start transmitting the PHY preamble which crashes the MAC header. Due to the limited size of the MAC header, the gap between these two curves is also small.

5.4.2 Slotted ALOHA

In slotted ALOHA, data are sent in synchronized time slots to lower down the probability of collisions by reducing the channel vulnerable time by half. From the implementation point of view, slotted ALOHA has more strict timing control than pure ALOHA, therefore it can be used for evaluating the timing performance and the capability of synchronization of GR-DMDL. The slotted ALOHA implementation is shown in 5.39, which has one independent time control branch in the design. The **standard_timer** exports *command* periodically according to the system time that is synchronized at everything PC with an Internet clock. This way, the time jitter will not be accumulated. In order to achieve a fair comparison with the pure ALOHA, the slot time is set to 40 ms, which is exactly the transmission time of a 1000 bytes frame with the PHY settings used in pure ALOHA measurement. However, in order to avoid the overlapping between slots, the guard interval is introduced to let the actual frame transmission time slightly smaller than the slot time, which is set to 2 ms in the measurement where the frame transmission time is reduced to 38 ms.

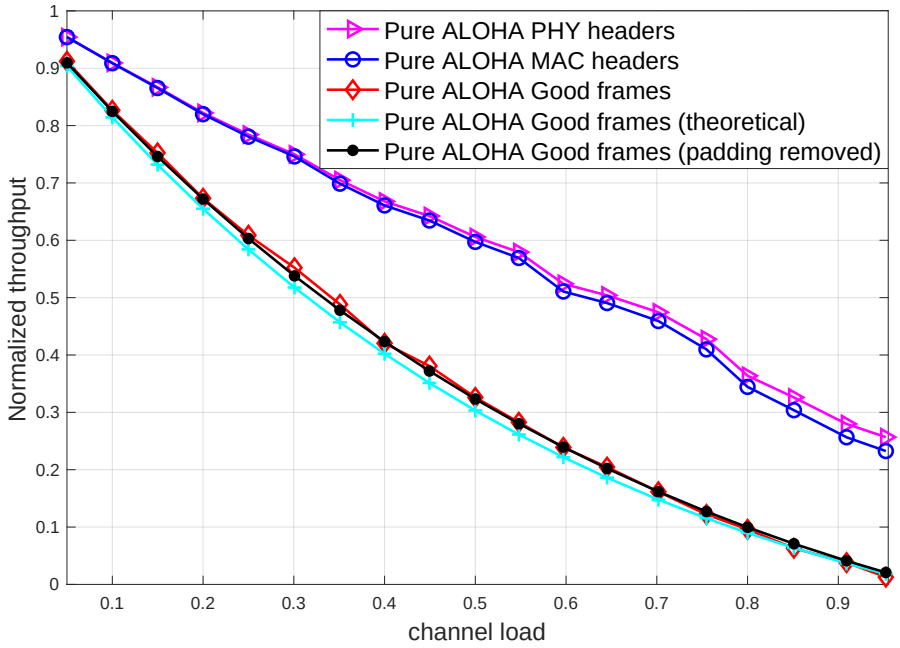
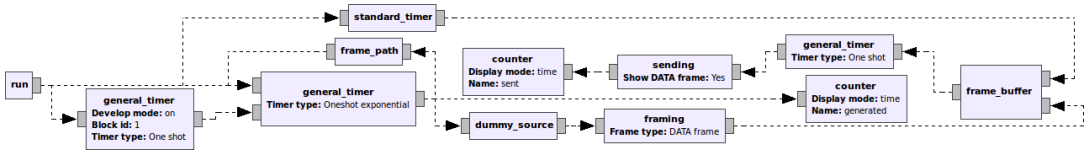


Figure 5.38: Normalized throughput in pure ALOHA network with two transmitting nodes.

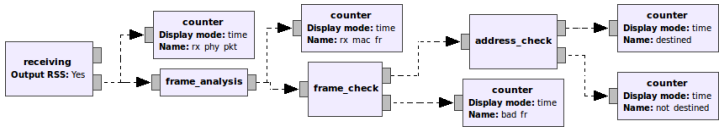
In slotted ALOHA, the deduction of theoretical throughput is slightly different from the pure ALOHA. Because both ALOHA use the same Poisson source, P_0 for slotted ALOHA is unchanged. However, the $P_{0\text{fr}}$ is changed to 1. It is due to the fact that in slotted ALOHA, if there is no frame waiting in the queue in the beginning of a time slot, then there will be no frame transmitted out in the time slot. Thus, the successful transmission probability in a slotted ALOHA network with two transmitting nodes is,

$$P_{\text{success}} = 1 - \frac{\lambda}{\mu}. \tag{5.4}$$

Besides the outperformed performance, Slotted ALOHA has more benefits. The padding problem which needs to be compensated in pure ALOHA does not exist in slotted ALOHA because the interference can only be the full frame in the collided slot. Due to the guard interval settings, It can also cover unstable time delay or jitter in the PHY layer.



(a) The GRC code of the transmitter in slotted ALOHA measurement.



(b) The GRC code of the receiver in slotted ALOHA measurement.

Figure 5.39: The GRC code used in slotted ALOHA measurement in which two TX nodes and one RX node are used.

The throughput of the slotted ALOHA is plotted in Figure 5.40 where the throughput of pure ALOHA is also given for comparison. The results show that the curve of the measured data almost overlaps with the theoretical result in the whole range of channel load. The correlation coefficient between them is 0.9992, and the MSE is only 0.017. The results show that the slotted ALOHA achieves not only better performance but also higher stability then pure ALOHA. Moreover, it verifies that the timing performance and synchronization of GR-DMDL are accurate.

More results of normalized throughput of slotted ALOHA are shown in Figure 5.41. In contrast to the pure ALOHA, the gap between the number of received PHY preamble and MAC headers in slotted ALOHA are similar to it in pure ALOHA protocols.

The GR-DMDL implementations of pure ALOHA and slotted ALOHA include many GR-DMDL blocks described in the previous part of this chapter. The empirical results in both pure ALOHA and slotted ALOHA measurements show that GR-DMDL implementation on GNU Radio can achieve very high accordance

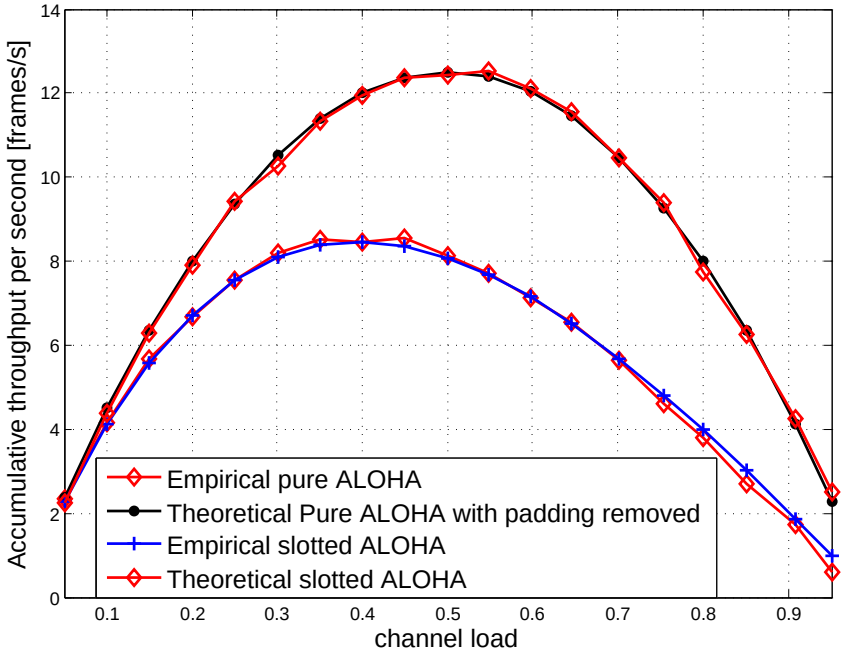


Figure 5.40: Throughput in slotted ALOHA network with two transmitting nodes.

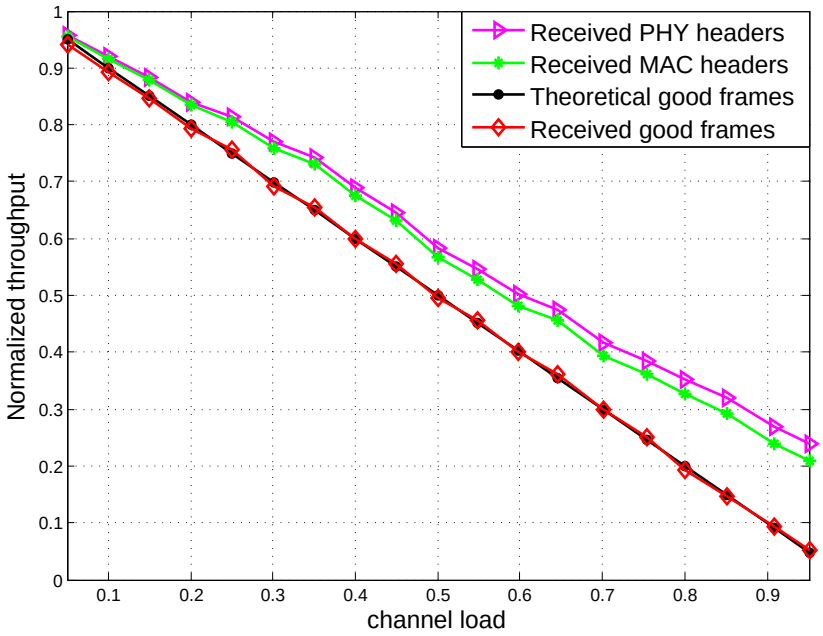


Figure 5.41: Normalized throughput in slotted ALOHA network with two transmitting nodes.

with the theoretical expectations. In the next section, the performance of the protocol with randomness and carrier sensing are considered in the p-persistent CSMA protocols.

5.4.3 *p*-persistent CSMA

The major problem of using the shared medium is that simultaneous transmissions may cause collision that can destroy all ongoing transmission. In order to avoid collision, nodes need to be able to get the channel status. CSMA/CA is a probabilistic MAC protocol which is based on the principle of "listen before talk". In wireless networks, a node running a CSMA/CA protocol, first senses the channel for other ongoing transmissions before it starts to transmit. If there is an ongoing transmission, the node first waits for the transmission to finish before starting its own transmission. The waiting time is determined by the Binary Exponential Backoff (BEB) algorithm, which avoids potential collisions caused by simultaneous transmitting attempts.

According to the detailed algorithms, CSMA has many variations, e.g., 1-persistent, non-persistent, and *p*-persistent. In all of the variations, *p*-persistent CSMA is the technology widely used in CSMA/CA systems such as the IEEE 802.11 DCF. Therefore, we choose to test the performance of GR-DMDL on CSMA type of protocols. In *p*-persistent CSMA, when a node is ready to transmit data, it senses the channel to detect the channel status. If the channel is idle, it transmits a frame with probability p. Otherwise, it continuously senses the channel until it turns to idle, and then sends the frame with the same probability p. Similarly to the slotted ALOHA, the concept of time slot is also similarly used in *p*-persistent CSMA. Nodes can only send a frame in the beginning of a time slot. If the node does not transmit, it waits until the next available time slot.

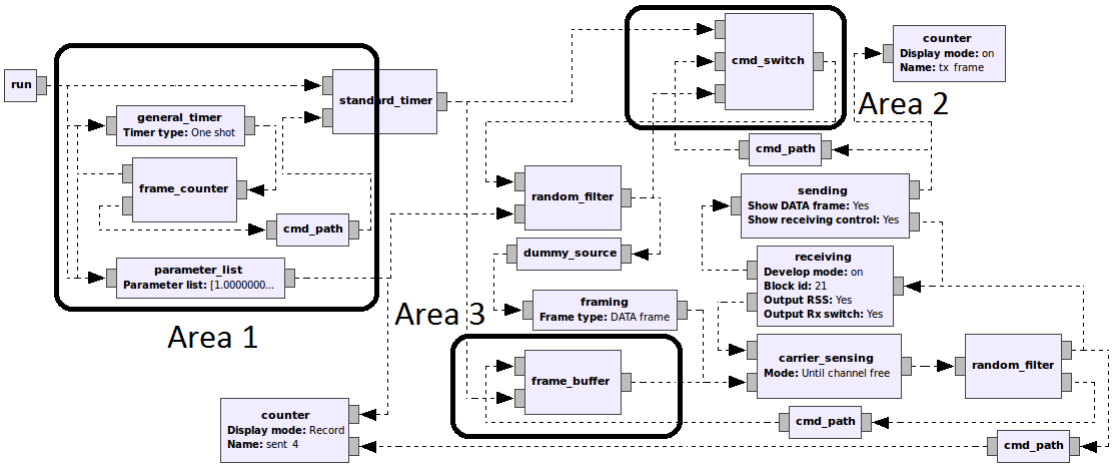
The performance of slotted *p*-persistent CSMA has been well studied in literatures. According to the discussion in [41], the throughput of *p*-persistent CSMA with two transmitting nodes is,

$$S = \frac{2p\sum_{k=0}^{\infty}\left(r^k - t\right)^{1+\frac{1}{a}}\left[\frac{pr^k - gt^k}{p-g}\right]\left(r^{k+1} - t^{1+\frac{1}{a}}p\left[\frac{r^{k+1}-t^{k+1}}{p-g}\right]\right)}{1 + a + a\sum_{k=1}^{\infty}\left(r^k - t^{1+\frac{1}{a}}p\left[\frac{r^k - t^k}{p-g}\right]\right)}, \tag{5.5}$$

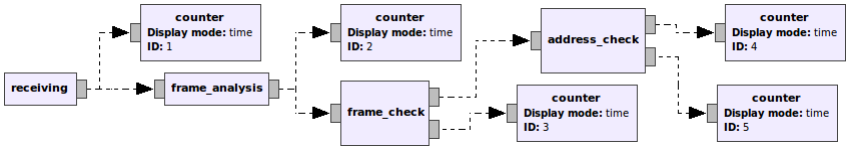
where p is the transmission probability at the beginning of a time slot which after an idle time slot, g is the frame generation probability at the beginning of a time slot in which no frame is being sent, and a is the propagation delay which is also defined as the slot time in terms of frame transmission time. To simplify the notation, we also let $r = 1 - p$ and $t = 1 - g$. Due to the relatively short distance in the laboratory environment, a is set to 4 ms according to the average processing delay measured in Subsection 5.3.3 and Subsection 5.3.4, and the length of tail effects shown in Figure 5.24b. The protocol is shown in Figure 5.42. In order to design

protocols exactly fit the description in [41], a more strict synchronization is applied in this design. As shown in Area 2 of Figure 5.42, the output *command* of the **standard_timer** block which is synchronized with the system clock is used to drive the whole system for transmission, carrier sensing and frame generation. It is set to periodic mode with period aT_{fr}. In Area 3, the **frame_buffer** block is used for fitting frame transmission into time slots. The buffer size of the it is set to 1 so it can only contain one rescheduled frame which passed the carrier sensing but fail to be sent due to the probability $(1-p)$. The *dequeue* port of the buffer is connected to the **standard_timer** block so even a rescheduled frame arrives at the buffer, it will be sent only after the next time slot.

Due to the large number of measurements are needed to be taken for evaluating p-persistent CSMA, a new structure for continuously taking multiple measurements is used as shown in Area 3 of Figure 5.42. The **parameter_list** block is able to save a vector of numbers, and output them individually when a trigger signal arrives at its input port. A **frame_counter** block is used for counting the number of expirations of the **general_timer** blocks. If the number of expirations is less than the desired number of measurements. The **frame_counter** reactivates the **general_timer** block for counting down another round of measurements and also pop up the corresponding parameter from the **parameter_list** to the **random_filter** block through its *reset_in* port to change the frame generation probability per time slot. The output of the **parameter_list** block also triggers the **counter** block to record the number of visits after the arrival of the last trigger on a text file and reset



(a) The GRC code of the transmitter in slotted p-persistent CSMA measurement.



(b) The GRC code of the receiver in slotted p-persistent CSMA measurement.

Figure 5.42: The GRC code used in slotted p-persistent CSMA measurement in which two TX nodes and one RX node are used.

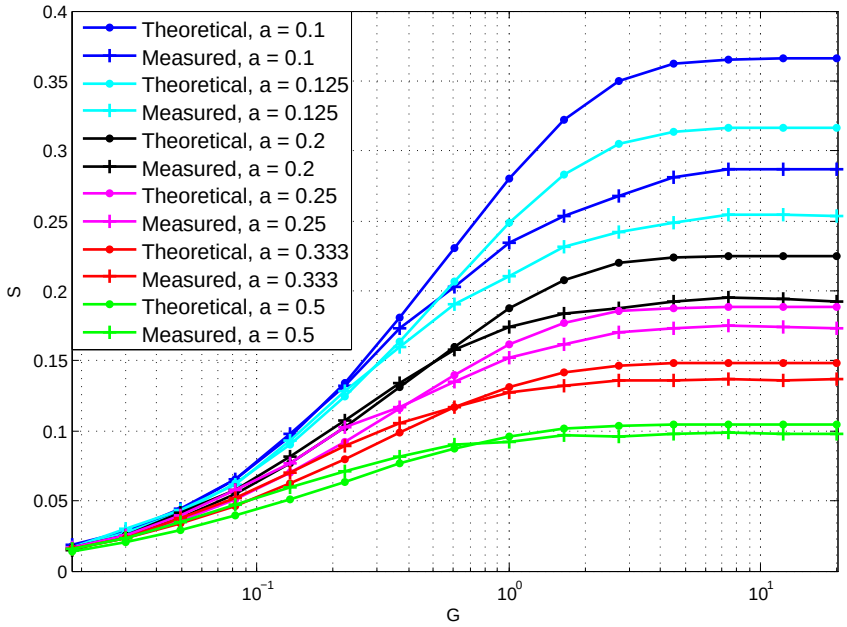


Figure 5.43: Throughput comparison of theoretical and empirical performance of p-persistent CSMA protocols with two transmitting nodes.

the counts to zero. By adding such a simple structure containing only two more blocks, the designed protocol is able to run measurements continuously. It keeps the experimental conditions stable during the experiments and also save time to reconfigure the protocol between measurements.

The experimental results are plotted in Figure 5.43 together with the theoretical values from (5.5). To ensure $\frac{1}{a}$ is an integer, a is set to 0.1, 0.125, 0.2. 0.25, 0.333, and 0.5, p is set to 0.03 and G is set to the channel load which is defined as,

$$G = \min(1, \frac{gM}{a}), \tag{5.6}$$

The experimental results are plotted in Figure 5.43 together with the theoretical values from (5.5). To ensure $\frac{1}{a}$ is an integer, a is set to 0.1, 0.125, 0.2, where g is the same frame generation probability per time slot. In Figure 5.43, the curves of experimental performance have the same trend as the curves of theoretical results with varying channel load G. The MSE between the empirical data and theoretical results is plotted in Figure 5.44. Both figures show that the implementation of p-persistent CSMA in GR-DMDL matches the theory.

