Link layer Designs for Short-range Wireless Access Spanning ISM to mmWave Bands

A Dissertation Presented
by

Ramanathan Subramanian

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Northeastern University
Boston, Massachusetts

April 2017

To my family.

Contents

List of Figures v

List of Tables vii

List of Acronyms viii

Acknowledgments x

Abstract of the Dissertation xi

1 Introduction 1
- 1.1 Link layer prototyping on SDR platforms . . . 1
- 1.2 Dynamic spectrum switching between Millimeter and Terahertz small cells . . . 4
- 1.3 Medium access protocol for mmWave vehicle-to-infrastructure network . . . 7
- 1.4 Thesis Contributions . . . 9
- 1.5 Novelty of the Contributions . . . 11
- 1.6 Outline of the Dissertation . . . 11

2 Systems Implementation of 802.11 WiFi Networks 13
- 2.1 System Architecture Overview . . . 14
- 2.2 Related Work . . . 15
  - 2.2.1 SDR Software Platforms . . . 15
  - 2.2.2 SDR on Heterogeneous Systems . . . 16
- 2.3 State-action based System Design . . . 18
  - 2.3.1 Slot-time synchronized operations . . . 19
  - 2.3.2 Designated Transmitter State Machine . . . 21
  - 2.3.3 Designated Receiver State Machine . . . 23
  - 2.3.4 System Blocks . . . 24
- 2.4 PHY Layer Algorithms . . . 25
  - 2.4.1 RF Front End Algorithms . . . 25
  - 2.4.2 Preamble Detection Algorithms . . . 26
  - 2.4.3 Parameter Selection . . . 27
  - 2.4.4 Same-Frequency Channel Operation . . . 29
- 2.5 MAC Layer Design . . . 29

2.5.1 MAC Overview . . . 29
2.6 Experimental Setup . . . 31
2.6.1 Communications System Toolbox USRP Support Package . . . 32
2.6.2 MATLAB Coder . . . 32
2.7 Experiments and Results . . . 32
2.7.1 Timing DATA Packet Reception at DRx . . . 33
2.7.2 RFFE Block Timing . . . 33
2.7.3 Two Node Performance (1 DTx and 1 DRx) . . . 34
2.7.4 Profile of Time Elapsed in DTx States . . . 36
2.7.5 Three Node Experimental Setup (2 DTxs and 1 DRx) . . . 37
2.7.6 Three Node Performance: Experimental Results . . . 40
2.8 Virtual Carrier Sensing - RTS/CTS Signaling . . . 42
2.9 GUI for the Testbed Software . . . 45

3 Software-Defined Network Controlled Spectrum Switching 47
3.1 Background and Architectural Assumptions . . . 48
3.1.1 THz Channel Model . . . 48
3.1.2 Architectural Assumptions . . . 50
3.2 Dynamic Spectrum Switching and Medium Access Protocol . . . 51
3.2.1 Distance-dependent spectrum switching . . . 51
3.2.2 Uplink/downlink optimization . . . 52
3.2.3 Throughput maximization, packet aggregation, and error recovery . . . 54
3.3 Capacity Modeling . . . 55
3.3.1 Capacity Formulation . . . 55
3.3.2 Case I - mmWave links . . . 56
3.3.3 Case II - THz links . . . 57
3.4 Multi-vehicle Scheduling . . . 59
3.4.1 Scheduling Problem Formulation . . . 60
3.4.2 Explanation of Algorithm 1 . . . 63
3.5 Data Exchange Evaluation . . . 65
3.5.1 Network Scenario . . . 66
3.5.2 Channel Modeling . . . 67
3.5.3 Data Shower Performance Analysis . . . 69
3.5.4 Scheduling Performance Analysis . . . 71

4 Resource Allocation Scheme for Multi-User mmWave V2I Network 73
4.1 Background . . . 73
4.1.1 Issues Specific to mmWave V2I Communication . . . 73
4.1.2 Hybrid beamforming . . . 74
4.2 Related Work . . . 75
4.2.1 User Association Phase . . . 75
4.2.2 Directional MAC Protocols . . . 75
4.2.3 Coherence Time and Coherence Bandwidth . . . 76
4.3 Proposed Design Approach . . . 78
4.4 Multi-user Directional Medium Access Protocol . . . 78