We find that with the fixed frame transmission duration, the greater a value, i.e., the longer the slot time the protocol deploys, the lower MSE it achieves. It is due to the fact that the corresponding slot time of small a, e.g., 0,1 or 0.125, with the frame transmission duration of 40 ms are less than 10 ms which is even shorter than some of the processing delay as shown in Figure 5.29 and Figure 5.34. A relatively small slot time not only introduces the processing delay into the account, but also lowers

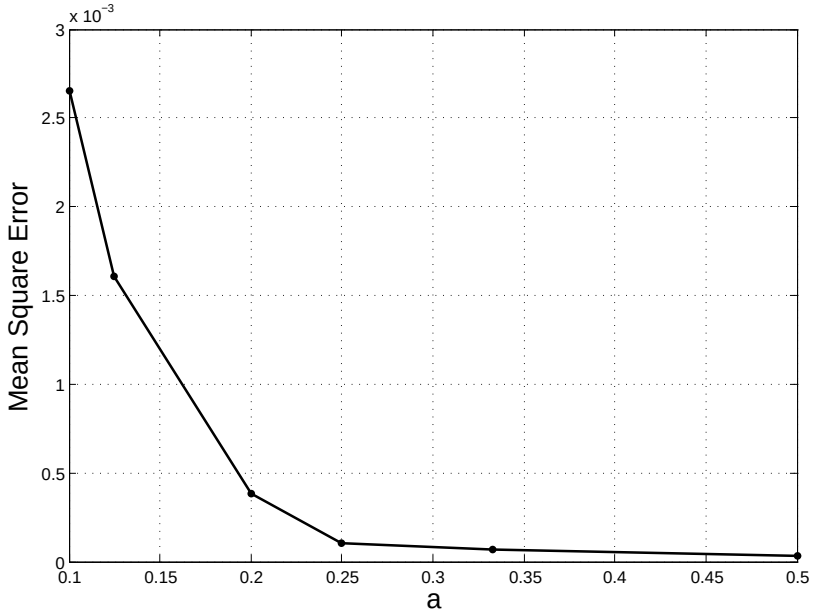


Figure 5.44: MSE of different a values between theoretical and empirical performance of p-persistent CSMA protocols with two transmitting nodes.

down the system stability. However, it is still worth noting that the performance of smaller value of a agrees very well with the theory in low channel load scenarios.

Besides the basic CSMA series protocols, the CSMA principle is also widely used in industry standards, such as the IEEE 802.11 standard and Bluetooth. In the next subsection, we introduce the implementation of the IEEE 802.11 like protocols.

5.4.4 IEEE 802.11 DCF

It has been 20 years from the first release of the IEEE 802.11 standard. In the last decades, the scope of the 802.11 family has been continuously updated and extended. However, the principle of its fundamental channel access method has never been changed. It uses carrier sensing and binary exponential backoff algorithms to efficiently manage the behaviors of multiple half duplex wireless nodes to lower the probability of collision in a shared medium.

In this section we will show the performance of the IEEE 802.11 DCF like protocol with two-way handshake. This time, the theoretical flowchart shown in Figure 4.30 is filled up with real GNU Radio blocks as shown in Figure 5.45. The GR-DMDL blocks almost perfectly fit the flowchart which proves that DMDL is an efficient tool to unify the protocol design and the protocol implementation. The only difference between the flowchart and the implementation is the source related blocks that lay on the right of the canvas. For the sake of consistency, the same type of source control as in the p-persistent CSMA implementation is used in the this implementation.

Parameters	Value
Sampling rate	400 k/s
Modulation	QPSK
Sample per symbol	4
DATA frame transmission time	40 ms
ACK frame transmission time	12 ms
SIFS	10 ms
Slot time	8 ms
DIFS	26 ms
Frame payload length	837 Bytes
MAC and PHY overhead	163 Bytes
Maximum retransmission limits	6
Timeout interval	100ms

Table 5.3: Key parameters used in IEEE 802.11 DCF measurements.

To show the capability of GR-DMDL in measuring details of a protocol and carrying out complex experiments with multiple parameters, several helping blocks are attached to the MAC protocol as shown in Figure 5.46. Area 1 is the logics of the protocol shown in Figure 5.45. Area 2 is the logics of measurement, control, which export experimental parameters periodically to measure the performance of the protocol with different configurations. As shown in Figure 5.46, the output *command* of the block **parameter_list** changes the frame generation rate of a time slot. The exported parameter also resets all the counters in Area 3, 4, and 5, so the counters and recorders are able to save the counts in each measurement. In the measurement design shown in Figure 5.46, up to 17 system metrics are traced and recorded. The parameters used in the IEEE 802.11 DCF measurements are listed in Table 5.3. Each measurement is carried out at lease one hour to ensure the statistical stability of the empirical results.

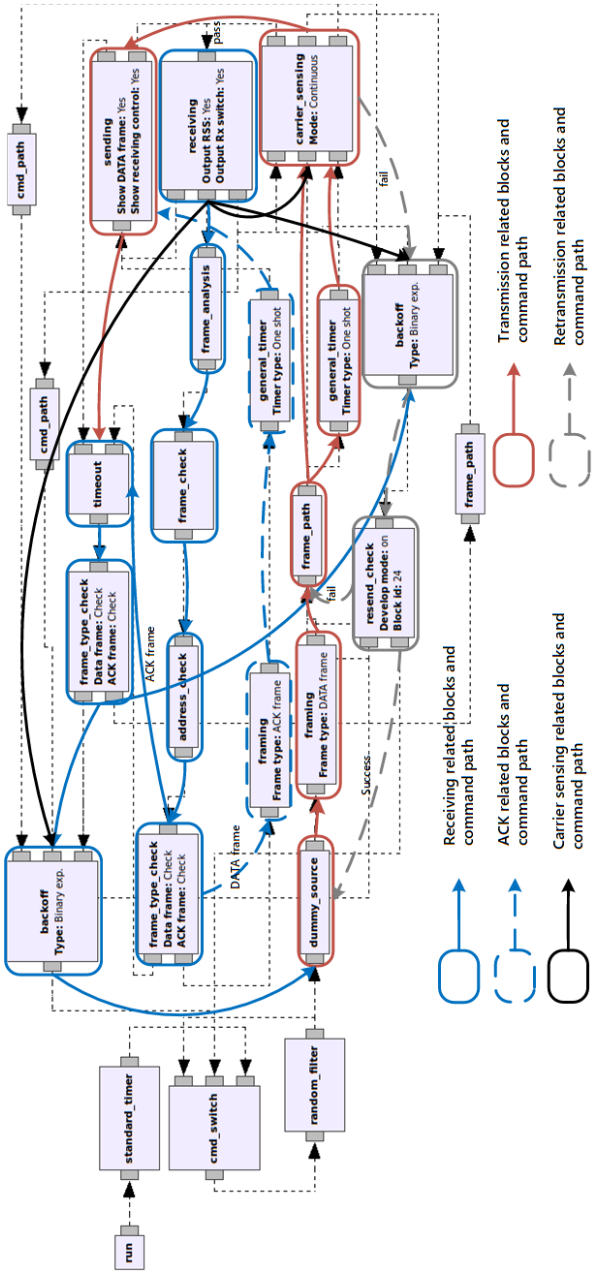


Figure 5.45: GNU Radio implementation of the classic 802.11 DCF flowchart shown in Figure 4.30.

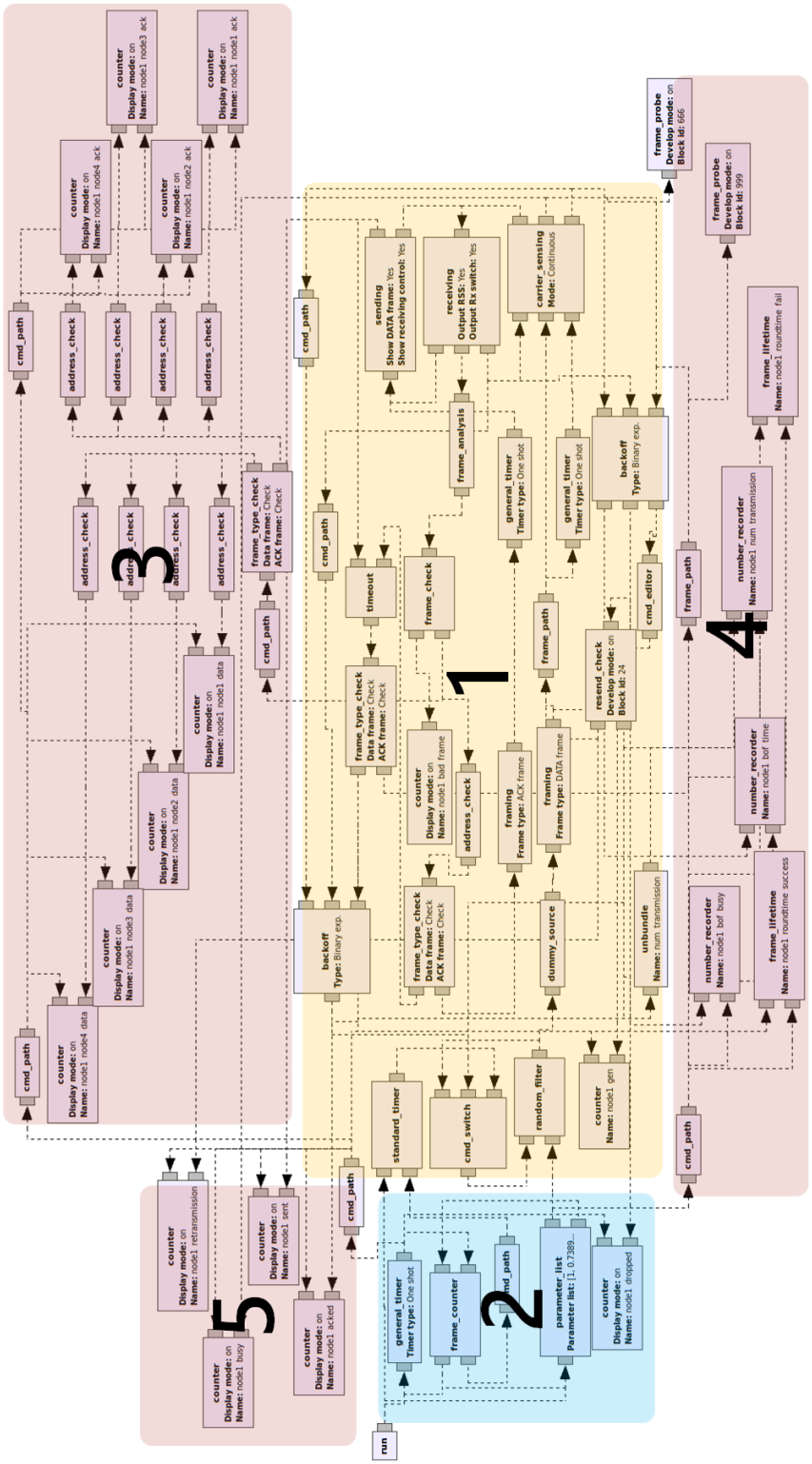


Figure 5.46: Implementation of the IEEE 802.11 DCF with all experimental setups.

There are two series of measurements were carried out. In the first series of measurements, the overall throughput and the average throughput of networks with different number of nodes are measured and compared. In the second series of measurements, the minimal contention window size of the binary exponential backoff algorithm is changed. The empirical data agrees well with the theoretical analysis in previous studies, e.g., the relation between the minimum contention window size and the throughput [31]. Thanks to the complete recording system in Figure 5.46, a number of detailed features of the IEEE 802.11 standard which have not been sufficiently investigated and discussed can also be obtained and studied.

5.4.4.1 Performance of network with respect to number of nodes

In the first serious measurements, the number of nodes ranges between 2 and 5. The function of generating frame at one node is removed to let it become an access point, and all other nodes are supposed to transmit to it. Therefore, all nodes together compose a classic WIFI network including one node as the access point and all others as the clients.

The goodput of the tested network with different number of nodes is shown in Figure 5.47. The goodput is recorded at each client and only the successfully acknowledged DATA frames are counted. The **counter** block with name *node1_acked* in Figure 5.46 area 2 traced the *command* sent out of the **backoff** block which performs the backoff with minimal contention window size after a DATA frame is successfully acknowledged. The minimal contention window is set to 1 in this measurement, and the maximum retransmission limits are set to 6. The overall goodput is the total goodput achieved by all nodes, and the average goodput is averaged from all nodes. The results show that the maximal overall goodput appears in the two-TX-node scenario while the maximal average goodput is achieved when only node is running. It is due to the fact that small contention window size increases the probability of collisions, thus lowers down the total goodput. It further verifies the explanation that the total goodput of the 3 nodes is higher than 4 nodes cases because the latter network has higher contention level. All curves are converging with the increment of frame generation probability at the source. The extreme case which happens at the rightmost measuring point where the frame generation rate is one which means nodes always have frame to send. When the frame generation rate is low, network with four nodes and three nodes outperforms the network with two nodes. It is owing to the fact that with low rate source the network can transmit every frame it generated. However, it is interesting that the network with two nodes achieves the highest performance when the network is saturated or closed to saturation. Despite of the relatively small minimum contention window size used in the experiment which is 4 in the standards, but only one in the measurement, it shows that the 802.11 DCF suffers from the crowed environment. A more complete investigation between the minimal contention window size and the performance is given in the latter part of the subsection.

The goodput results verify that the GR-DMDL implementation is capable of implementing carrier sensing based complex MAC schemes. The following figures

demonstrate more results from the same measurements and prove that GR-DMDL is able to measure all aspects of a MAC protocol.

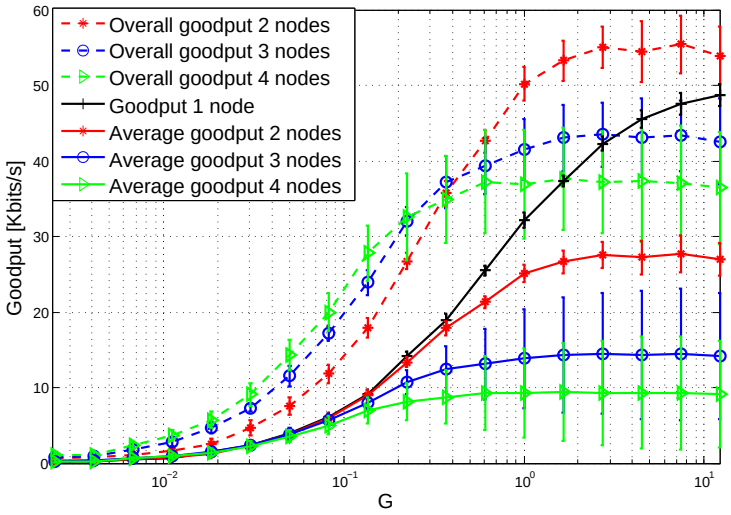


Figure 5.47: Up link Goodput of networks with different number of client nodes running 802.11 DCF. Goodput is calculated by only counting the payload part of successfully acknowledged frames at each transmitter. Number of nodes is varying from 1 to 4.

The number of sent frames is also counted by the **counter** block named *node1_sent* in Figure 5.46. It is recorded the actual number of frames are transmitted over the air, therefore it is able to help in understanding how saturated the shared medium is. According to Figure 5.48, networks with more nodes send more frames, which is agreed with the intuition. However, due to the collision and failed carrier sensing caused backoff, nodes in a crowded network send less frames. Again, due to the backoff algorithm which cancels some transmission attempts, the sent frames in a four-TX-node network is less than twice of the frames sent by a network with only one node when both of them are saturated.

The generation rate of the entire network and at each node is also recorded. The curves in Figure 5.49 show the similar trends as the curve in Figure 5.47. Comparing these two figures, we find that the generation rates are higher than the goodput in all scenarios. It is due to the fact that some of the frames, most likely in high competitive environments, are lost due to the exceed of maximum retransmission limits. We note that the differences between the total generated frames and total acknowledged frames in two-TX-node network and single link network are very little. Similar to Figure 5.47, the two-TX-node network achieves the highest rate due to the relatively low contention.

The number of retransmission of networks containing a different number of nodes is shown in Figure 5.50. It is recorded by the **counter** block named *node1_num_retransmission* in Area 5 of Figure 5.46. It counts every *command* out of the port *resend_check_pass_out* of the **resend_check** blocks which contains the frames that either failed in carrier sensing or is not acknowledged during the timeout

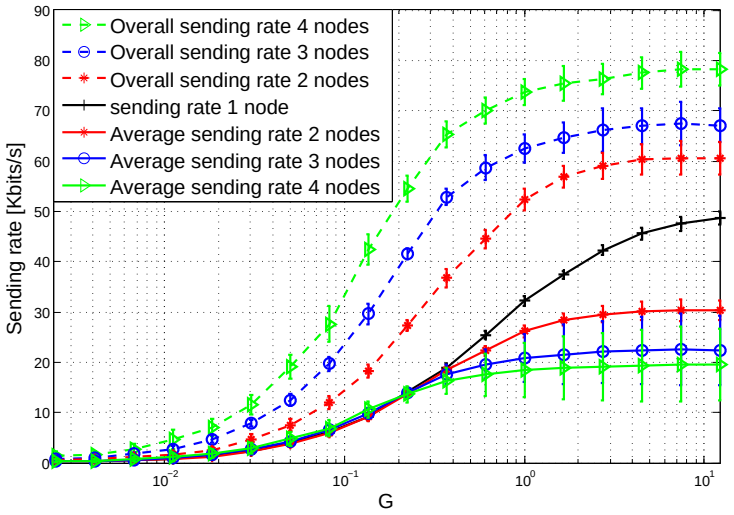


Figure 5.48: Sending rate of networks with different number of client nodes running 802.11 DCF. Sending rate at a node is calculated by including all payload of data frames sent by the transmitter.

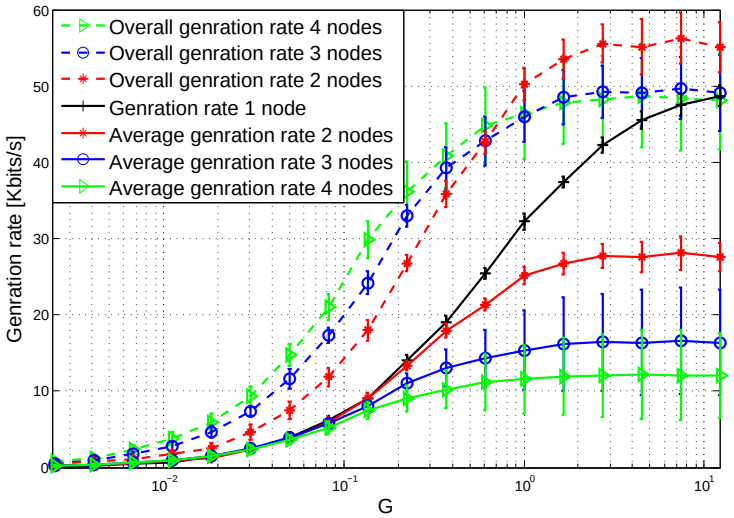


Figure 5.49: Generation rate at networks with different number of client nodes running IEEE 802.11 DCF. Generation rate at a node is calculated by including all payload of all data frames generated by the node.

period. The curves in the figure agree with the above measurement results. The single link network never retransmits due to its contention free medium, the average number of retransmission at nodes in other networks are quite similar. However the total number of retransmission in the four-TX-node network and the three-TX-node network are much higher. However, it is worth noting that not all retransmissions are actually sent out from the radio because retransmission may also fail in carrier sensing and backoff again without being physically sent.

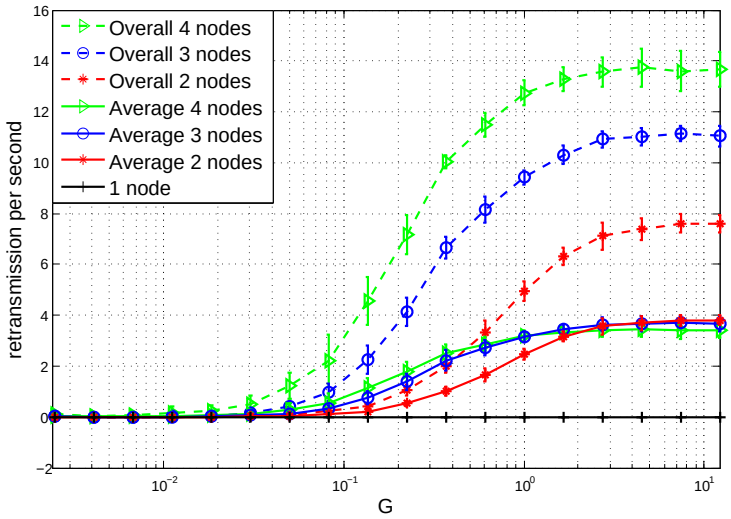


Figure 5.50: Average retransmissions per second at a single node in a 4-node wireless network running 802.11 DCF. Retransmissions at a node is only calculated by including retransmitted DATA frames from the node.

HOW The curves of dropped frames are plotted in Figure 5.51. In 802.11 DCF, frames are dropped because of they are retransmitted more than the maximum retransmission limits. Similar to the number of retransmissions, the number of dropped frames of the single link network is always zero. Due to the relatively low contention level, the number of dropped frames of two-TX-node network is also quite low. That is, even though frames are retransmitted in the two-TX-node network, the maximum retransmission limits are hard to reach. Due to the binary exponential backoff algorithm, frames are relatively easy to be sent even it is retransmitted couple of times. Nevertheless, the number of dropped frames increases significantly if the network size is increased. With saturated source, the total number of retransmissions in the four-TX-node network is less than 1.3 fold of the retransmission in the three-TX-node network, but the former yields almost double total dropped frames. The results reveal that crowded network drops much more frames than network with fewer nodes, in particular with saturated traffic.

Another interesting metric is the successful transmission ratio that is defined as the ratio between the number of frames that are actually acknowledged out of all sent frames. It is also a criterion of how efficient the network uses the medium. The results are shown in Figure 5.52. In low traffic scenarios, the single link network, the two-TX-node network, and even the three-TX-node network can achieve perfect transmission, i.e., all sent frames are successfully acknowledged in time. In contrast, the four-TX-node network starts to lose even in the low traffic scenarios. It is due to the fact that network with more nodes has higher risks of collision that destroys all collided frames. However, the successful transmission rate also highly depends on the contention window size. As shown in Figure 5.66, the same four-TX-node network can achieve the perfect successful transmission ratio as the network with less nodes if the minimum contention size is greater than 1. The results shown in

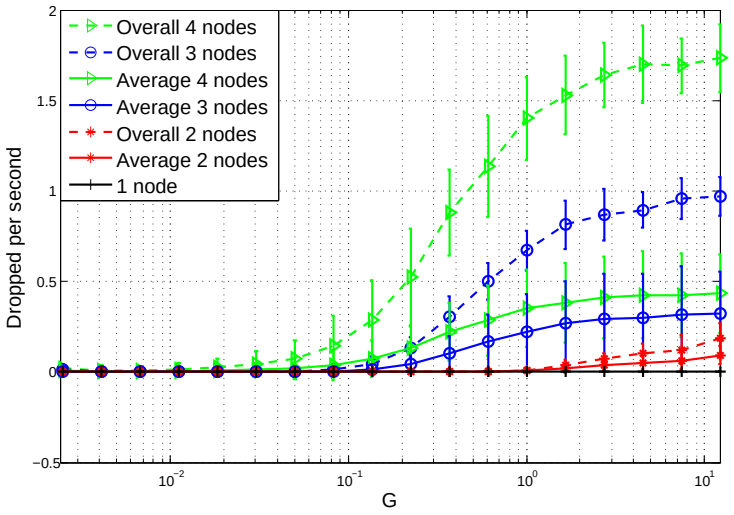


Figure 5.51: Average number of dropped frames per second at a single node in a 4 nodes wireless network running 802.11 DCF. Dropped rate at a node is only calculated by including dropped DATA frames from the node which is caused by exceeding maximal number of retransmissions.

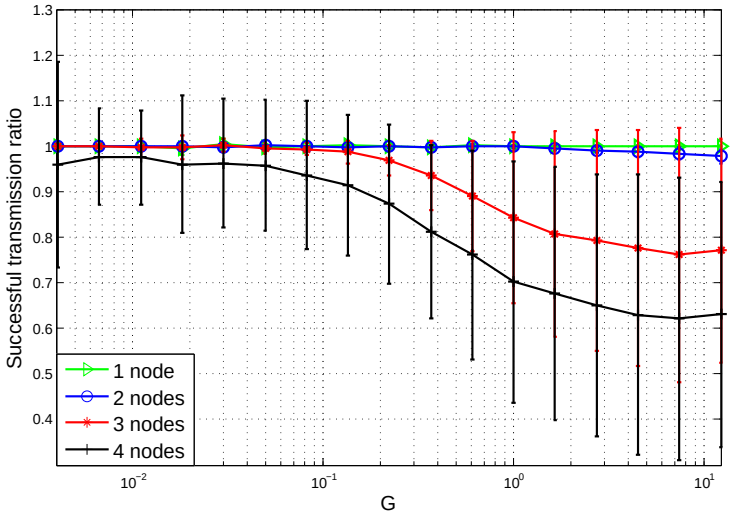


Figure 5.52: Successful transmission ratio of 4 nodes running 802.11 DCF. Successful transmission ratio at a node is calculated by the number of correctly acknowledged frames divides sent DATA frames.

Figure 5.66 will be revisited in the later part of this subsection.

The round trip time of a DATA frame is also one of the key metrics of a wireless network because it reflects the speed of the network processing data and also indicate the average delay the upper layer protocols may have. The round time is defined as the duration started at the time point that the DATA frame is generated until it is correctly acknowledged, therefore, retransmissions and the caused backoff duration is the major impact, i.e., more retransmissions causes longer round trip time. In

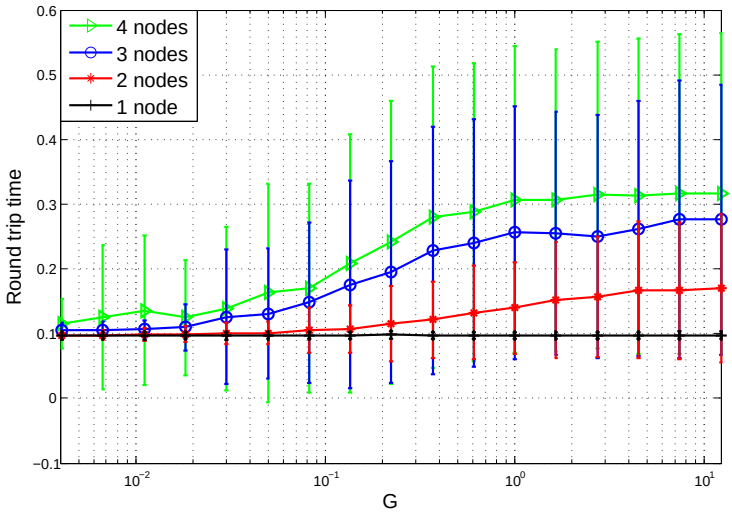


Figure 5.53: Average round trip time of successfully acknowledged DATA frames. The round trip time at a node is calculated by counting the duration between the DATA frame is generated until it is correctly acknowledged.

area 4 of Figure 5.46, the block **frame_lifetime** named *node1_roundtime_success* stores the round trip time of every successfully delivered DATA frame. By averaging all records, the systematic performance of round trip time is plotted in Figure 5.53. Again, the round trip time of frames in the single link network is constant because no retransmission and backoff takes place. As expected, the network with more nodes has longer round trip time. Another interesting observation is, the three-TX-node network achieves almost perfect successful transmission ratio, however, its round trip time is obviously higher than the two-TX-node network and the single link network. It is due to the fact that nodes in the three-TX-node network has a higher probability of detecting channel busy in carrier sensing. Detecting the channel busy stops the physical transmission, however, it still prolong the round trip time because it causes one more backoff.

The next interesting performance metric is the average number of transmissions per each generated DATA frame. This metric is different from the value measured in Figure 5.50. It records the number of transmission attempts of every DATA frame. In other words, it is also equal to the number of carrier sensing performed for every DATA frame. As shown in Figure 5.54, the single link network always has the average number of transmission equal to 1 because it does not retransmit. The maximal number of retransmissions for the two-TX-node network is less than 2, which means that even in the heavily saturated scenario, DATA frames in two-TX-node network are transmitted no more than two times. The four-TX-node network has the highest number of transmissions, however, due to the maximum retransmission limits, which means that a DATA frame can only be transmitted 6 times, the number of transmissions of the four-TX-node network is only slightly higher than the three-TX-node network.

As analyzed above, the binary exponential backoff algorithm enables the IEEE

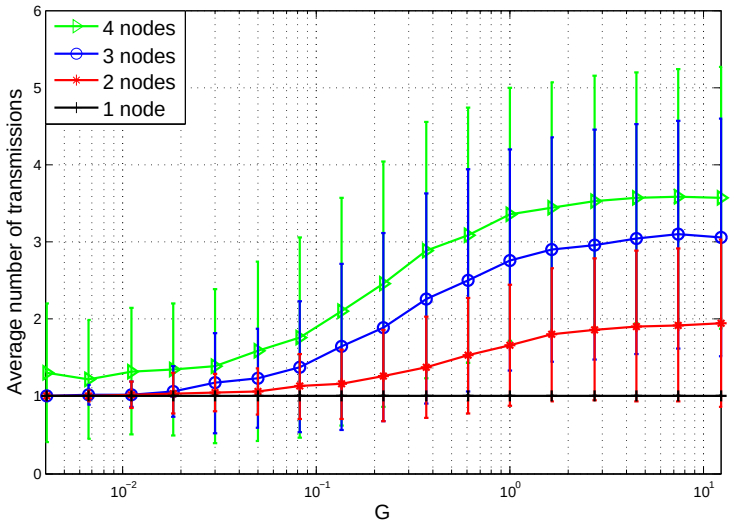


Figure 5.54: Average number of transmissions of DATA frame of 4 nodes running 802.11 DCF. The number at a node is calculated by tracking all generated DATA frames until it is successfully acknowledged or dropped.

802.11 DCF to efficiently avoid collision. It is also able to improve the fairness among the IEEE 802.11 DCF nodes that are supposed to be completely distributed. Therefore, recording the backoff time can provide an in-depth understanding of the standard. To get the duration of backoff for all backoff activities, a **number_recorder** block is inserted in the Area 4 of Figure 5.46. The value of each backoff is exported by the **backoff** block from a specifically designed output message passing port. The backoff time of the single link network is always 0. In the multi-node networks, the average backoff time in networks with more nodes is also higher. It is due to the fact that the number of nodes also indicates the contention level.

5.4.4.2 Performance of network with respect to minimum contention window size

We showed in the last measurements that the single link network and the two-TX-node network perform well with even heavily saturated traffic and small minimum contention window. Therefore, it is more beneficial to investigate the relations between the minimum contention window size and the performance metrics in a multi-node network. Considering that the very small contention window size, i.e., less or equal than two, or too large contention window size, e.g., greater than the one used in industry standards, are less meaningful, the minimum contention window size varies from 4 time slots to 32 time slots in the following measurements.

The average goodput of nodes is illustrated in Figure 5.56. The networks with small minimum contention window size outperform the networks with large minimum contention window size. It is due to the fact that large minimum window size

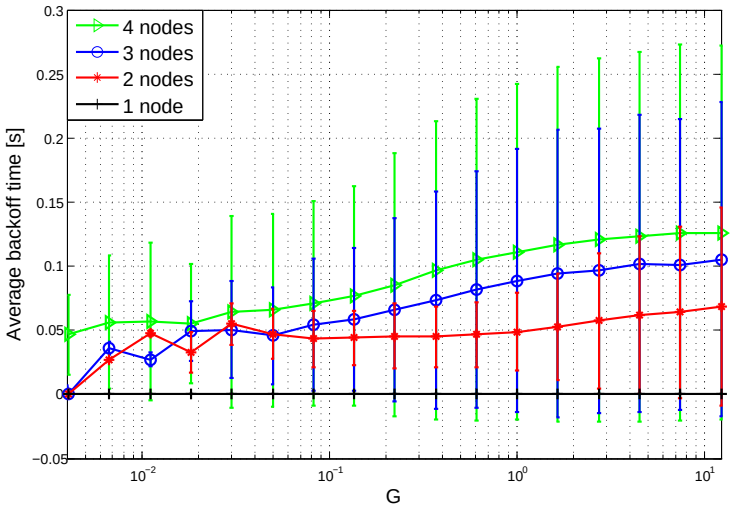


Figure 5.55: Average backoff time of 4 nodes running 802.11 DCF. Backoff time is calculated by averaging the duration of all backoffs.

forces the protocol to wait relatively long time even after a successful acknowledged transmission. The figure also shows that for the same minimum contention window size, the three-TX-node network always outperforms the four-TX-node network with the same configuration. Another interesting observation is that the average goodput gap between the three-TX-node network and the four-TX-node network with the same configuration is increased with the increment of the minimum contention window size. The the three-TX-node network and the four-TX-node network with minimum contention window size 32 have almost the same average goodput, while the three-TX-node network achieves 1.5 fold goodput compared to the four-TX-node network if the contention window size is 4. The results show that higher minimum contention window size can improve the fairness in multi-node networks.

When looking into the overall uplink goodput measurements shown in Figure 5.57, one can observe that the goodput of network with different number of nodes also depends on the minimum contention window size. With small minimum contention window size, the three-TX-node network outperforms the four-TX-node network because in the three-TX-node network the contention level is lower. It leads to the lower probability of collision and retransmission that finally improves the overall goodput. On the contrary, network with more nodes achieves higher average goodput with greater minimum contention window size. It is owing to the fact that the longer backoff duration lowers down the risk of collision. The figure also shows that when the source generating rate is low, the four-TX-node network outperforms the three-TX-node network. On the contrary, their goodputs are quite similar when the source generates saturated traffic.

The average and total uplink sending rates are also measured. As shown in Figure 5.58, nodes in the three-TX-node network and the four-TX-node network send a similar number of frames. However, the differences appear as the minimum contention window size is getting smaller. Due to the relatively low contention

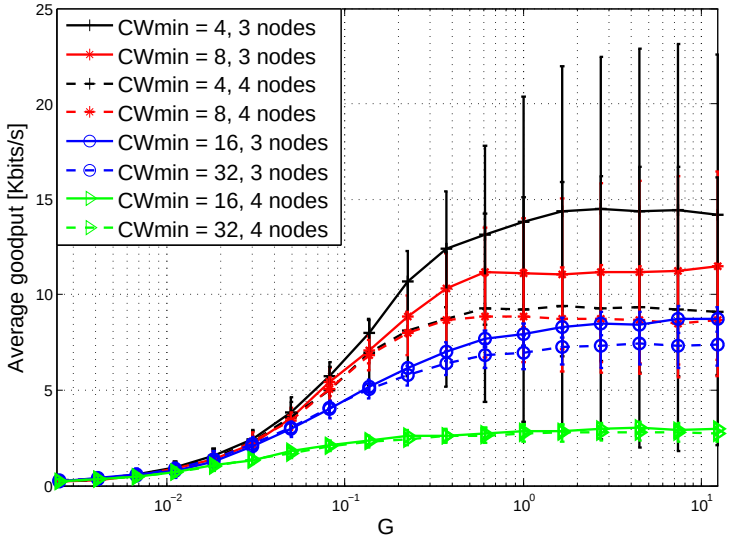


Figure 5.56: Average up link Goodput at a single node in a 4 nodes wireless network running 802.11 DCF. Goodput is calculated by only counting the acknowledged frames at each transmitter.

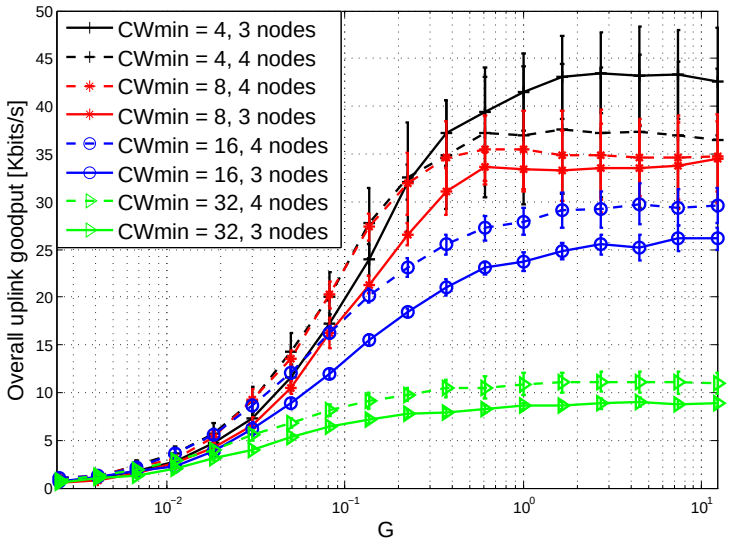


Figure 5.57: Overall uplink Goodput of 4 nodes running 802.11 DCF. Goodput is calculated by only counting the acknowledged frames at each transmitter.

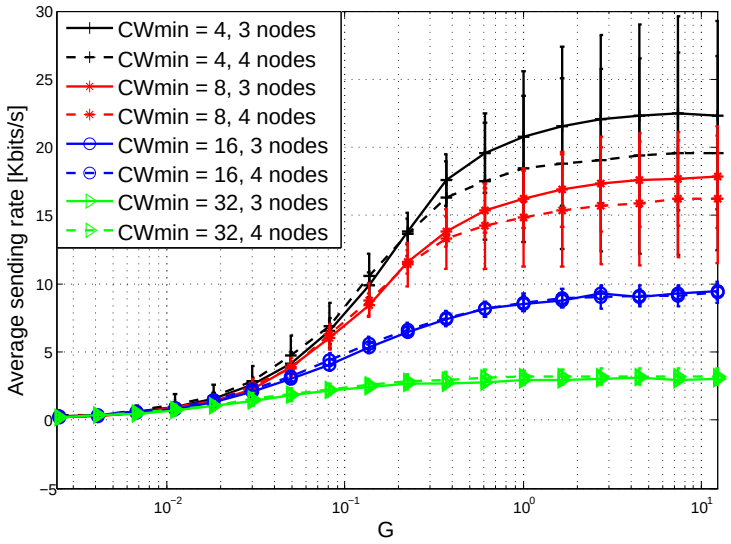


Figure 5.58: Average up link sending rate at a single node in a 4 nodes wireless network running 802.11 DCF. Sending rate at a node is calculated by including all data sent by the transmitter.