4.5 Resource Allocation for Multi-User mmWave Vehicular Communications 81
4.5.1 Radio Frame Design 81
4.5.2 Resource Block (RB) Allocation 82
4.6 Simulation Environment 85

5 Conclusion **87**

Bibliography **89**

A Proof of Proposition 1 **99**

B Proof of Corollary 1 **100**

C Proof of Proposition 2 **102**

List of Figures

2.1 System Architecture 14
2.2 System Methodology 18
2.3 Transceive Function Behavior as Defined by Operational State 20
2.4 States for the Designated Transmitter (DTx) 22
2.5 States for the Designated Receiver (DRx) 24
2.6 Comparison of Execution Time for 5 Methods of Computing Cross-Correlation . . 27
2.7 CSMA/CA/ACK Timeline Chart - Energy Detection 30
2.8 CSMA/CA/ACK Timeline Chart - Exponential Random backoff and Retransmission 30
2.9 Transceiver Hardware Setup 31
2.10 Process Time per USRP frame at DRx 33
2.11 RFFE block timing using interpreted MATLAB and MEX 34
2.12 Two Node Performance: Packet Error Rate 35
2.13 Two Node Performance: Bi-directional Link Latency 36
2.14 Timeline Breakup of DATA-ACK Packet Exchange at DTx 36
2.15 Timeline Breakup of DATA-ACK Packet Exchange at DTx 37
2.16 Three Node System with 2 DTxs and 1 DRx 38
2.17 MAC Header - DATA packet [1] 39
2.18 MAC Header - ACK packet [1] 39
2.19 Three Node Performance - Packet Error Rate of the Links 41
2.20 Three Node Performance - Bi-directional Link Latencies 41
2.21 MAC Layer Fairness - Averaged Link Latencies 42
2.22 States for the Designated Transmitter (DTx) 43
2.23 States for the Designated Receiver (DRx) 43
2.24 Screen log at DTx 44
2.25 Screen log at DRx 44
2.26 GUI with the important PHY parameters tab selected 46
2.27 GUI with the important MAC parameters tab selected 46

3.1 Network architecture for SDN controlled mmWave/THz connections. 48
3.2 Selecting durations for uplink (A-C) and downlink (C-E) 52

3.3 Protocol overview for the uplink phase when the distance between the mule and the tower antennas is smaller than the THz threshold. The data chunks are labeled with literals, whereas the numbers represent the packet IDs. A similar procedure applies when mmWave is used for data communication. 53
3.4 Google maps showing the suggested route for a vehicle moving from *1 Summer Street* to *451 D St.*. The end-to-end distance is roughly 1.2 miles and the estimated travel time is about 7 minutes, depending on the traffic conditions. The yellow circle represents the mmWave operational distance. 65
3.5 Empirical LoS, NLoS and outage probabilities for a mmWave link at 73GHz as a function of the separation distance between transmitter and receiver. 66
3.6 LoS and Outage probabilities for a THz link at 0.85THz with 0dBm transmitted power as a function of the separation distance between transmitter and receiver. . . 66
3.7 Capacity for a THz link at 0.85THz as a function of the transmitted power and the separation distance between transmitter and receiver. Outage events have been considered. 67
3.8 Capacity achievable by adopting the proposed THz/mmWave mode selection, as a function of the transmitted power in the THz band and the separation distance between transmitter and receiver. 67
3.9 Data Shower Bulk as a function of the minimum separation distance $d_{\min}$ between the transmitter and the receiver and the average mule velocity. Single-way journey between the vehicle, moving with constant-speed along a straight-trajectory, and the tower. 68
3.10 Data Shower Bulk as a function of the average mule velocity. Single-way journey between two towers located at *451 D St.* and *1 Summer Street* and owned by *Markley Group LLC* and *XO Communications*, respectively, through the route suggested by Google Maps. 68
3.11 Vehicle to data center distance as function of time for a single Monte Carlo realization. Minimum separation distance during closest approach is roughly 5 m. Used as input to generate Figure 3.12. 69
3.12 Amount of exchanged data in every time slot by adopting the proposed greedy scheduling algorithm (Algorithm 1). Each switch is identified by the dotted vertical line with an associated index. 69
3.13 Comparing the total exchanged data transferred with the three scheduling approaches. 70