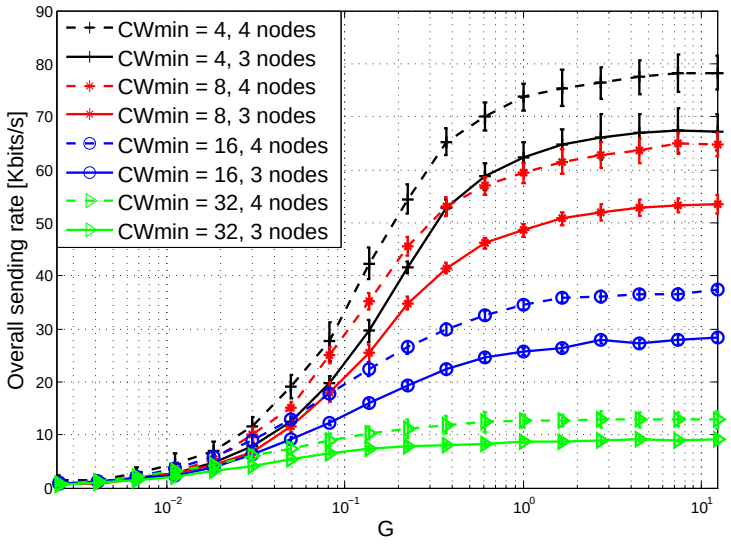


Figure 5.59: Overall sending rate of 4 nodes running 802.11 DCF. Sending rate at a node is calculated by including all data sent by the transmitter.

level, nodes in the three-TX-node network send more frames than nodes in the four-TX-node network. However, the difference of the average sent frames is quite limited. So the four-TX-node network sends more frames in total. The results from this measurement also verify the results shown in Figure 5.58 that the networks with more nodes send more frames.

In Figure 5.60, the number of generated frames in the measurements follows almost the same trend as the number of sent frames. The major difference between them

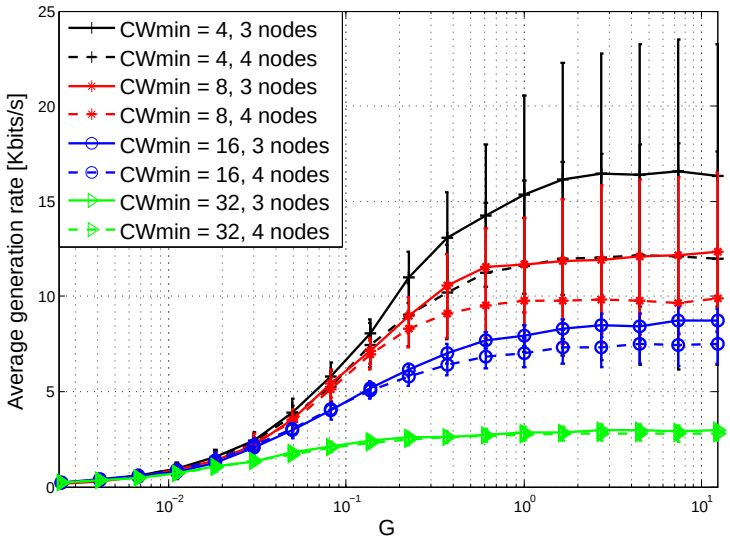


Figure 5.60: Average generation rate at a single node in a 4 nodes wireless network running 802.11 DCF. Generation rate at a node is calculated by including payload of all data frames generated by the node.

is the absolute value. Due to the retransmissions, one generated frame may be sent multiple times, therefore, the average sending rate is higher than the generation rate. Another interesting observation from the overall generation rate is that with the smallest minimum contention window size, the three-TX-node network generates almost the same amount of frame as the four-TX-node network. Consider the high contention caused frame lost in the four-TX-node network, the three-TX-node network delivers more frames, which can be verified by the acknowledged frames in Figure 5.57.

The average number of retransmissions is shown in Figure 5.62. The results show that only a few of retransmissions take place with large minimum contention window size. For example, the number of retransmissions with 32 as the minimum contention window size is less than 0.2 for both the three-TX-node network and the four-TX-node network. Especially for three node network, the retransmission is almost zero with all source rate. It is shown that increasing the minimum contention window size can effectively reduce the retransmissions. Another interesting observation found from Figure 5.62 is that the differences between the three-TX-node network and the four-TX-node network is quite limited. The conclusion may need to be further verified by a large scale network, all tested setups show that the average number of retransmissions depends more on the minimum contention window size than the number of frames in the network. This conclusion can be further verified from Figure 5.50. Except for the single link network, the number of retransmissions are almost the same in heavy traffic scenarios.

As shown in Figure 5.63, for the similar average number of retransmissions per node, the networks with more nodes retransmit more frames. In general, retransmissions generates unnecessary repetitions which waste both the frequency

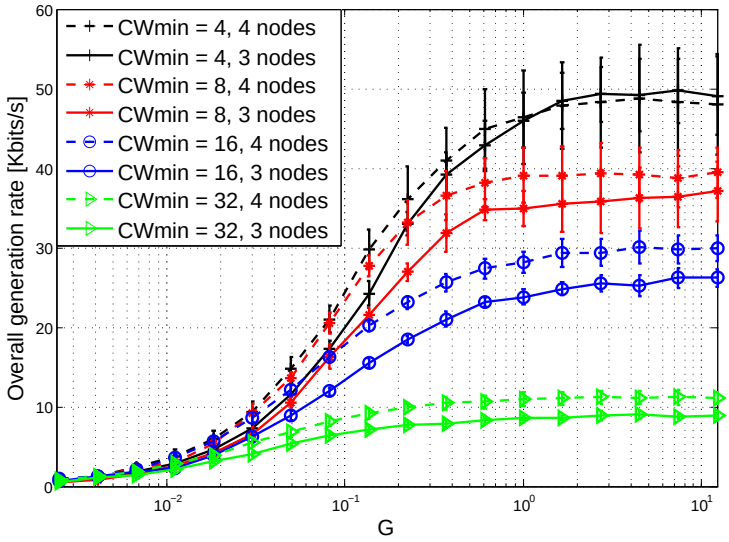


Figure 5.61: Overall generation rate of 4 nodes running 802.11 DCF. Generation rate at a node is calculated by including payload of all data frames generated by the node.

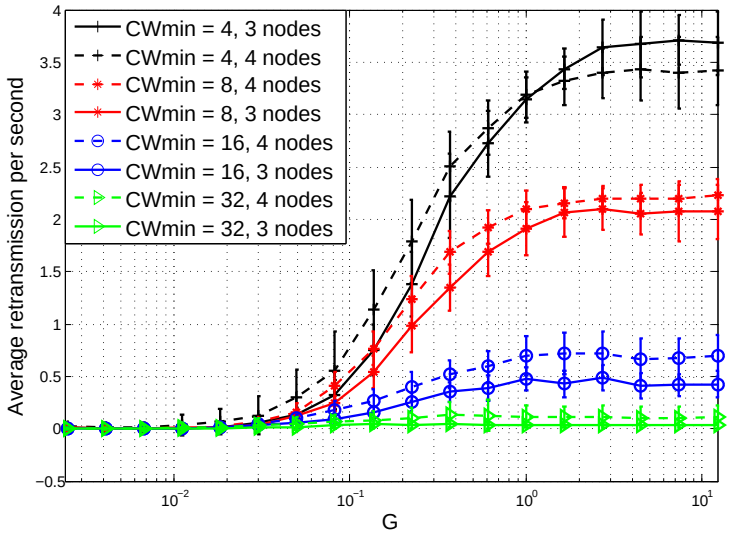


Figure 5.62: Average retransmissions per second at a single node in a 4 nodes wireless network running 802.11 DCF. Retransmissions at a node is only calculated by including retransmitted DATA frames from the node.

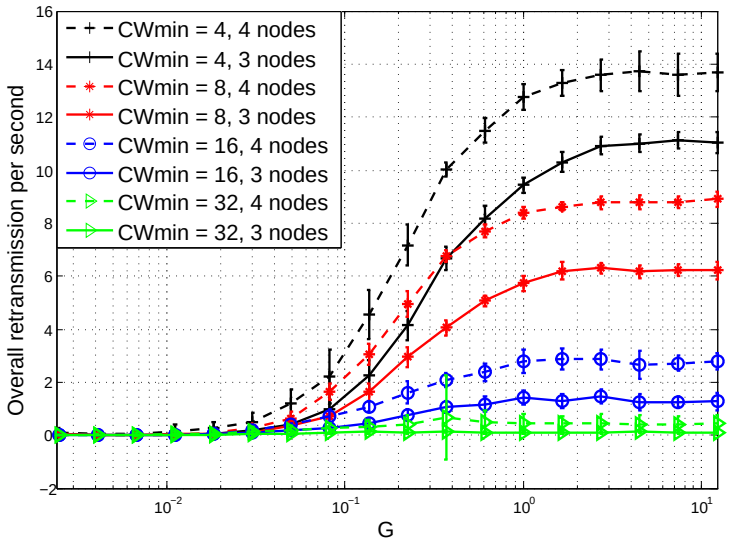


Figure 5.63: Overall retransmissions per second of 4 nodes running 802.11 DCF. Retransmissions at a node is only calculated by including retransmitted DATA frames from the node.

and time resources. This number of retransmission measurements demonstrate that the minimum contention window size could be used to control the number of retransmissions.

If the number of retransmission reaches the retransmission limits which is 6 in the standard and in the measurements, nodes are supposed to discard the current DATA frame and start transmit the next one. The average number of dropped frames is shown in Figure 5.64. It shows that all the networks with minimum contention window size greater than or equal to 16 have almost no frame dropped. For the same minimum contention window size, more nodes drop more frames.

The trend observed from the average number of dropped frames can also found in Figure 5.65. The number of dropped frames increases significantly as the minimum contention window size decreases. Again, this result shows that the minimum contention window size has a significant impact on the number of dropped frames.

Based on the previously recorded data, the successful transmission ratio is calculated and illustrated in Figure 5.66. Similar results are observed from Figure 5.52 that the four-TX-node network with smallest minimum contention window size cannot achieve perfect handshake even with the lowest traffic load. It is also apparent that the minimum contention window size in a network has a significant impact on the successful transmission ratio. For instance, the four-TX-node network has minimum contention window size 4 has 4 times failed ratio then the same network with minimum contention window size 3.

Unlike all above metrics, the round trip time behaves quite differently. As shown in Figure 5.67, the longest round trip time appears in the four-TX-node network with minimum contention window size 8. It is due to the fact that although smaller

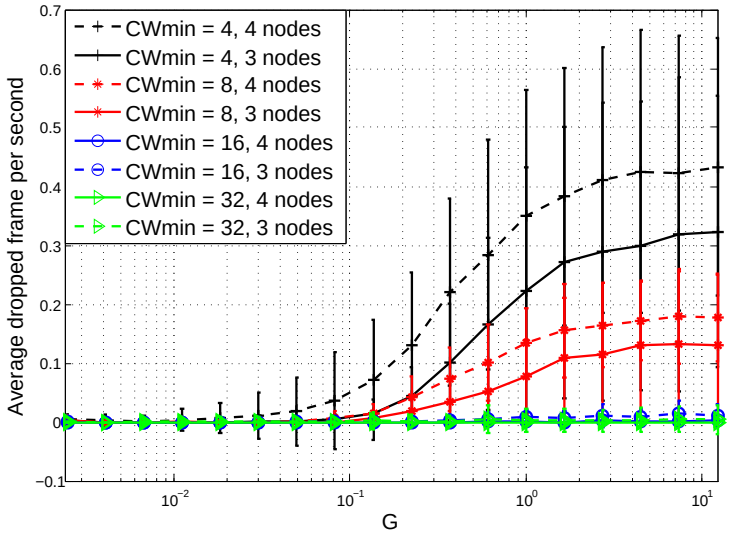


Figure 5.64: Average number of dropped frames per second at a single node in a 4 nodes wireless network running 802.11 DCF. Dropped rate at a node is only calculated by including dropped DATA frames from the node which is caused by exceeding maximal number of retransmissions.

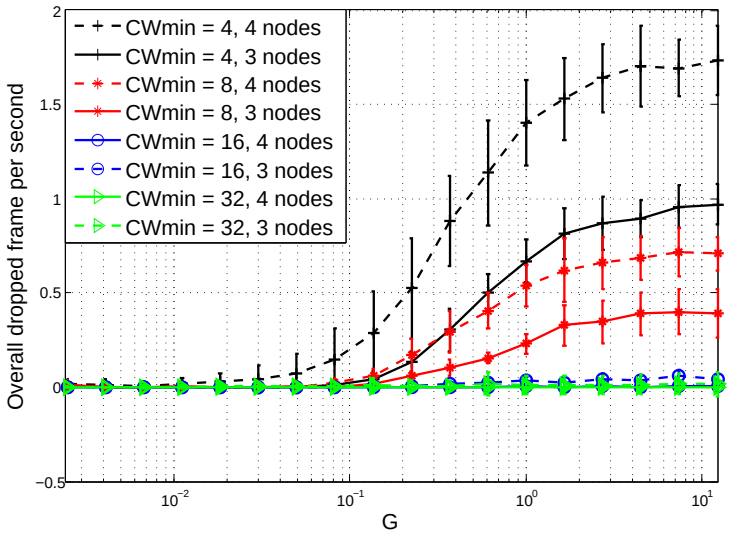


Figure 5.65: Overall number of dropped frames per second in a 4 nodes wireless network running 802.11 DCF. Dropped rate at a node is only calculated by including dropped DATA frames from the node which is caused by exceeding maximal number of retransmissions.

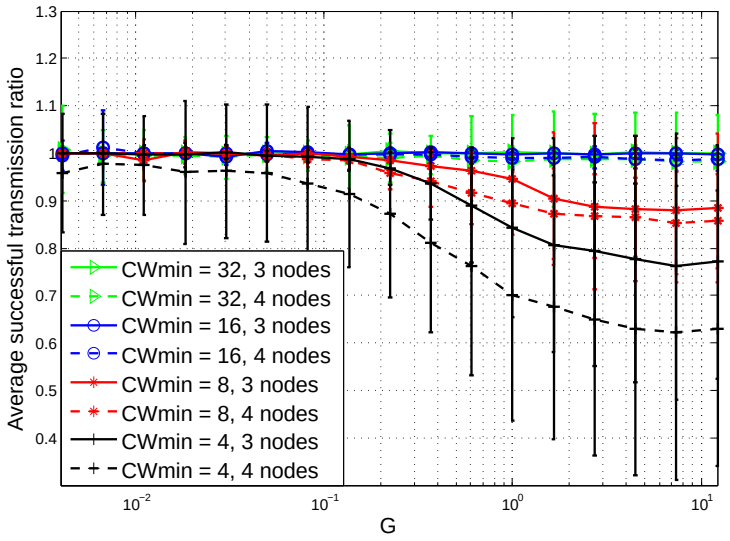


Figure 5.66: Successful transmission ratio of 3 nodes and 4 nodes running 802.11 DCF. Successful transmission ratio at a node is calculated by the number of correctly acknowledged frames divides sent DATA frames.

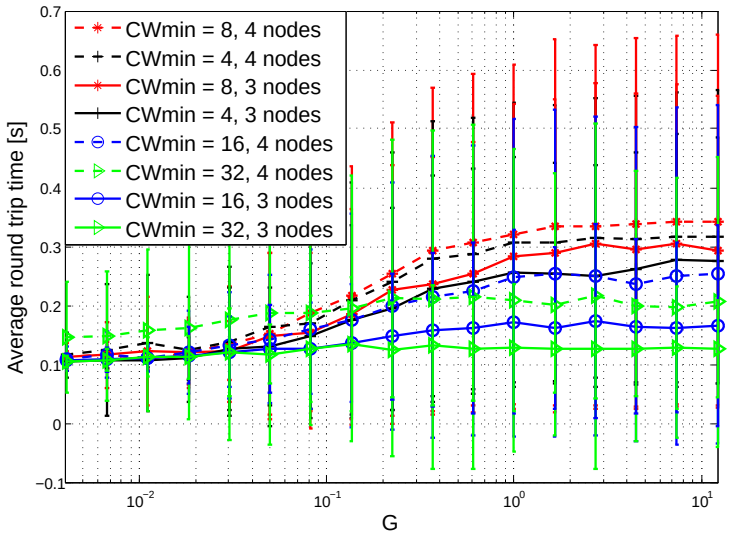


Figure 5.67: Average round time of successfully acknowledged DATA frames of 4 nodes running 802.11 DCF. The round time at a node is calculated by counting the duration between the DATA frame is generated until it is correctly acknowledged.

minimum contention window size leads more backoff, larger minimum contention window size causes longer backoff duration. Therefore, the differences between networks with different minimum contention window sizes are slightly reduced.

The average number of transmission per DATA frame is also recorded and plotted in Figure 5.68. The results agree with the intuition that the DATA frames in

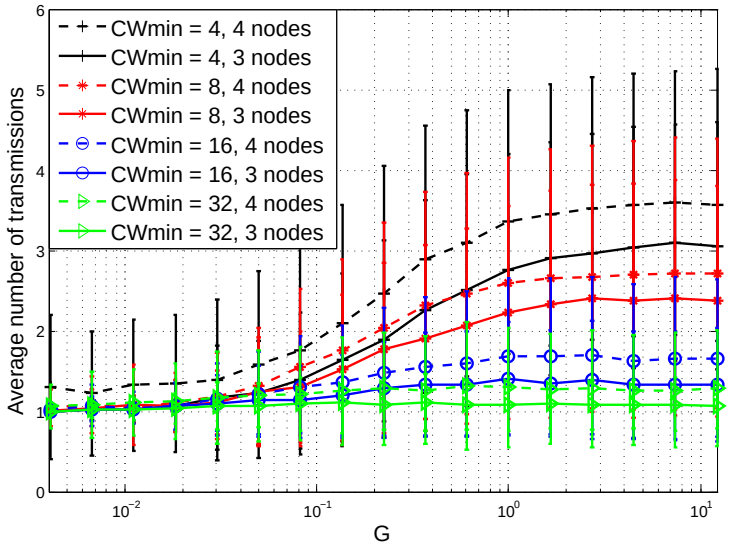


Figure 5.68: Average number of transmissions of DATA frame of 4 nodes running 802.11 DCF. The number at a node is calculated by tracking all generated DATA frames until it is successfully acknowledged or dropped.

the network with more nodes and smaller minimum contention window size are transmitted more times. Morever, the average number of transmissions per DATA frame of the four-TX-node network with minimum contention window size 4 is obviously higher than other networks. It also verified the results of the successful transmission ratio in Figure 5.66 and the round trip time shown in Figure 5.67 that the four-TX-node network with minimum contention window size 4 starts to fail even with the very low traffic thus the longer round trip time is caused.

The average backoff duration can further explain the characteristics of the curves shown in the round trip time measurements. Except the four-TX-node network with minimum contention window size 4, the differences in the average backoff duration of other networks are not as huge as other metrics measured before. It is due to the fact that the network with smaller contention window size may backoff more times, however, each time they backoff shorter durations.

All the measurements in this section show that GR-DMDL is able to handle MAC protocol design, implementation, and measurements as a whole. Due to its HFSM modeling and component based structure, all intermediate steps and variables are easily to track, monitor and recorded. The empirical data show the impact of backoff durations and number of nodes on the system performances. This work can be continuously carried out by optimizing the backoff schemes according to the number of nodes or traffic patterns. DMDL and its GNU Radio-USRP implementation are a suitable SDR platform to perform the measurements and evaluations.

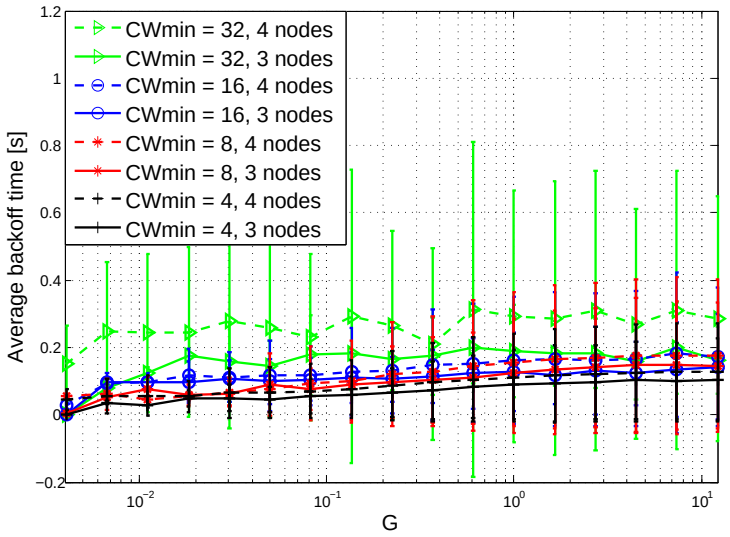


Figure 5.69: Average backoff time of 4 nodes running 802.11 DCF. Backoff time is calculated by averaging the duration of all backoffs.

5.4.5 CogMAC

In contrast to ALOHA and CSMA, cognitive MAC protocols are relatively more complicated due to their advanced technologies in environment awareness, information sharing and PU detection/protection. In order to show the capacity of GR-DMDL on implementing cognitive MAC protocols, we developed CogMAC using GR-DMDL on GNU Radio. We developed CogMAC instead of CogMAC+ here because CogMAC has been implemented in both WARP [48] and LabVIEW with USRP [83]. It is also a good opportunity to compare different SDR platforms which has distinguishing architectures. The GNU Radio implementation of CogMAC are shown in Figure 5.70.

Compared with the hierarchical structure of the LabVIEW implementation, the implementation of CogMAC by GR-DMDL has only one layer thus it is much clearer and simpler. Since we already elaborated the protocol design in detail in Chapter 3, only a short introduction of the implementation is given here. As marked by the numbers in Figure 5.70, the whole implementation can be divided into the following parts,

- ▷ Area 1: generates the repetitive frames for the multichannel preamble reservation.
- ▷ Area 2: controls the whole measurements and channel the testing parameters, which is the number of multiframes in the current measurements.
- ▷ Area 3: creates the complete transmitting frame sequence including both the multiframe transmission and all remaining frames.

- ▷ Area 4: stores the transmitting frame sequence and pop frames one by one. By counting the number of popped frames, triggers the CCA2 carrier sensing after the corresponding frame.
- ▷ Area 5: performs CCA2 carrier sensing.

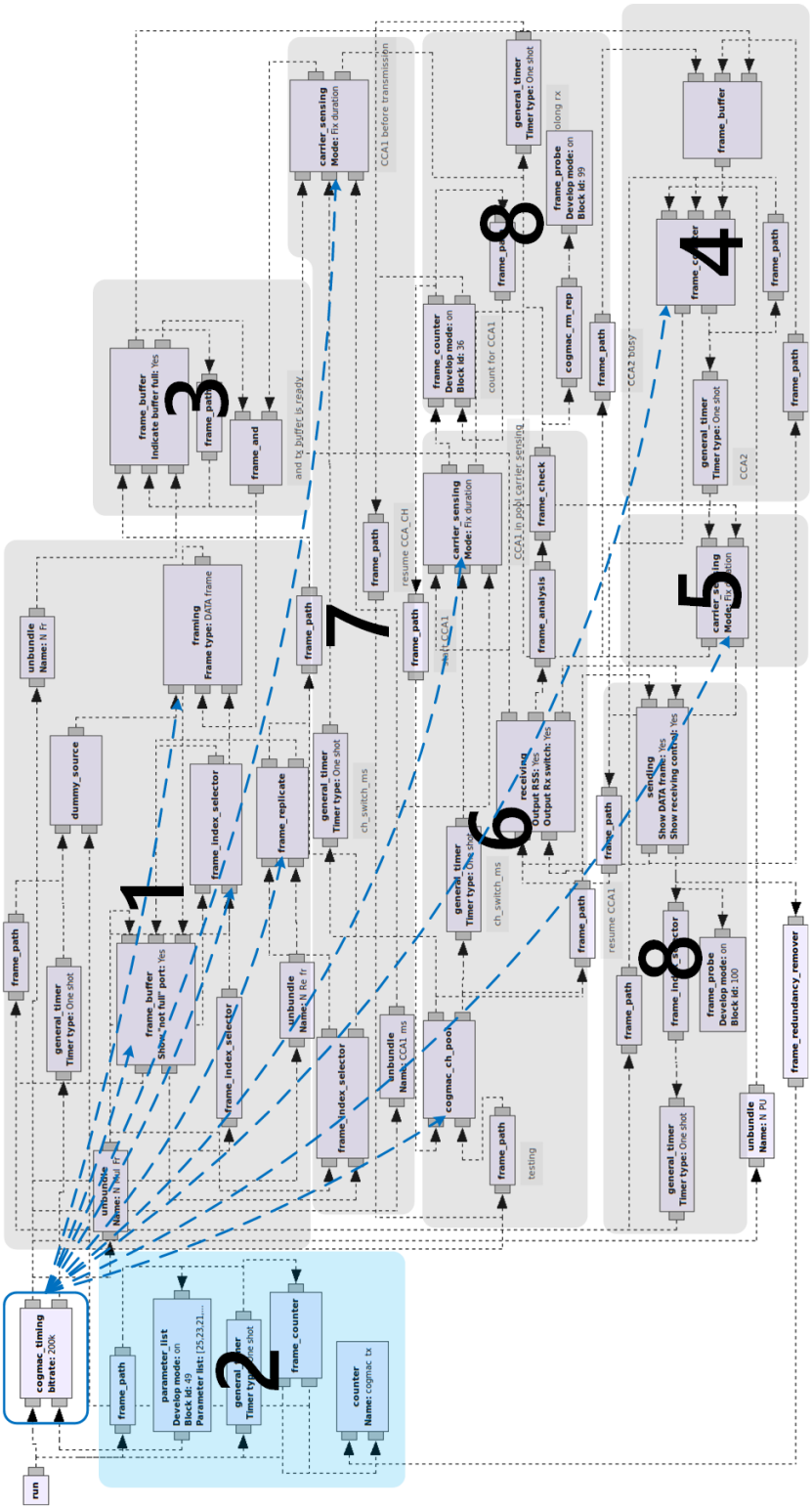


Figure 5.70: Implmentation of CogMAC in GNU radio by using GR-DMDL.

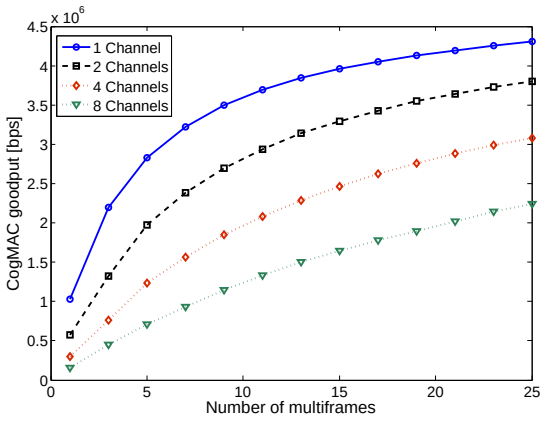
- ▷ Area 6: carries out pool sensing and prepares for CCA1 transmission if new transmitting frame sequence is coming.
- ▷ Area 7: performs CCA1 carrier sensing before transmission.
- ▷ Area 8: Performs reception and other receiving related functionalities.
- ▷ **cogmac_timing** block and the connections: the block calculates all mathematics of the protocol and export the results to all related blocks, e.g., the **frame_index_selector** block in Area 1 which controls the generation of repetitive frames and **carrier_sensing** block in Area 6 which reset the carrier sensing duration of CCA1. This way, the protocol is able to be reconfigured on-the-fly.

Area 1, Area 3, Area 4, Area 5, Area 6, Area 7, and Area 8 are the core code of CogMAC. The GR-DMDL blocks in Area 2 set the durations and change the experimental parameters. All CogMAC timing parameters are updated in the **cogmac_timing** block which is surrounded by the blue rectangle. The new values are sent to blocks which are the destination of the dashed line in Figure 5.70.

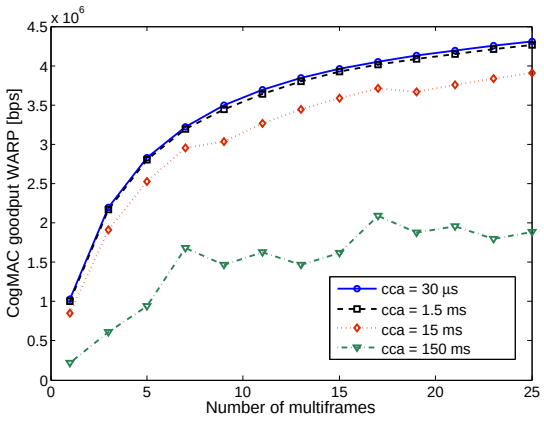
To achieve a fair comparison between the protocol implementations on the three platforms, we repeated the two baseline measurements in Subsection 3.4.1, i.e., the measurement of goodput with respect to different channel pool size and the measurement of goodput with respect to CCA2 durations. The results are shown in Figure 5.71(e) and Figure 5.71(f) together with the results from measurements using WARP and LabVIEW-USRP2 which has been shown in Figure 3.10. Results from all three platforms show the same trends that as the number of multiframes increases, the goodput increases. However, the curves from the USRP based measurements, i.e., the curves in Figure 5.71(c) to Figure 5.71(f), show more details of the relation among the goodput, the number of repetitive transmssions, and the number of multiframes. In contrast to the smooth curves in Figure 5.71(a) and Figure 5.71(b), the curves from the USRP based measurements are more like connected line segments. It is due to the fact that in CogMAC, CCA1 repeatedly interrupts repetitive transmission so that the goodput curves should behave like a step function where the jumps take place when one more CCA1 is performed as the number of frames increases. If we just compare the two USRP based measurements, the effect is more significant in the GR-DMDL based measurements. It is due to the fact that we implemented all GR-DMDL blocks in C++ which is faster and more accurate then LabVIEW.

5.5 Summary

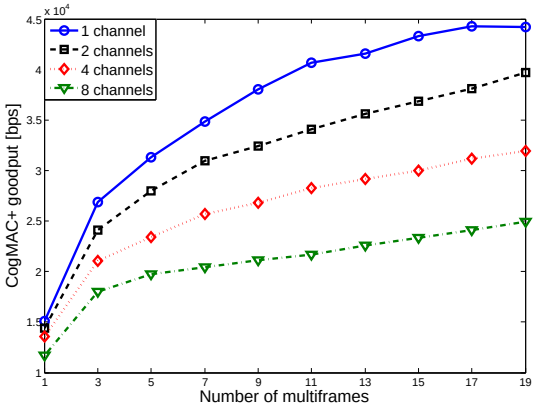
In this chapter, GR-DMDL, the DMDL implementation on GNU Radio is demonstrated, explained, and measured. The elements of DMDL, the MAC function blocks, are first implemented and elaborated in detail. The timing specifications of the system are well measured and discussed. We also implement several well



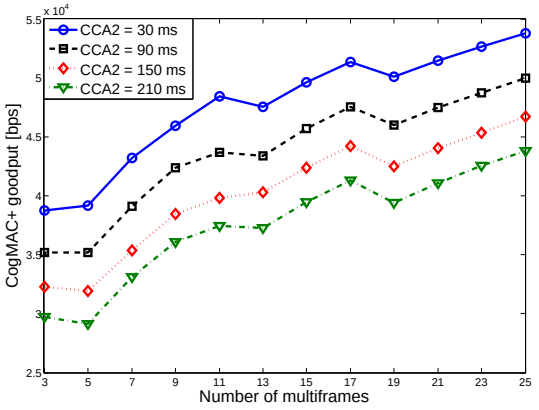
(a) Achieved goodput on WARP broad with respect to different channel pool sizes. [48]



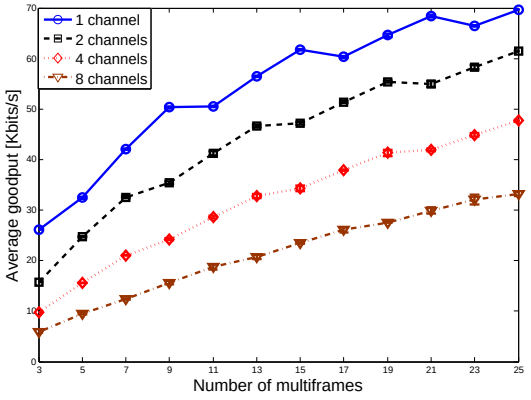
(b) Achieved goodput on WARP with respect to four different CCA2 durations. [48]



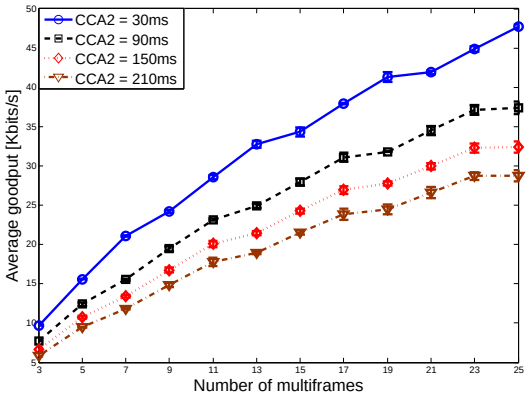
(c) Achieved goodput on LabVIEW-USRP2 platform with respect to different channel pool sizes. [83]



(d) Achieved goodput on LabVIEW-USRP2 platform with respect to four different CCA2 durations. [83]



(e) Achieved goodput by the GR-DMDL implementation on GNU Radio-USRP2 platform with respect to different channel pool sizes.



(f) Achieved goodput by the GR-DMDL implementation on GNU Radio-USRP2 platform with respect to four different CCA2 durations.

Figure 5.71: Baseline comparison of the achieved goodput of one transmitter-receiver pairs running CogMAC protocol.

know MAC protocols to evaluate GR-DMDL. The empirical data of ALOHA and p-persistent CSMA agrees well with the theoretical models, which shows that GR-DMDL is an accurate tool for verification and validation. In the later part of the chapter, an IEEE 802.11 DCF like protocol and CogMAC are implemented. We carry out a number of long term measurements with multiple nodes to show the capacity of GR-DMDL in conducting large scale and long time measurements. The results show that GR-DMDL is able to trace the execution of a MAC protocol comprehensively, which is ideally for investigating the MAC protocols in detail. The implementations themselves also demonstrate that GR-DMDL is able to build complex MAC protocols in an ultra clear and straightforward way.

6 Conclusions and Future Work

The technological progresses in integrated circuit design and manufacturing make the communication systems become increasingly powerful, which will significantly enhance the deployments of artificial intelligence and environmental cognition technologies in future communication systems. The emerging technology will provide an unprecedented capacity and robustness to the next generation of communication systems. However, it will also trigger a long term evolution in system design and implementation. The design of new MAC protocols should include more complex logics and smart strategies. To carry out such intricate design and shorten the development circle of MAC layer schemes, the modular-based platform-independent design method will be widely adopted in the development. Moreover, intelligence and cognitive approaches in wireless network design will gain significance to handle complex radio environments, where both cooperation and competition exist.

This dissertation introduces a number of new protocols and methods such CoopDMAC, SBMAC, CTMAC, and CogMAC+ to offer reliable wireless services by deploying new technologies such as multichannel transmission and directional mm-Wave communications. Many of the proposed protocols are implemented and evaluated in different hardware platforms which provide realistic traffic, spectrum and network conditions. We also propose a platform-independent MAC description language that based on the previous experiences. The main results proposed in this dissertation are summarized in Section 6.1. Potential future work are outlined in Section 6.2.

6.1 Summary of the Key Results

Directional communication technology has a great potential in wireless networks due to its substantial performance gain in increasing SNR of peer communications and its ability of significantly reducing the interference. In the case of high-frequency systems, such as emerging 60 GHz networks, it is an inherent feature of the system design. However, applying directional communications in wireless networks adds more complexity for MAC designer to handle. For example, the interference in directional wireless networks depends on the topology of the networks and the relative positions of involved nodes. Therefore, cooperation and information exchanging become necessities in directional networks, in particular for distributed network where no center control exists. In this dissertation, we analyze the new emerging requirements in directional networks. The directional carrier sensing is investigated and a novel Dual Directional Carrier Sensing (DDCS) scheme is proposed for not only sensing the direction of the following transmission, but also sensing the opposite direction to detect potential interference as widely as possible. The theoretical analysis also showed that even if the scheme is performed, the

probability of collision in directional networks is still higher than in conventional omni-directional networks. Therefore, we proposed the Idle Listening - Directional Medium Reservation scheme to perform information exchanging and topology acquisition. With the help of the network topology, directional nodes are able to perform handshake to improve the knowledge of network topology and share the temporal and spatial information about the directional channel occupation and further avoid collision. Mobility is also addressed in our work by estimating the relative speed of nodes and steering the beam of the antenna according to the changes of the network topology. With a detailed frame structure design, all aforementioned schemes are jointly tested in our customized simulator.

For centralized directional networks, the focus of this dissertation is on analyzing and improving existing industry standard, namely the IEEE 802.11ad standard. To fully utilize the benefits brought by the directivity, we propose Spatial Buffering MAC (SBMAC) to enhance the uplink data rate of the IEEE 802.11ad DMG STAs with low SNR by using relay STAs to buffer and forward their traffic. Thanks to the directional interference, the buffering transmissions can be carried out in parallel with the scheduled uplink transmissions if the interference level is low enough. Therefore, we are able to let the DMG STAs with low uplink data rate to send data to the DMG STA which have higher uplink data rates when other high data rate DMG STA is transmitting to the center STA. In other word, the center STA only communicates with DMG STAs with relatively high date rate. Due to the significant data rate gap between different MCSs of the IEEE 802.11ad standard, the average data rate can be improved, and the delay of the network can be decreased. The idea of SBMAC is verified on a hardware platform in the 60 GHz bands with LDPC coding which is the default coding scheme of the IEEE 802.11ad standard. Based on the measurement results, we carried out simulations including many DMG STAs. The simulation results prove that SBMAC outperforms the original scheduling method proposed by the standard up to 3-fold. Besides the spatial reuse, we also proposed Cross Talk MAC (CTMAC) to further improve the efficiency and throughput of directional wireless networks. By jointly considering the benefits of directional transmission and frame aggregation, CTMAC enables parallel transmissions even under the strongest interfered topologies. By using a packet level scheduling algorithm with an adaptive frame aggregation scheme, CTMAC completely avoids the interference between mainlobes and optimizes the data transmission. The simulation results show that CTMAC outperforms the conventional scheme up to 2-fold with good antenna directivity.