4.1 Directional MAC Protocol operating in three phases. Lock step switching of random start, fixed orientation beam patterns at BS. 79
4.2 Radio Frame Showing the Important PHY Parameters 82
4.3 Resource Block Allocation in the Time-Frequency Grid 83
4.4 mmWave V2I Simulator: BS Serves Multiple Associated Vehicles 86

List of Tables

2.1 Substate Operation Combinations 22
2.2 Important Parameters 27
2.3 Average Goodput for Varying Payload Sizes 41

3.1 Parameter Setting 64

List of Acronyms

ISM Industrial Scientific Medical

FCC Federal Communications Commission

SDR Software Defined Radio

USRP Universal Serial Radio Peripheral

UHD USRP Hardware Driver

PHY Physical

MAC Medium Access Control

DCF Distributed Coordination Function

CSMA Carrier Sense Multiple Access

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

FSM Finite State Machine

DBPSK Differential Binary Phase Shift Keying

DSSS Direct Sequence Spread Spectrum

ACK Acknowledgment

MEX MATLAB Executable

GPL GNU Public License

RTS Request To Send

CTS Clear To Send

PER Packet Error Rate

BER Bit Error Rate

4G 4th Generation

5G 5th Generation

mmWave millimeter wave

THz TeraHertz

LOS Line Of Sight

NLOS Non Line Of Sight

SDN Software-Defined Network

SD-BS Software-Defined-Base Station

BS Base Station

MS Mobile Station

Kbps Kilo bits per second

Mbps Mega bits per second

Gbps Giga bits per second

Acknowledgments

It is a pleasure to acknowledge those who have helped me complete my Ph.D. degree during my time at Northeastern.

I thank my advisor, Prof. Kaushik Chowdhury, for his constant guidance and continued support during the process of the thesis work. He has encouraged me to come up with new ideas, design and systems. His confidence and patience are the primary reasons that I could successfully prototype the 802.11 link layer on the USRPs.

Sincere thanks to Prof. Stefano Basagni and Prof. Miriam Leeser for being on my Ph.D. committee, reviewing my thesis and providing valuable comments. Their feedbacks and suggestions have kept me in the right direction.

I thank MathWorks engineers, Mike McLernon and Ethem Sozer, for their close collaboration and support on the SDR project. I acknowledge the Development Collaborative Research Grant (DCRG) from MathWorks, Inc. for financially supporting me for two years.

I thank Eric Doyle, Benjamin Drozdenko, Rameez Ahmed. I had a great learning experience working with them as a team. I would like to thank them for having been instrumental in demonstrating the 802.11 link layer on the USRPs. Thanks to Shivam Sharma for investing significant time in the development of the Testbed GUI.

I would like to thank my collaborators in research, Prof. Marcello Caleffi and Prof. Sara Cacciapuoti. I gained new theoretical knowledge and practical skills from my research on Vehicle to Infrastructure communication in mmWave and THz bands.

I thank my friends Sarath Shanker, Meenu Swaminathan and Naveen Manikandan and my wife, Kirthana Ganesh Babu, for having been a constant source of motivation and support. I also thank Carlos Bocanegra, Yousof Naderi and, Rahman Doost for having interesting discussions on my research work.

Abstract of the Dissertation

Link layer Designs for Short-range Wireless Access Spanning ISM to mmWave Bands

by

Ramanathan Subramanian

Doctor of Philosophy in Electrical and Computer Engineering

Northeastern University, April 2017

Dr. Kaushik Chowdhury, Advisor

Design and rapid prototyping of new medium access protocols are critical to support networked systems that are progressively reaching higher data rates with flexible use of spectrum bands. The thesis tackles both a systems and protocol fronts, developing a MATLAB-based link layer for the widely used Universal Software Radio Peripheral (USRP) software defined radios as well as designing link layer switching and medium access protocol for next generation millimeter wave (mmWave) and Terahertz (THz) bands.