Cognitive wireless networks are an attractive solution to increase spectrum utilization efficiency and enable cooperation between different user groups. A typical application scenario is to enable SU to carry out reliable communications in a multichannel environment, where the rendezvous problem is the key issue. In this dissertation, we propose CogMAC+, a decentralized multichannel MAC protocol, which is based on the Multichannel Preamble Reservation scheme and CSMA principle. It achieves parallel transmissions for multiple secondary users by enabling them to estimate the duration of channel occupancy of each other. In order to achieve the optimal sensing schedule, it allows SUs to dynamically access channels

according to the channel occupancy of the PUs and other secondary networks. Moreover, CogMAC+ uses an adaptive energy detection scheme to dynamically set the energy detection threshold according to the carrier sensing status, the false positive detection ratio and the estimated noise level. CogMAC+ has been verified and evaluated empirically in an SDR testbed with realistic spectrum occupancies of licensed users. The results indicate that CogMAC+ is able to carry out reliable parallel communication in multiple channels even in a severely interfered environments.

To accommodate the significantly increasing demands of applications, services and users, the future communication systems are becoming unprecedentedly complex. Moreover, due to the fast-changing radio environments, the speed of new system design and deployments are dramatically increased. Correspondingly, early prototyping and quick proof of concept on hardware platforms are becoming the key features in developing next generation MAC protocols. Therefore, in the second part of the dissertation, we first model MAC layer scheme via a hierarchical approach which interprets MAC protocols as hierarchical finite state machines. In the light of the new MAC model, we propose Docomposite MAC Description Language (DMDL) aimed at providing a unified tool for MAC design and implementation. DMDL is a full graphical MAC description language based on modular principle that decomposes MAC protocols into elementary components following the synchronous data flow model. In DMDL, a MAC protocol is represented as a directed graph with the MAC function components at the vertices. DMDL uses a token-like controlling unit to pass signaling and data elements among MAC function components via the directed connections. The high decomposability and modularity enables DMDL to be a pure graphical language, and the concurrency nature of the synchronous data flow inherently ensures concurrency as a fundamental feature of DMDL. This also enables DMDL to have strong extensibility and maintainability. To demonstrate the advantages of DMDL, we developed GR-DMDL, the DMDL implementation in GNU Radio. The performance of GR-DMDL is measured and calibrated. Several popular MAC protocols were implemented. There performance in terms of throughput shows near perfect match with theory, which indicates that GR-DMDL is an efficient and accurate MAC designing tool with high flexibility and reusability.

6.2 Future Work

The work presented in this dissertation could be potentially extended in a number of directions. For CogMAC+, we plan to update the channel weighting and dynamic channel selection algorithm of CogMAC+. We are investigating the partially observable Markov decision process framework for estimating the channel occupancy in a better way. This sort of predictor could be added to the current CogMAC+ protocol to provide a more efficient channel selection. Other directions include the use of machine learning to study the behaviors of other secondary users and even the primary user. The outcomes can help us to design smart algorithms to dynamically select channel sensing order and CCA1 intervals. We also plan to have later a public

release of the LabVIEW code with an instruction manual.

We also plan to build a SDR based directional MAC testbed which can be used in designing and prototyping novel directional communication systems. The directional antenna can be implemented by using multiple USRPs and an antenna array. We are building this antenna array based digital beamforming system using GR-DMDL. This way, we can implement all proposed directional MAC protocols in SDRs to further verify our simulation results. Thanks to the high portability of SDRs, we also plan to build some small remote control cars carrying SDRs to investigate the performance of antenna array in tracking and localization. It would be a potential research topic due to its bright future in car-car communications and Internet of Things.

The future wireless networks, such as 5G mobile network and new IEEE standards for high speed WLANs, require fast protocol design, early prototyping, and quick proof of concept on hardware platforms to balance the increasingly complexity and design difficulty. Moreover, network and infrastructure softwarization is another hot topic of 5G development which requires strong practical knowledge. The designing philosophy of DMDL is perfectly in accordance with the trends. DMDL can be further developed in several directions. We will implement DMDL on other SDR platforms, which would be helpful to accelerate the prototyping speed on these platforms. Besides, DMDL can be tailored according to the specific system architecture of the targeted platforms, e.g., implementing some of the blocks in hardware if the onboard FPGA is accessible. We also notice that MAC protocols implemented by DMDL have a very clear mathematical description, i.e., a directed graph with a standardized signaling system. Therefore, applying artificial intelligence, machine learning, and neural network based methods directly on the protocol itself is possible. One potential future research direction could be designing a smart network which can evolve from learning and information sharing.

Acronyms

ACK Acknowledgment

ADC Analog Digital Converter

AEDTA Adaptive Energy Detection Threshold Adjustment

AMPDU Aggregated MAC Protocol Data Unit

AMSDU Aggregated MAC Service Data Unit

AP Access Point

A-BFT Association Beamforming Training

BEB Binary Exponential Backoff

BHI Beacon Header Interval

BI Beacon Interval

BTI Beacon Transmission Interval

ATI Announcement Transmission Interval

CA Collision Avoidance

CAV Channel Allocation Vector

CBAP Contention Based Access Period

CCC Common Control Channel

CCI Co-Channel Interference

CDMA Code Division Multiple Access

COTS Commercial-Of-The-Shelf

CPU Central Processing Unit

CRC Cyclic Redundancy Check

CSMA Carrier Sensing Multiple Access

CTS Clear To Send

DA Destination Address

DAC Digital Analog Converter

DCF Distributed Coordination Function

DCTS Directional Clear To Send

DD Directional-Directional

DDCS Dual Directional Carrier Sensing

DE Discrete Event

DMDL Decomposite MAC Description Language

DNAV Directional Network Allocation Vector

DMG Directional Multi-Gigabit

DMR Directional Medium Reservation

DRTS Directional Request To Send

DTI Date Transfer Interval

DTT Data Transmission Time

DVCS Directional Virtual Carrier Sensing

D-CoopMAC Directional Cooperative MAC

ED Exponential Distribution

EWMA Exponential Weighted Moving Average

FDMA Frequency-Division Multiple Access

FPGA Field Programmable Gate Array

FSM Finite State Machine

GPS Global Positioning System

HED Hyper-Exponential Distribution

HFSM Hierarchical Finite State Machine

HoL Head of Line

HTS Help To Send

IEEE Institute of Electrical and Electronics Engineers

IL Idle Listening

IoT Internet of Things

LR Location Report

LOS Line of Sight

M2M Machine to Machine

MAC Medium Access Control

MCS Modulation and Coding Scheme

MM Mainlobe to Mainlobe

MMT Multichannel Multiframe Transmission

MPDU MAC Protocol Data Unit

MPR Multichannel Preamble Reservation

MSDU MAC Service Data Unit

MSE Mean Square Error

MVCS Multichannel Virtual Carrier Sensing

NFE Noise Floor Estimation

NIC Network Interface Controller

NLR Node Location Request

ns-3 Network Simulator 3

PBSS Personal Basic Service Set

PCP Personal Basic Service Set Control Point

PHY Physical

PLCP PHY Layer Convergence Procedure

PPDU PHY layer Protocol Data Unit

PU Primary User

QPSK Quadrature Phase Shift Keying

RSSI Received Signal Strength Indicator

RTS Request To Send

SA Source Address

SBMAC Spatial Buffering MAC

SDF Synchronous DataFlow

SDR Software Defined Radio

SIFS Short Interframe Space

SINR Signal to Noise Ratio

SIR Signal-to-Interference Ratio

SM Sidelobe to Mainlobe

SNR Signal to Noise Ratio

SP Service Period

SS Sidelobe to Sidelobe

STA Station

SU Secondary User

TDMA Time Division Multiple Access

TRS Transmission and Reception Switch

TTDMA Truncated Time Division Multiple Access

UDP User Datagram Protocol

UML Unified Modeling Language

USRP Universal Software Radio Peripheral

VI Virtual Instrument

WARP Wireless Open Access Research Platform

WMP Wireless MAC Processor

XFSM eXtended FSM

List of Figures

2.1 A general PBSS of IEEE 802.11ad standard and the Beacon Interval. 9

2.2 Frame format of IEEE 802.11ad standard. (a) frame format of normal MPDU; (b) frame format of AMSDU; (c) frame format of AMPDU. 11

2.3 Proof of Theorem1. 13

2.4 Sensed area of DDCS scheme. 16

2.5 An example of two pairs of directional nodes are communicating concurrently. 18

2.6 Frame structure of RTS frame and RTR frame. 19

2.7 Frame structure of DRTS frame and DCTS frame. 20

2.8 A complete transmission cycle of Coop-DMAC in time domain. . . . 21

2.9 Cumulative goodput with respect to the beamwidth of Coop-DMAC and 802.11ad CBAP. The number of multiframes of Coop-DMAC and the length of A-MPDU of 802.11ad are all fixed to 5. 23

2.10 Cumulative goodput with respect to the beamwidth of the directional antennas. Nodes are allocated at the vertices of the corresponding n-sides regular polygons whose circumscribed circles have the same radius of 4 m. Nodes are transmitting in pairs. The number of multiframes is fixed to 20. 24

2.11 Cumulative goodput with respect to the non-interferable factors. Six transceivers are allocated according to the non-interferable factors. The number of multiframes is fixed to 5. 25

2.12 Cumulative goodput with respect to the speeds of nodes. Four transceivers are moving in uniform circular motion of radius 1 m. The center of their orbits are at the vertices of a square with side 8m. The number of multiframes is fixed to 20. 26

2.13 Instant goodput with respect to the executed time. Node A, node B, and node C are fixed at points (−4 m, 2 m), (−4 m, −2 m), and (4 m, 0 m), respectively. Node D is initially located at point D_0, (1 m, −3 m). The beamwidths of all nodes are set to arctan $\frac{8}{15}$. Node D is supposed to move along the x-axis at speed 2 m/s. The number of multiframes is fixed to 5. 27

2.14 A example of a modified IEEE 802.11ad PBSS when SBMAC is applied. Parallel transmissions and the new intervals are also shown. 29

2.15 Experimental setup including one transmitter, one receiver and one interferer. 33

2.16 Coded BER and raw BER of IEEE 802.11ad OFDM PHY MCSs. One transmitter, one receiver and one interferer are used in this measurement. The transmitter and the interferer are allocated close by and turned in the direction of the receiver so the receiver has the LOS connections to both of them. The distances between the receiver and the transmitter/interferer are 3 m. SINRs are changed by changing the gain of the transmitting power of the interferer. . . 35

2.17 Maximal achievable MCSs vs. angle of interferer at the receiver in azimuth plane are measured. One transmitter, one receiver and one interferer are used in this measurement. The distances between the receiver and the transmitter/interferer are 3 m. The transmitter is allocated in the direction of 0°, and the interferer is allocated in the direction of θ. During the measurement, the transmitter and the receiver are always facing towards each other. 36

2.18 Uplink data rate with respect to the beamwidth of the directional antennas. The data rate of all STAs is 100 Mbps. The directional gain of the antenna is 25 dBi. 60 STAs are randomly allocated in the coverage of one PCP/AP. The directional gain of the antenna is 25 dBi and the beamwidth is 5°. The data rate of all STAs varies from 20 Mbps to 220 Mbps. 36

2.19 Uplink data rate with respect to the beamwidth of the directional antennas. The data rate of all STAs is 100 Mbps. The directional gain of the antenna is 10 dBi. 60 STAs are randomly allocated in the coverage of one PCP/AP. The directional gain of the antenna is 25 dBi and the beamwidth is 5°. The data rate of all STAs varies from 20 Mbps to 220 Mbps. 37

2.20 Uplink data rate with respect to the source data rate of each STAs. 60 STAs are randomly allocated in the coverage of one PCP/AP. The directional gain of the antenna is 25 dBi and the beamwidth is 5°. The data rate of all STAs varies from 20 Mbps to 220 Mbps. 38

2.21 (a) Uplink data rate with respect to the number of STAs. 10 to 110 STAs are randomly allocated in the coverage of one PCP/AP as shown in (b). The directional gain of the antenna is 25 dBi and the beamwidth is 5°. The data rate of all STAs is set to 100 Mbps. The directional gain of the antenna is 25 dBi and the beamwidth is 5°. . 38

2.22 (a) Uplink data rate with respect to the source data rate of each STAs. 10, 30, 70 STAs are randomly allocated in the coverage of one PCP/AP which are composed by two sectors as shown in (b). The directional gain of the antenna is 25 dBi and the beamwidth is 5°. The directional gain of the antenna is 25 dBi and the beamwidth is 5°. 39

2.23 (a) Uplink data rate with respect to the source data rate of each STAs. 10, 30, and 70 STAs are randomly allocated in the coverage of one PCP/AP which are composed by two sectors as shown in (b). The directional gain of the antenna is 25 dBi and the beamwidth is 5°. 40

2.24 Frame format of IEEE 802.11ad standard. (a) frame format of normal MPDU; (b) frame format of AMSDU; (c) frame format of AMPDU. 42

2.25 Three types of interference which are generated by parallel transmissions. (a) sidelobe-sidelobe interference; (b) sidelobe-mainlobe interference; (c) mainlobe-mainlobe interference. . 43

2.26 A typical scenario of mainlobe-mainlobe interference and the enabled parallel transmissions. 45

2.27 (a) Achieved goodput with respect to the antenna radiation efficiency. The beamwidth of the antenna is 30 °, the number of MPDUs in a AMPDU is 20. (b) Achieved goodput with respect to the antenna beamwidth. The beamwidth of the antenna is varying from 15 ° to 120 °. The radiation efficiency is 0.5, and the number of MPDUs in a AMPDU is 20. 47

2.28 (a) Achieved goodput with respect to the number of MPDUs in a AMPDU. The beamwidth of the antenna is 30 °. The radiation efficiency is 0.5, and the number of MPDUs in a AMPDU is varying from 2 to 16. (b) Achieved goodput with respect to the length of MSDU. The beamwidth of the antenna is 30 °. The radiation efficiency is 0.5, and the number of MPDUs in a AMPDU is 20. The length of the MSDU is varying from 990 Bytes to 7920 Bytes. 48

3.1 Functionalities comparisons of CogMAC and CogMAC+ and the major improvement of CogMAC+. 53

3.2 Multichannel preamble reservation scheme. As the two (unsynchronized) receivers detect an ongoing transmission, they stay in the channel f1 until a complete frame is received. 55

3.3 Different TIMEOUT durations in CogMAC+: (a) start sensing in the middle of a frame (b) start receiving during default CCA operations (c) start sensing when channel is interfered or PU transmission is carrying out (d) interference or PU transmission start during default CCA operations (e) the maximal timeout duration. 56

3.4 Different timing parameters. 57

3.5 Relation between two types of carrier sensing durations in CogMAC+. 59

3.6 Channel allocation vector in multichannel virtual carrier sensing scheme for parallel transmissions. Node 2 and Node 3 detect an ongoing transmission from Node 1 to the specified destination Node 2 in channel f1 at t_{2_s1} and t_{3_s}, respectively. Following the multichannel preamble reservation scheme, Node 2 stays in the channel f1 to receive the remaining frames. However, Node 3 temporarily removes channel f1 from its channel pool for the interval $t_{3_f1_CAV}$ and carries out its own transmission in channel f2 to another node. 62

3.7 An example of noise floor estimation and adaptive energy detection threshold adjustment. Node 1 are the only node in the scenario which has two channels, f1 and f2 in its channel pool. The node starts a frame transmission in channel f1 after the first round pool sensing operation. An external transmission starts from $t_{f2_ex_s}$ in channel f2. 64

3.8 One transmitter-receiver pair is used in this experiment for validate NFE and AEDTA. The achieved goodput with respect to different distance between the transmitter and the receiver with AEDTA and fixed energy detection thresholds are compared. The transmitter generating randomized traffic with a fixed frame size of 500 bytes. Fixed CCA2 duration of 30 ms and PU allowance time of 500 ms are used. PU allowance time of is 500 ms. 67

3.9 A snapshot of our LabVIEW-USRP2 testbed. 69

3.10 Comparison of the achieved goodput with respect to different parameters on USRP2 and WARP testbeds. One transmitter-receiver pair is used in these experiments with the transmitter generating saturated traffic with a fixed frame size of 1000 bytes. Fixed CCA2 duration of 30 ms and 30 μs and PU allowance time of 500 ms and 15 ms are used in (a) USRP2 and (b) WARP testbeds, respectively. PU allowance time of 500 ms and 15 ms are used in (c) USRP2 and (d) WARP testbeds, respectively. PU allowance time of 500 ms and 15 ms are used in (e) USRP2 and (f) WARP testbeds, respectively. . 71

3.11 Goodput comparison of (a) two transmitter-receiver pairs scenario and (b) four transceivers scenario with respect to different numbers of multiframes and different carrier sensing durations. All transmitting nodes generate saturated unicast traffic with a fixed frame size of 1000 bytes. Fixed CCA2 duration of 30 ms and PU allowance time of 500 ms are considered. 73

3.12 Achieved goodput with respect to different numbers of multiframes and different PU allowance time. Two transmitter-receiver pairs is used in the experiment with the transmitters generating saturated unicast traffic with a fixed frame size of 1000 bytes. Fixed CCA2 duration of 30 ms is considered. 73

3.13 Achieved goodput with respect to different frame sizes. Four transceiver scenario is considered in the experiment with the transmitter generating saturated unicast traffic. A fixed PU allowance time of 500 ms and a fixed number of multiframes are considered. . . 74

3.14 Achieved goodput with respect to the number of multiframes. Up to 8 transmitter-receiver pairs are considered in the experiemnt as all of the transmitters generating saturated unicast traffic. A fixed PU allowance time of 500 ms and a fixed CCA2 duration 30 ms are considered. The number of transmitter-receiver pairs is changed from one to eight. 75

3.15 Achieved goodput with respect to the number of multiframes. Up to 6 transmitter-receiver pairs are considered in the experiment as all of the transmitters generating saturated unicast traffic. The number of channels is fixed to three in this experiment. A fixed PU allowance time of 500 ms and a fixed CCA2 duration 30 ms are considered. The number of transmitter-receiver pairs is changed from one to six. . . . 76

3.16 Achieved goodput with respect to the number of multiframes. Two transmitter-receiver pairs are considered in the experiment with the transmitter generating saturated unicast traffic. A fixed PU allowance time of 500 ms and a fixed CCA2 duration of 30 ms are considered. Different average sequence loads are changed in the experiment. . . . 77