We first design a state-action based 802.11 standards-compliant finite state machine (FSM) and show its operation on a real network testbed. Using a software-only approach, the user has parameter flexibility for a number of variables as well as options such as classical data-acknowledgement or request/clear-to-send modes. The design also supports other advanced features like modifying the back off window behavior and changing the channel sensing methods. To make the research reproducible and allow for extensibility by the community, the software along with a GUI is made publicly available, released under the GNU Public License (GPL).

Next, we explore a new software-defined network (SDN) framework for vehicles equipped with transceivers capable of dynamically switching between THz and mmWave bands, apart from existing classical LTE cellular bands. A novel SDN controlled admission policy that preferentially handoffs between the mmWave and THz small cells, accommodates asymmetric uplink/downlink traffic, performs error recovery and handles distinct link states that arise due to motion along practical vehicular paths is presented. A polynomial-time scheduling algorithm is designed for scheduling multiple vehicles at a given infrastructure tower, accounting for their cumulative bandwidth needs,

contact times and coordination overheads.

Finally, we design a directional MAC protocol that encompasses a novel resource allocation scheme for the mmWave Vehicle-to-Infrastructure (V2I) network in an urban setting. We specifically consider a network where each Base Station (BS), equipped with hybrid beamforming antenna arrays, concurrently serves multiple vehicles. Using coherence bandwidth and coherence time specific to the mmWave vehicular channel, we provide an optimal resource allocation scheme towards efficient multi-user scheduling.

In summary, this thesis addresses several challenges in the design of medium access protocols for short-range wireless networks that can operate in distinct spectrum band(s). The publicly available code base, protocols, analytical models, algorithms and the insights resulting from simulation-based case studies will help researchers in significantly reducing the development time and effort. This will enable future reliable link layer designs and architecting robust network of radios, paving the way for the emergence of far-reaching wireless applications.

Chapter 1

Introduction

Networked systems will continue to demand progressively higher data rates to support emerging applications and be increasingly capable of flexible use of spectrum bands to deal with spectrum scarcity. In this regard, the design and rapid prototyping of new multiple access techniques is critical for the performance of the associated lower link layer as it has a direct impact on the performance of the higher layers. The recent adoption of rules by FCC in millimeter wave spectrum is indicative of the growing pressure to identify new spectrum that will ease the spectrum scarcity in the already congested ISM bands and the inadequacy of current 4G systems in supporting emerging applications such as self-driving cars, augmented reality that demand much higher data rates and low latencies, and of the potential opportunities in the next generation wireless broadband technologies.

1.1 Link layer prototyping on SDR platforms

Software defined radio (SDR) allows unprecedented levels of flexibility by transitioning the radio communication system from a rigid hardware platform to a more user-controlled software paradigm.

A basic SDR system is composed of a computer connected to a RF front end capable of receiving and transmitting radio signals. A RF front end requires an antenna suited for specified RF bands of interest, a transceiver chip that is comprised of at least one local oscillator, analog-to-digital converter (ADC), and digital-to-analog converter (DAC), and an interface (e.g. Ethernet cable) that connects the front end to the computer. The computer may have a general purpose processor to process the digital output and programs to realize specialized tasks such as filtering, amplification, and modulation, which have traditionally been implemented in hardware. The design concept of

the SDR is advantageous because it reduces the need for special purpose hardware and allows the developer to add new functionality to the radio by modifying the software. The flexibility inherent in the SDR allows for the potential to support many wireless standards, whereas a single hardware transceiver can only support a few or one standard. Hence, the SDR device can be seen as an increasingly affordable alternative towards prototyping new link layers.

Challenges in designing highly customizable SDR platforms

Any modern wireless standard relies on accurate timing to complete the standards-specified tasks. In SDR, as the received and transmitted signals are represented as arrays of data samples collected by the front-end, software processing contributes to delays. Additionally, when multiple nodes operate in a shared channel, timing issues add to the challenge of ensuring synchronized behavior between multiple nodes. In the absence of hardware clocks, the SDR must devise a means of calculating how much time has elapsed, so that transmission and reception functions are performed at the appropriate intervals. The processing functions and their internal parameters must also be open for change, should a better algorithm be designed, or if no set thresholds may be possible, as is the case in highly challenging environments with variable noise floor. Finally, the software running on the SDR must be structured in a hierarchical manner, so that its functionality can be separated into layers that are compliant with the Open Systems Interconnection (OSI) model. Thus, the base drivers that interface with the RF front-end platform should be abstracted from the physical (PHY) layer functionality, which in turn should be abstracted from the medium access control (MAC) layer logic. In summary, there are many design challenges that must be overcome before a highly customizable SDR platform is made available for general purpose use.

There is also a urgent need for reducing programming complexity in current SDR platforms. Significant expertise is required to successfully navigate the hardware design, software implementation, wireless standards requirements, and computational timing limitations, which requires specialized training and lengthens time to project completion. Further, it is time consuming to design and implement such SDRs as they typically require thorough knowledge of the operating environment and a careful tuning of the program.

Realizing a MATLAB-based 802.11 compliant link layer

We first design a state-action based 802.11 standards-compliant finite state machine (FSM) and show its operation on a real network testbed and outline strategies on how to create a such a

design, wherein the same node switches between transmitter and receiver functions. We outline strategies on how to create a state-action based design, wherein the same node switches between transmitter and receiver functions. We then implement the design as a bidirectional transceiver that runs on the commonly used USRP® platform and implemented in MATLAB® using standard tools like MATLAB Coder™ and MEX to speed up the processing steps.

Our design allows optimal selection of the parameters towards meeting the timing requirements set forth by various processing blocks associated with a DBPSK physical layer and CSMA/CA/ACK MAC layer so that all operations remain functionally compliant with the IEEE 802.11b standard for the 1 Mbps specification. The code base of the system is enabled through the Communications System Toolbox™ and incorporates channel sensing and exponential random back-off for contention resolution. The current work provides an experimental testbed that enables creation of new MAC protocols starting from the fundamental IEEE 802.11b standard. Key performance metrics such as packet error rate, bidirectional link latency, and goodput are measured and reported. Our design approach guarantees consistent performance of the bi-directional link, and the three node experimental results demonstrate the robustness of the system in mitigating packet collisions and enforcing fairness among nodes, making it a feasible framework in higher layer protocol design.

In addition, options such as classical data-acknowledgement or request/clear-to-send can be selected. The request/clear-to-send packet exchanges are specified in the standard to to address the hidden terminal problem. This required us to update finite state machine already implemented DATA/ACK functions. We implemented it in such a way that allows the user to pick between the two state machines so as to have either a DATA/ACK exchange or a RTS/CTS/DATA/ACK exchange. Other advanced features like modifying the back off window behavior by allowing for linear/exponential scaling of the contention window and changing the channel sensing methods not limited to energy detection are made available.

Advantages of our approach

The state machine design approach drove code development and enabled modularity of the code base. Using a software-only approach and parametrizing the important variables allowed for full parameter flexibility. This gives the user freedom to reconfigure the parameter values both in the PHY and MAC layer as needed.

To facilitate quick deployment, it includes an initialization script for the setting and tuning of the reconfigurable parameters at the physical layer based on the specific channel measurements at

the chosen experimental site.

To make the research reproducible and allow for extensibility by the community, the software is made publicly available, released under the GNU Public License (GPL). The software also includes a GUI, developed using MATLAB GUIDE, that is ready to be used for over-the-air experimentation. The help file in the GUI specifies the selection criteria for each parameter along with their default values.

Our work provides an experimental testbed that enables creation of new MAC protocols starting from the fundamental IEEE 802.11b standard.