3.17 Cumulative goodput comparison of CogMAC+ with CogMAC in the presence of external transmission in different channels with different average sequence loads. Two transmitter-receiver pairs are considered in the experiment. Transmitters generating saturated unicast traffic. A fixed PU allowance time of 500 ms and a fixed CCA2 duration of 30 ms are considered. 78

3.18 Achieved goodput of two networks with respect to different numbers of multiframes. Four transmitter-receiver pairs are considered in the experiment, in which two pairs use BPSK modulation, and another two pairs use Quadrature Phase Shift Keying (QPSK). Frame size is fixed to 500 bytes for both of the networks. All transmitters generate saturated unicast traffic to their preassigned receivers. A fixed CCA2 duration of 30 ms and a PU allowance time of 500 ms are considered. 79

4.1 Hierarchical structure of models of the classic CSMA/CA used in the IEEE 802.11 DCF. 85

4.2 FSM of a complete CSMA MAC with both TX and RX functionalities. 87

4.3 Major representations of MAC protocols. 88

4.4 A sample of DMDL *command*. 90

4.5 Detailed structures of *command* used in this section. 91

4.6 A DMDL block. 92
4.7 Ports and their names used in DMDL. 92
4.8 Complete form of DMDL timer. 93
4.9 Generate uniformly and Poisson distributed *command* using oneshot timer. 93
4.10 Syntactic sugar of uniformly and Poisson distributed *command* generation using oneshot timer. 94
4.11 Backoff block. 95
4.12 timeout block. 95
4.13 Using framing block to generate a DATA frame. 96
4.14 Using framing block to generating an ACK frame. 96
4.15 Frame buffer block and its behaviors. 97
4.16 Frame analysis block. 98
4.17 Major frame processing blocks. 100
4.18 The interfacing blocks connecting with PHY layer and the hardware. 100
4.19 Carrier sensing block. 101
4.20 Randomization block. 101
4.21 Command Editor block. 102
4.22 Command Counter block. 102
4.23 Using Command Counter block to pick up frames whose index is the multiple of 3. 102
4.24 Register block. 103
4.25 Dummy Source block. 103
4.26 Command Path block. 103
4.27 Run block as the first block in a MAC which drives the whole implementation. 103
4.28 Using DMDL to design a simple Poisson source with packet arrival rate 10/s and the payload length is 100 Bytes. The source only generates payload for 60 seconds. 104
4.29 Using DMDL to design the p-persistent CSMA. 105
4.30 DMDL implementation of the classic IEEE 802.11 DCF. 107

5.1 The GUI of the **general_timer** block in GRC. 111

5.2 The GUI of **timeout** block in GRC. 112
5.3 The **general_timeout** block. 112
5.4 The GUI of **backoff** block in GRC. 113
5.5 Structure of MAC frame and PHY packet. 113
5.6 The **framing** block and its properties. 114
5.7 Other blocks used for framing. 115
5.8 The GUI of **frame_analysis** block in GRC. 116
5.9 The GUI of **frame_check** block in GRC. 116
5.10 The GUI of **address_check** block in GRC. 117
5.11 The GUI of **frame_index_selector** block in GRC. 117
5.12 The GUI of **frame_type_check** block in GRC. 118
5.13 The **beacon_interpreter** block. 118
5.14 The GUI of **frame_buffer** block in GRC. 119
5.15 The GUI of **sending** block in GRC. 121
5.16 sub-blocks in **sending** block 122
5.17 Details of the **sending** block. 124
5.18 Details of the **receiving** block. 125
5.19 The GUI of **receiving** block in GRC. 126
5.20 The GUI of **carrier_sensing** block in GRC. 127
5.21 The GUI of **dummy_source** block in GRC. 128
5.22 The GUI of **frame_probe** block in GRC. 129
5.23 The GRC code used in frame waveform measurement. 133
5.24 Waveforms of GR-DMDL frame transmission. Frame length is 700 bytes. Modulation is QPSK and Sample per symbol is 4. Sampling rate is 400 KHz. 134
5.25 Mechanisms for avoiding self receiving. 134
5.26 Design and implementation of single frame transmission test in GRC. 136
5.27 Key timing parameters in block level synchronization. 137
5.28 Frame level timing characteristics in single frame transmission measurements. 138
5.29 Delay between nodes. 139
5.30 Delay between blocks in transmitting node. 140

5.31 Design and implementation of two way handshake test in GRC. . . . 141

5.32 Key timing parameters in two way handshake. 142

5.33 Frame level timing characteristics in two-way handshake measurements. 142

5.34 Delay between blocks in the same nodes. 143

5.35 Delay between blocks in different nodes. 144

5.36 The GRC code used in pure ALOHA measurement in which two TX nodes and one RX node are used. 145

5.37 Throughput in pure ALOHA network with two transmitting nodes. . 147

5.38 Normalized throughput in pure ALOHA network with two transmitting nodes. 148

5.39 The GRC code used in slotted ALOHA measurement in which two TX nodes and one RX node are used. 149

5.40 Throughput in slotted ALOHA network with two transmitting nodes. 150

5.41 Normalized throughput in slotted ALOHA network with two transmitting nodes. 150

5.42 The GRC code used in slotted p-persistent CSMA measurement in which two TX nodes and one RX node are used. 152

5.43 Throughput comparison of theoretical and empirical performance of p-persistent CSMA protocols with two transmitting nodes. 153

5.44 MSE of different a values between theoretical and empirical performance of p-persistent CSMA protocols with two transmitting nodes. 154

5.45 GNU Radio implementation of the classic 802.11 DCF flowchart shown in Figure 4.30. 156

5.46 Implementation of the IEEE 802.11 DCF with all experimental setups. 157

5.47 Up link Goodput of networks with different number of client nodes running 802.11 DCF. Goodput is calculated by only counting the payload part of successfully acknowledged frames at each transmitter. Number of nodes is varying from 1 to 4. 159

5.48 Sending rate of networks with different number of client nodes running 802.11 DCF. Sending rate at a node is calculated by including all payload of data frames sent by the transmitter. 160

5.49 Generation rate at networks with different number of client nodes running IEEE 802.11 DCF. Generation rate at a node is calculated by including all payload of all data frames generated by the node. . 160

5.50 Average retransmissions per second at a single node in a 4-node wireless network running 802.11 DCF. Retransmissions at a node is only calculated by including retransmitted DATA frames from the node. 161

5.51 Average number of dropped frames per second at a single node in a 4 nodes wireless network running 802.11 DCF. Dropped rate at a node is only calculated by including dropped DATA frames from the node which is caused by exceeding maximal number of retransmissions. . . 162

5.52 Successful transmission ratio of 4 nodes running 802.11 DCF. Successful transmission ratio at a node is calculated by the number of correctly acknowledged frames divides sent DATA frames. 162

5.53 Average round trip time of successfully acknowledged DATA frames. The round trip time at a node is calculated by counting the duration between the DATA frame is generated until it is correctly acknowledged. 163

5.54 Average number of transmissions of DATA frame of 4 nodes running 802.11 DCF. The number at a node is calculated by tracking all generated DATA frames until it is successfully acknowledged or dropped. 164

5.55 Average backoff time of 4 nodes running 802.11 DCF. Backoff time is calculated by averaging the duration of all backoffs. 165

5.56 Average up link Goodput at a single node in a 4 nodes wireless network running 802.11 DCF. Goodput is calculated by only counting the acknowledged frames at each transmitter. 166

5.57 Overall uplink Goodput of 4 nodes running 802.11 DCF. Goodput is calculated by only counting the acknowledged frames at each transmitter. 166

5.58 Average up link sending rate at a single node in a 4 nodes wireless network running 802.11 DCF. Sending rate at a node is calculated by including all data sent by the transmitter. 167

5.59 Overall sending rate of 4 nodes running 802.11 DCF. Sending rate at a node is calculated by including all data sent by the transmitter. . . 167

5.60 Average generation rate at a single node in a 4 nodes wireless network running 802.11 DCF. Generation rate at a node is calculated by including payload of all data frames generated by the node. 168

5.61 Overall generation rate of 4 nodes running 802.11 DCF. Generation rate at a node is calculated by including payload of all data frames generated by the node. 169

5.62 Average retransmissions per second at a single node in a 4 nodes wireless network running 802.11 DCF. Retransmissions at a node is only calculated by including retransmitted DATA frames from the node. 169

5.63 Overall retransmissions per second of 4 nodes running 802.11 DCF. Retransmissions at a node is only calculated by including retransmitted DATA frames from the node. 170

5.64 Average number of dropped frames per second at a single node in a 4 nodes wireless network running 802.11 DCF. Dropped rate at a node is only calculated by including dropped DATA frames from the node which is caused by exceeding maximal number of retransmissions. . . 171

5.65 Overall number of dropped frames per second in a 4 nodes wireless network running 802.11 DCF. Dropped rate at a node is only calculated by including dropped DATA frames from the node which is caused by exceeding maximal number of retransmissions. 171

5.66 Successful transmission ratio of 3 nodes and 4 nodes running 802.11 DCF. Successful transmission ratio at a node is calculated by the number of correctly acknowledged frames divides sent DATA frames. 172

5.67 Average round time of successfully acknowledged DATA frames of 4 nodes running 802.11 DCF. The round time at a node is calculated by counting the duration between the DATA frame is generated until it is correctly acknowledged. 172

5.68 Average number of transmissions of DATA frame of 4 nodes running 802.11 DCF. The number at a node is calculated by tracking all generated DATA frames until it is successfully acknowledged or dropped. 173

5.69 Average backoff time of 4 nodes running 802.11 DCF. Backoff time is calculated by averaging the duration of all backoffs. 174

5.70 Implmentation of CogMAC in GNU radio by using GR-DMDL. . . . 176

5.71 Baseline comparison of the achieved goodput of one transmitter-receiver pairs running CogMAC protocol. 178

List of Tables

2.1 Simulation Parameters 23

2.2 Simulation Parameters 47

3.1 Some Notations Used in CogMAC+. 54

3.2 PHY/MAC parameter values. 70

5.1 Key parameters used in single frame transmission measurements. . . 136

5.2 Key parameters used classic MAC protocols evaluation. 145

5.3 Key parameters used in IEEE 802.11 DCF measurements. 155

Bibliography

[1] *GNU Radio, http://www.gnu.org/software/gnuradio, Last Visited: 09.07.2018.*

[2] *National Instruments LabVIEW 2012 Release Details, http://www.ni.com/labview/release-archive/2012/, Last Visited: 09.07.2018.*

[3] *Technical Presentations: Potential of Modified Signal Detection Thresholds, https://mentor.ieee.org/802.11/dcn/15/11-15-0300-00-00ax-potential-of-modified-signal-detection-thresholds.pptx, Last Visited: 09.07.2018.*

[4] „IEEE Standard for Definitions of Terms for Antennas“. *IEEE Std 145-2013 (Revision of IEEE Std 145-1993)* (2014), March, pages 1–50.

[5] A. Ali, H.-Q. Wang, H.-W. Lv and X.-M. Chen: „A Survey of MAC Protocols Design Strategies and Techniques in Wireless Ad Hoc Networks“. *Wiley-Blackwell Journal of Communications, 2014.*

[6] A. Girault, B. Lee and E.A. Lee: „Hierarchical Finite State Machines with Multiple Concurrency Models“. *IEEE Transactions on computer-aided design of integrated circuits and systems* 18 (1999), Nr. 6, pages 742–760.

[7] A. Khattab, J. Camp, C. Hunter, P. Murphy, A. Sabharwal and E. W. Knightly: „WARP: a flexible platform for clean-slate wireless medium access protocol design“. *ACM SIGMOBILE Mobile Computing Communications Review* 12 (2008), Nr. 1, pages 56–58. – ISSN 1559–1662.

[8] A. Maltsev: *Path Loss Model Development for TGad Channel Models, https://mentor.ieee.org/802.11/dcn/09/11-09-0553-01-00ad-path-loss-model-development-for-tgad-channel-models.ppt, Last Visited: 09.07.2018.*

[9] A. Tanenbaum, and D. Wetherall: *Computer Networks.* Pearson, 2011.

[10] A. Valmari: „The State Explosion Problem“. *Lectures on Petri nets I: Basic models* (1998), pages 429–528.

[11] A.D. Domenico, E.C. Strinati and M.D. Benedetto: „A Survey on MAC Strategies for Cognitive Radio Networks“. *IEEE Journal of Communications Surveys & Tutorials* 14 (2012), Nr. 1, pages 21–44.

[12] B. Ginzburg and A. Kesselman: „Performance Analysis of A-MPDU and A-MSDU Aggregation in IEEE 802.11n“. *Proceedings of the IEEE Sarnoff symposium* IEEE, 2007, pages 1–5.

[13] B.-L. Chen, J.-M. Chen, Y. Gao and J. Zhang: „Coexistence of LTE-LAA and Wi-Fi on 5 GHz with Corresponding Deployment Scenarios: A Survey“. *IEEE Communications Surveys & Tutorials* 19 (2017), Nr. 1, pages 7–32.

[14] B. Lee and E.A. Lee: „Hierarchical Concurrent Finite State Machines in Ptolemy“. *Proceedings of the International Conference on Application of Concurrency to System Design* IEEE, 1998, pages 34–40.

[15] C. Cordeiro and K. Challapali: „C-MAC: A Cognitive MAC Protocol for Multi-Channel Wireless Networks“. *Proceedings of the IEEE International Dynamic Spectrum Access Networks Symposium.* Dublin, Ireland, 2007.

[16] C. Cordeiro, K. Challapali, D. Birru and S. Shankar: „IEEE 802.22: The first world-wide wireless standard based on cognitive radios“. *Proceedings of the IEEE Symposium on New Frontiers in Dynamic Spectrum.* Baltimore, USA, 2005.

[17] C. Cormio and K. Chowdhury: „A survey on MAC protocols for cognitive radio networks“. *Elsevier Journal of Ad Hoc Networks* 7 (2009), Nr. 7, pages 1315–29. – ISSN 1570–8705.

[18] C. Hemanth and T.G. Venkatesh: „Performance Analysis of Contention-Based Access Periods and Service Periods of 802.11ad Hybrid Medium Access Control“. *IET Networks* 3 (2013), Nr. 3, pages 193–203.

[19] C. Shin, H. Park and H.M. Kwon: „PHY-Supported Frame Aggregation for Wireless Local Area Networks“. *IEEE Transactions on Mobile Computing* 13 (2014), Nr. 10, pages 2369–2381.

[20] D. Halperin, W.-J. Hu, A. Sheth and D. Wetherall: „Predictable 802.11 packet delivery from wireless channel measurements“. *ACM SIGCOMM Computer Communication Review* 41 (2011), Nr. 4, pages 159–170.

[21] D. Harel: „Statecharts: A Visual Formalism for Complex Systems“. *Science of computer programming* 8 (1987), Nr. 3, pages 231–274.

[22] E. Nanda, U. Joshi, V. Ribeiro and H. Saran: „RODMAC: A RObust and Distributed MAC protocol for efficient use of white spaces“. *Proceedings of the International Conference on Communication Systems and Networks.* Bangalore, India, 2013.

[23] E. Shihab, L. Cai, and J.-P. Pan: „A distributed asynchronous directional-to-directional MAC protocol for wireless ad hoc networks“. *IEEE Trans. on Vehicular Technology* 58 (2009), Nr. 9, pages 5124–5134.

[24] E.A. Lee: „Modeling Concurrent Real-Time Processes Using Discrete Events“. *Springer Journal on Annals of Software Engineering* 7 (1999), Nr. 1, pages 25–45.

[25] E.A. Lee: „The Problem with Threads“. *Journal Computer* 39 (2006), Nr. 5, pages 33–42.

[26] Ettus Research: *Universal Software Radio Peripheral (USRP), http://www.ettus.com, Last Visited: 09.07.2018.*

[27] F. Cremona, M. Lohstroh, D. Broman, S. Tripakis and E.A. Lee: „Hybrid Co-Simulation: It's About Time“/ Electrical Engineering and Computer Sciences, University of California at Berkeley. 2017 (UCB/EECS-2017-6). – Technical Report.

[28] F. Vazquez Gallego, J. Alonso-Zarate, C. Verikoukis and L. Alonso: „A survey on prototyping platforms for the development and experimental evaluation of medium access control protocols“. *IEEE Journal of Wireless Communications* 19 (2012), Nr. 1, pages 74–81.

[29] Flann Microwave: *Antennas: Standard Gain Horn, http://www.flann.com, Last Visited: 09.07.2018.*

[30] G. Bianchi: „Performance Analysis of the IEEE 802.11 Distributed Coordination Function“. *IEEE Journal on selected areas in communications* 18 (2000), Nr. 3, pages 535–547.

[31] G. Bianchi, L. Fratta and M. Oliveri: „Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs“. *Proceedings of the IEEE Intertenational Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2. Taipei, Taiwan, 1996, pages 392–396.

[32] G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli and I. Tinnirello: „MAClets: Active MAC Protocols over Hard-Coded Devices“. *Proceedings of the International Conference on Emerging Networking Rxperiments and Technologies*. Nice, France, 2012, pages 229–240.

[33] GNU Project: *Message Passing, https://gnuradio.org/doc/doxygen/page_msg_passing.html, Last Visited: 09.07.2018.*

[34] GNU Project: *Polymophic Types, https://gnuradio.org/doc/doxygen/page_pmt.html, Last Visited: 09.07.2018.*

[35] GNU Project: *UDP Source Class Reference, https://gnuradio.org/doc/doxygen/classgr_1_1blocks_1_1udp__source.html, Last Visited: 09.07.2018.*

[36] GNU Project: *UHD Interface, https://gnuradio.org/doc/doxygen/page_uhd.html, Last Visited: 09.07.2018.*

[37] H. ElSawy, E. Hossain and M. Haenggi: „Stochastic Geometry for Modeling, Analysis, and Design of Multi-tier and Cognitive Cellular Wireless Networks: A Survey“. *IEEE Comm. Surveys & Tutorials* (2013), Juli, pages 996–1019.

[38] H.-N. Dai, K.-W. Ng, and M.-Y. Wu: „On busy-tone based MAC protocol for wireless networks with directional antennas“. *Wireless personal communications* 73 (2013), Nr. 3, pages 611–636.

[39] H.-N. Dai, K.-W. Ng, M.-L. Li, and M.-Y. Wu: „An overview of using directional antennas in wireless networks“. *International Journal of Communication Systems* 26 (2013), Nr. 4, pages 413–448.

[40] H. Park, Y. Kim, T. Song and S. Pack: „Multiband Directional Neighbor Discovery in Self-Organized mmWave Ad-Hoc Networks“. *IEEE Transactions on Vehicular Technology* 64 (2015), Nr. 3, pages 1143–1155.

[41] H. Takagi and L. Kleinrock: „Throughput Analysis for Persistent CSMA Systems“. *IEEE Transactions on Communications* 33 (1985), Nr. 7, pages 627–638.

[42] Hamdaoui, Bechir and Shin, Kang G: „OS-MAC: An efficient MAC protocol for spectrum-agile wireless networks“. *IEEE Transactions on Mobile Computing* 7 (2008), Nr. 8, pages 915–930.

[43] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran and S. Mohanty: „A survey on spectrum management in cognitive radio networks“. *IEEE Communications Magazine* 46 (2008), Nr. 4, pages 40–48.