1.2 Dynamic spectrum switching between Millimeter and Terahertz small cells

Small cell densification in urban areas is a cost-effective way to reliably expand network coverage and provide significantly increased capacity for end users [2]. Outdoor small cell deployments are expected to proliferate starting 2017 [3]. While this is advantageous, the spectrum scarcity and congestion problem in the sub-6 GHz bands remain. Small cells that can utilize the available massive spectrum bandwidth in the millimeter-wave (mmWave) (around $30 - 100\,\mathrm{GHz}$) and Terahertz (THz) (around $0.1 - 10\,\mathrm{THz}$) frequencies promise a paradigm shift, in realizing fiber-equivalent wireless links [4], leading up to several Tbps of effective data transfer rates, and further, freeing up the lower bands for macrocell to small cell communications.

The 802.15 THz group report from March 2015 advocates even higher frequencies to 'future-proof' the access technology, where frequencies in the 0.1-10 THz range could be used to achieve several Tbps transmission rates. However, this so called *data shower* is possible only for very short distances of few meters. Propagation in mmWave and THz bands is limited by the severe pathloss and atmospheric absorption. To counteract the significant attenuation, and extend coverage, high directivity gain antennas are used. The links in mmWave bands so formed may be in the range of 200 meters [5] which is considerably longer than the link distance in the THz bands, typically in the order of few meters. The high data rate, limited coverage and reduced interference are attractive features and make these bands an excellent candidate for small cells.

Recent efforts have pointed towards the need of SDN-based resource sharing, by centralizing the physical and medium access control (MAC) functions, along with typical operator tasks of load balancing and admission control policy. We adopt this approach in our work, wherein a

SDN controller helps establish robust communication in mmWave and THz bands, where close to 7 GHz and 100 GHz chunks of contiguous bandwidth are available respectively. Clearly, SDN-based dynamic spectrum switching here enables efficient use of both bands instead of a single constant choice.

Massive vehicle to infrastructure data transfers

We propose a radically different paradigm for massive vehicle-to-infrastructure data transfers using a combination of vehicular networks and wireless, short-range access links composed of ultra-high bandwidth at millimeter (mmWave) and Terahertz (THz) frequencies.

5G calls for vehicles to be equipped with advanced communication capability to facilitate both vehicle to vehicle and vehicle to infrastructure data exchanges, primarily from a viewpoint of enhancing road safety, self-driving cars and for multimedia content sharing. The ultra-high bandwidth available at millimeter (mmWave) and Terahertz (THz) frequencies can effectively realize short-range wireless access links in small cells enabling such use cases. Our network architecture emerges from this vision, with vehicles able to exchange extremely high data rates through their on-board mmWave and THz transceivers.

Reliable and continuous high bandwidth connectivity within the next generation of vehicles will enable driver-less cars with on-the-road infotainment services using bulk media downloads, ultra-fast massive data transfers towards data backhauling and city-scale traffic optimization realized by uploading massive high-rate sensor data to the cloud for processing. Google's self-driving car, for example, generates sensor data at the rate of 750 MBps [6] and automated driving cars are expected to generate in the order of 1 TB of sensor data in a single trip [7]. The sensor data can be used to remotely monitor the current state and predict a potential breakdown of the vehicle. Another potential use case can be to have the vehicles' camera images along with the location information be sent to the cloud for automakers to build detailed and accurate maps [8]. Self-driving cars, which are limited in their sensing range, will greatly benefit from precise maps, downloaded say when connecting to infrastructure, that reflect recent updates to navigate urban areas or the highways. Note that upload/download of such data will demand high throughput but there is no real-time requirement.

The ability to achieve data transfer rates in the order of several gigabits-per-second is key to enable such applications, so far unattainable through state of the art dedicated short-range communication (DSRC) and 4G cellular communication [9].

Data center traffic backhauling

User reliance on cloud-based services have incurred an explosive growth over the past several years, with data centers becoming an integral infrastructural component of several major companies. To improve resilience, multiple data centers may be managed by the same provider, and they are often geographically distributed with content replication. This introduces massive demands on bandwidth consumption as data is moved across these locations [10]. However, this type of data is not interactive, and is majorly composed of bulk transfers that may incorporate delay tolerance [11]. Additionally, capital costs of installing networking equipment that connect these different data centers are a dominant fraction of the overall overhead, depending on both the fiber miles and traffic volumes [12].