[44] IEEE 802.11 Working Group: „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 3: Enhancements for Very High Throughput in the 60 GHz Band“. *IEEE standard 2012.*

[45] IEEE 802.11 Working Group: „ IEEE 802.11ac. 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 4: Enhancements for Very High Throughput for Operation in Bands Below 6 GHz“. *IEEE Std* (2012), pages 1–628.

[46] J. Ansari, N. Perpinias, A. Nahring, P. Mähönen and M. Petrova: „Empirical Characterization of mm-wave Communication Links in Realistic Indoor Scenarios“. *Proceedings of the IEEE Wireless Communications and Networking Conference.* New Orleans, LA, USA, 2015, pages 1799–1804.

[47] J. Ansari, X. Zhang, A. Achtzehn, M. Petrova and P. Mähönen: „A flexible MAC development framework for cognitive radio systems“. *Proceedings of the IEEE Wireless Communications and Networking Conference.* Cancun, Mexico, 2011.

[48] J. Ansari, X. Zhang and P. Mähönen: „A Decentralized MAC Protocol for Opportunistic Spectrum Access in Cognitive Wireless Networks“. *Elsevier Journal of Computer Communications* 36 (2013), Nr. 13, pages 1399 – 1410.

[49] J. Arnold, L, Simic, M. Petrova and P. Mähönen: „Demo: Spectrum-agile mm-wave packet radio implementation on USRPs“.

Proceedings of the 2015 Workshop on Software Radio Implementation Forum ACM, 2015, pages 5–8.

[50] J.-C. Jia,Q. Zhang and X.-M. Shen: „HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management“. *IEEE Journal on Selected Areas in Communications* 26 (2008), Nr. 1, pages 106–117.

[51] J. Lu, D. Steinbach, P. Cabrol, P. Pietraski and R.V. Pragada: „Propagation Characterization of an Office Building in the 60 GHz Band“. *Proceedings of the European Conference on Antennas and Propagation.* Hague, Netherlands, 2014, pages 809–813.

[52] J. Marinho and E. Monteiro: „Cognitive radio: survey on communication protocols, spectrum decision issues, and future research directions“. *Springer Journal of Wireless Networks* 18 (2012), Nr. 2, pages 147–164.

[53] J. Park, P. Paweczak and D. Cabric: „Performance of joint spectrum sensing and MAC algorithms for multichannel opportunistic spectrum access ad hoc networks“. *IEEE Transactions on Mobile Computing* 10 (2011), Nr. 7, pages 1011–1027.

[54] J. Polastre: „Sensor network media access design“/ EECS Department, University of California Berkeley, USA, 2003. 2003. – Technical Report.

[55] J.L. Peterson: *Petri Net Theory and the Modeling of Systems.* Prentice Hall, 2011.

[56] K.-G. Bian and J.-M. Park: „Maximizing Rendezvous Diversity in Rendezvous Protocols for Decentralized Cognitive Radio Networks“. *IEEE Transactions on Mobile Computing* 12 (2013), Nr. 7, pages 1294–1307.

[57] K. Gilles: „The Semantics of a Simple Language for Parallel Programming“. *Elsevier Journal on Information Processing* 74 (1974), pages 471–475.

[58] K. Haneda, J. Järveläinen, A. Karttunen, M. Kyrö and J. Putkonen: „Indoor short-range radio propagation measurements at 60 and 70 GHz“. *Proceedings of the European Conference on Antennas and Propagation.* Hague, Netherlands, 2014, pages 634–638.

[59] K. Havelund and R. Joshi: „Modeling Rover Communication Using Hierarchical State Machines with Scala“. *Proceedings of the International Conference on Computer Safety, Reliability, and Security* Springer, 2017, pages 447–461.

[60] K. R. Chowdhury and T. Melodia: „Platforms and testbeds for experimental evaluation of cognitive ad hoc networks“. *IEEE Communications Magazine* 48 (2010), Nr. 9, pages 96–104.

[61] L. Stabellini: „Quantifying and modeling spectrum opportunities in a real wireless environment“. *Proceedings of the IEEE Wireless Communications and Networking Conference.* Sydney, Australia, 2010.

[62] L, Yin, T.S. Hong, C.X. Liu, X. Yue and H.E. Z: „A reprogramming protocol based on state machine for wireless sensor network“. *Proceedings of the International Conference on Electrical and Control Engineering (ICECE)* IEEE, 2010, pages 232–235.

[63] Lu, Tingyu and Zhao, Chenglin and Zhang, Yongjun and Peng, Xiao: „Optimal Threshold of Welch's Periodogram for Spectrum Sensing Under Noise Uncertainty“. *Proceedings of the Third International Conference on Communications, Signal Processing, and Systems* Springer, 2015.

[64] L.X. Cai, L. Cai, X.-M. Shen and J.W. Mark: „REX: A randomized EXclusive region based scheduling scheme for mmWave WPANs with directional antenna“. *IEEE Transactions on Wireless Communications* 9 (2010), Nr. 1, pages 113–121.

[65] M. D. Silvius, A. B. MacKenzie and C. W. Bostian: „Rendezvous MAC protocols for use in cognitive radio networks“. *Proceedings of the 28th IEEE conference on Military Communications.* Boston, Massachusetts, USA, 2009.

[66] M. Dardaillon, K. Marquet, J. Martin, T. Risset and H.-P. Charles: „Cognitive Radio Programming: Existing Solutions and Open Issues“/ INRIA. 2013 (RR-8358). – Technical report

[67] M. S. Oude Alink, A. B. J. Kokkeler, E. A. M. Klumperink, G. J. M. Smit and B. Nauta: „Spectrum Sensing With High Sensitivity and Interferer Robustness Using Cross-Correlation Energy Detection“. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 3 (2013), Dec, Nr. 4, pages 566–575.

[68] M. Takai, J, et al.: „Directional virtual carrier sensing for directional antennas in mobile ad hoc networks“. *Proc. of the 3rd ACM Int. Symp. on Mobile ad hoc networking & computing*, 2002, pages 183–193.

[69] M. Takata, K. Nagashima, and T. Watanabe: „A dual access mode MAC protocol for ad hoc networks using smart antennas“. *Proceedings of the IEEE Int. Conf. on Communications*, volume 7, 2004, pages 4182–4186.

[70] M. Timmers, S. Pollin,A. Dejonghe, L. Van Der Perre and F. Catthoor: „A Distributed Multichannel MAC Protocol for Multihop Cognitive Radio Networks“. *IEEE Transactions on Vehicular Technology* (2010), pages 446–459.

[71] M.-X. Gong, D. Akhmetov, R. Want and S.-W. Mao: „Directional CSMA/CA Protocol with Spatial Reuse for mmWave Wireless Networks“. *Proceedings of the IEEE Global Telecommunications Conference.* Miami, Florida, USA, 2010, pages 1–5.

[72] N. Abramson: „THE ALOHA SYSTEM: Another Alternative for Computer Communications“. *Proceedings of the joint computer conference* ACM, 1970, pages 281–285.

[73] N. Guo, R.C Qiu, S.S Mo and K. Takahashi: „60-GHz Millimeter-Wave Radio: Principle, Technology, and New Results“. *EURASIP Journal on Wireless Communications and Networking* 2007 (2007), Nr. 1, pages 48–48.

[74] N. Halbwachs: *Synchronous Programming of Reactive Systems*. volume 215. Springer Science & Business Media, 2013.

[75] National Instruments: *LabVIEW, http://www.ni.com/en-us/shop/labview.html, Last Visited: 09.07.2018.*

[76] Network Simulator 3: *Network Simulator 3, https://www.nsnam.org/, Last Visited: 09.07.2018.*

[77] O. Bazan, and M. Jaseemuddin: „A survey on MAC protocols for wireless adhoc networks with beamforming antennas“. *IEEE Communications Surveys & Tutorials* 14 (2012), Nr. 2, pages 216–239.

[78] P. Gallo, F. Gringoli and I. Tinnirello: „On the Flexibility of the IEEE 802.11 Technology: Challenges and Directions“. *Proceedings of the 2011 future network and mobile summit* IEEE, 2011, pages 1–10.

[79] P.-P. Xu and H.-Y. Chu: „A novel link scheduling strategy for concurrent transmission in mmWave WPANs based on beamforming information“. *Proceedings of the IEEE Wireless Communications and Networking Conference*, 2014, pages 1709–1714.

[80] P. Wang: *GR-DMDL: Decomposite MAC Description Language in GNU Radio, https://github.com/joqoko/gr-inets, Last Visited: 09.07.2018.*

[81] P. Wang and M. Petrova: „Cross talk MAC: A directional MAC scheme for enhancing frame aggregation in mm-Wave wireless personal area networks“. *Proceedings of the IEEE International Conference on Communications Workshops*. Kuala Lumpur, Malaysia, 2016, pages 602–607.

[82] P. Wang, J. Ansari, M. Petrova, and P. Mähönen: „Demo: CogMAC+ - A Decentralized Multichannel MAC Protocol for Cognitive Wireless Networks“. *Proceedings of the 2014 ACM workshop on Software radio implementation forum*. Chicago, USA, 2014.

[83] P. Wang, J. Ansari, M. Petrova, and P. Mähönen: „CogMAC+: A Decentralized MAC Protocol for Opportunistic Spectrum Access in Cognitive Wireless Networks“. *Elsevier Journal on Computer Communications* 79 (2016), pages 22–36.

[84] P. Wang, M. Petrova, and P. Mähönen: „Coop-DMAC: A Cooperative Directional MAC Protocol for Wireless Networks“. *Proceedings of the International Symposium on a World of Wireless Mobile and Multimedia Networks*. Boston, USA, 2015, pages 1–9.

[85] P. Wang, M. Petrova, and P. Mähönen: „DMDL: A Hierarchical Approach to Design, Visualize, and Implement MAC Protocols“. *Proceedings of the IEEE Wireless Communications and Networking Conference*. Barcelona, Spain, 2018, pages 1–6.

[86] P.-Y. Ren, Y.-C. Wang, Q.-C. Du and J. Xu: „A survey on dynamic spectrum access protocols for distributed cognitive wireless networks“. *EURASIP Journal on Wireless Communications and Networking* (2012), pages 60.

[87] P.M. Smulders: „Statistical Characterization of 60-GHz Indoor Radio Channels“. *IEEE Transactions on Antennas and Propagation* 57 (2009), Nr. 10, pages 2820–2829.

[88] Q. Chen, J.Q. Tang, D.T.C. Wong, X.M. Peng and Y.G. Zhang: „Directional Cooperative MAC Protocol Design and Performance Analysis for IEEE 802.11 ad WLANs“. *IEEE Transactions on Vehicular Technology* 62 (2013), Nr. 6, pages 2667–2677.

[89] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya: „Using directional antennas for medium access control in ad hoc networks“. *Proceedings of the 8th annual international conference on Mobile computing and networking*, 2002, pages 59–70.

[90] R. Ramanathan: „On the Performance of Ad Hoc Networks with Beamforming Antennas“. *Proceedings of the ACM international symposium on Mobile ad hoc networking computing* ACM, 2001, pages 95–105.

[91] R.C. Daniels, J.N. Murdock, T.S. Rappaport and R.W. Heath: „60 GHz Wireless: Up Close and Personal“. *IEEE Microwave Magazine* 11 (2010), Nr. 7, pages 44–50.

[92] R.J. Weiler, M. Peter, W. Keusgen and M. Wisotzki: „Measuring the Busy Urban 60 GHz Outdoor Access Radio Channel“. *Proceedings of the IEEE International Conference on Ultra-Wideband*. Paris, France, 2014, pages 166–170.

[93] Romaszko, S: „A rendezvous Protocol with the Heterogeneous Spectrum Availability Analysis for Cognitive Radio Ad Hoc Networks“. *Hindawi Publishing Cooperation Journal of Electrical and Computer Engineering* 2013 (2013), pages 1.

[94] R.R. Choudhury, and NF. Vaidya: „Deafness: A MAC problem in ad hoc networks when using directional antennas“. *Proc. of the 12th IEEE International Conference on Network Protocols*, 2004, pages 283–292.

[95] S. Aly and K. Mustafa: „Protocol Verification and Analysis Using Colored Petri Nets“. *Depaul University, 2003.*

[96] S. Debroy, S. De and M. Chatterjee: „Contention Based Multichannel MAC Protocol for Distributed Cognitive Radio Networks“. *IEEE Transactions on Mobile Computing* 13 (2014), Nr. 12, pages 2749–2762.

[97] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng and J.-P. Sheu: „A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks“. *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks.* Dallas / Richardson, TX, USA, 2000.

[98] S. M. Mishra, A. Sahai and R. W. Brodersen: „Cooperative sensing among cognitive radios“. *Proceedings of the IEEE International Conference on Communications.* Istanbul, Turkey, 2006.

[99] S.A. Edwards and E.A. Lee: „The Semantics and Execution of a Synchronous Block-Diagram Language“. *Elsevier Journal Science of Computer Programming* 48 (2003), Nr. 1, pages 21–42.

[100] Sivers IMA: *FC1005V/00 mm-wave upconvertor, https://www.siversima.com/wp-content/uploads/FC1005V00-Data-Sheet.pdf, Last Visited: 09.07.2018.*

[101] S.M. Ross: *Introduction to Probability Models.* Academic press, 2014.

[102] T. R. Newman, S. Shajedul Hasan, D. DePoy, T. Bose and J. H. Reed: „Designing and deploying a building-wide cognitive radio network testbed“. *IEEE Communications Magazine* 48 (2010), Nr. 9, pages 106–112.

[103] T. Yucek and H. Arslan: „A survey of spectrum sensing algorithms for cognitive radio applications“. *IEEE Communications Surveys & Tutorials* 11 (2009), Nr. 1, pages 116–130.

[104] T. Zhao, J. Luo, W. Yan, and X.-M. Li: „DART: A Directional Antenna Based RouTing Protocol for Ad-Hoc Networks with High Mobility“. *Proceedings of the IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, 2012, pages 1–4.

[105] Tinnirello, I ; Bianchi, G ; Gallo, P ; Garlisi, D ; Giuliano, F ; Gringoli, F: „Wireless MAC processors: Programming MAC protocols on commodity Hardware“. *Proceedings of the IEEE International Conference on Computer Communications and Networks.* Orlando, FL, USA, 2012.

[106] T.S. Rappaport, E. Ben-Dor, J.N. Murdock and Y.-J. Qiao: „38 GHz and 60 GHz Angle-Dependent Propagation for Cellular & Peer-to-Peer Wireless Communications“. *Proceedings of the IEEE International Conference on Communications.* Ottawa, Canada, 2012, pages 4568–4573.

[107] X.-Y. Wang, A. Wong and P.-H. Ho: „Extended knowledge-based reasoning approach to spectrum sensing for cognitive radio“. *IEEE Transactions on Mobile Computing* 9 (2010), Nr. 4, pages 465–478.

[108] X.-Y. Zhu, A. Doufexi and T. Kocak: „A performance evaluation of 60 GHz MIMO systems for IEEE 802.11 ad WPANs“. *Proceedings of the IEEE International Symposium on Personal Indoor and Mobile Radio Communications* IEEE, 2011, pages 950–954.

[109] X. Zeng, R. Bagrodia and M. Gerla: „GloMoSim: a Library for Parallel Simulation of Large-Scale Wireless Networks“. *Proceedings of the twelfth Workshop on Parallel and Distributed Simulation, 1998* IEEE, 1998, pages 154–161.

[110] X. Zhang, J. Ansari, G. Yang and P. Mähönen: „Trump: Supporting Efficient Realization of Protocols for Cognitive Radio Networks“. *Proceedings of the IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks* IEEE, 2011, pages 476–487.

[111] X. Zhang, J. Ansari, G. Yang, and P. Mähönen: „TRUMP: Efficient and Flexible Realization of Medium Access Control Protocols for Wireless Networks“. *IEEE Transactions on Mobile Computing* 15 (2016), Nr. 10, pages 2614–2626.

[112] Y.-S. Huang, P.A. Walsh, Y.-H. Li and S.-W. Mao: „A distributed polling service-based MAC protocol testbed“. *International Journal of Communication Systems* 27 (2014), Nr. 12, pages 3901–3921.

[113] Y. Song and J. Xie: „BRACER: A Distributed Broadcast Protocol in Multi-Hop Cognitive Radio Ad Hoc Networks with Collision Avoidance“. *IEEE Transactions on Mobile Computing* 14 (2015), Nr. 3, pages 509–524.

[114] Y. Takatsuka, M. Takata, M. Bandai, and T. Watanabe: „A MAC protocol for directional hidden terminal and minor lobe problems“. *Proceedings of the IEEE Wireless Telcom. Symposium*, 2008, pages 210–219.

[115] Y.-X. Lin and V.WS. Wong: „Frame Aggregation and Optimal Frame Size Adaptation for IEEE 802.11 n WLANs“. *Proceedings of the IEEE Global Telecommunications Conference* IEEE, 2006, pages 1–6.

List of Publications

Journal paper

[1] P. Wang, J.Ansari, M. Petrova and P. Mähönen: „CogMAC+: A Decentralized MAC Protocol for Opportunistic Spectrum Access in Cognitive Wireless Networks“. *Elsevier Journal Computer Communications* 79 (2016), pages 22–36.

Conference contributions

[1] P. Wang, M. Petrova and P. Mähönen: „DMDL: A Hierarchical Approach to Design, Visualize, and Implement MAC Protocols“. *Proceedings of the IEEE Wireless Communications and Networking Conference.* Barcelona, Spain, 2018, pages 1–6.

[2] P. Wang and M. Petrova: „Cross talk MAC: A directional MAC scheme for enhancing frame aggregation in mm-Wave wireless personal area networks“. *Proceedings of the IEEE International Conference on Communications Workshops.* Kuala Lumpur, Malaysia, 2016, pages 602–607.

[3] P. Wang, M. Petrova and P. Mähönen: „Coop-DMAC: A Cooperative Directional MAC Protocol for Wireless Networks“. *Proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks.* Boston, USA, 2015, pages 1–9.

[4] P. Wang, J. Ansari, M. Petrova and P. Mähönen: „Demo: CogMAC+ - A Decentralized Multichannel MAC Protocol for Cognitive Wireless Networks“. *Proceedings of the ACM workshop on Software radio implementation forum.* Chicago, USA, 2014, pages 1–4.

Curriculum Vitae

Peng Wang

Personal Information

Date of birth	18. March 1986
Place of birth	Hunan, China
Nationality	Chinese

Education

10/2008-06/2011	Master of Science in Communication Engineering at RWTH Aachen University, Germany
09/2004-07/2008	Bachelor of Science in Information Engineering at Shanghai Jiao Tong University, China
09/2001 - 07/2004	Changsha the first middle high school, Hunan, China

Work Experience

10/2011-10/2017	Research Associate at the Institute for Networked Systems (INETS) of RWTH Aachen University, Germany, under the supervision of Professor Marina Petrova
02/2011-06/2011	Internship in Radio Protocol and Multi Tech department of Ericsson, Aachen, Germany
05/2009-10/2010	Student Assistant at the Institute for Theoretical Information Technology of RWTH Aachen University