Fiber-based backhauling required to connect the small cells at scale to the core network will pose serious deployment challenges in terms of deployment time and wiring expenditure. Wireless backhauling using mmWave links, considered as an alternative solution, will be difficult to come by in urban settings (with trees and buildings of varying heights) given the reduced likelihood of LoS propagation conditions. In that regard, vehicles serving as digital mules will reduce deployment costs of fiber-based backhauling solutions [12, 13]. It is important to note that fiber is expensive and can become congested and using vehicles may aid in bulk transfer of delay-tolerant information between data centers [11]. Further, there are inherent advantages of using vehicles as mobile-data caches. The vehicles are likely to contain region-specific content that can increase localized hits [14].

We investigate this new application in the context of V2I where vehicles are equipped with dual mmWave and THz transceivers for enabling non real-time inter-data center backhauling in urban areas. Here, the vehicles serve as mules that download data from a given center, physically move to the next location and then upload the data, using a mix of THz and mmWave bands. By using vehicles as data mules, the source-destination BSs themselves need not have direct LoS conditions between their individual antennas or incur infrastructural deployment costs. This approach mitigates reliance on physical cabling, and also makes use of near-deterministic vehicular motion that serve as data mules, relaying information between different data centers.

Quantifying the end-to-end data transfer rates involved analytically deriving the resulting capacity of such a small cell network that accounts for the channel characteristics unique to both these spectrum bands, relative distance and the contact times between a given transceiver pair. Careful simulation-based case studies were carried out for the use case of data center backhauling using actual road maps and data center locations within Boston city to showcase the benefits of our approach.

We then formulated the optimal procedure for scheduling multiple vehicles at a given infrastructure tower, with regards to practical road congestion scenarios. The search for the optimal schedule is shown to be a NP-hard problem. Hence, we design a computationally-feasible polynomial-time scheduling algorithm that runs at the SDN controller and compare its performance against the optimal procedure and random access.

1.3 Medium access protocol for mmWave vehicle-to-infrastructure network

Millimeter wave (mmWave) communications is increasingly seen as a means to meet the communication constraints demanded by the emerging Intelligent Transportation Systems (ITSs) applications. In this work, we provide a directional MAC protocol that encompasses a novel resource allocation strategy unique to the mmWave Vehicle-to-Infrastructure (V2I) network in an urban setting. We consider a network where each Base Station (BS), equipped with hybrid beamforming antenna arrays, concurrently serves multiple vehicles.

The BS positioned at the road-side and installed atop, say, the traffic lights, the lamppost, or other road-side infrastructure handles the V2I communications among multiple vehicles. Since vehicles can be making simultaneous access requests, multiple request-to-send (RTS) packets from the MSs can potentially collide at a BS. As each MS transmits independently while being deaf to others' transmissions, collisions are likely to be frequent and has to be embraced.

We identify and address two significant challenges i.e. resolving multiple concurrent access requests and efficient resource allocation, towards realizing robust mmWave V2I communications.

The mmWave channel is identified to possess a sparse nature due to the use of large bandwidths and multiple closely spaced antennas. Thanks to hybrid beamforming techniques, multiple concurrent beams can be realized, and further exploiting the spatial sparsity, the multiple RTS requests can be resolved as each request can be serviced with a distinct RF chain. Post the successful association, the BS must then quickly schedule and serve the associated vehicles.

The time-frequency resource at the BS must be efficiently allocated considering the asymmetric communication requirements of the vehicles. [15] show that time-frequency scheduling is more frequent compared to the spatial scheduling based on a reasonable change in the covariance matrix of the channel. Moreover, hybrid beamforming results only in few beam directions (sparse in space), and so, we can restrict the packing to time and frequency dimension. Using models for

the coherence bandwidth and coherence time specific to mmWave vehicular channel, we design a radio frame structure and provide a resource allocation scheme that the BS utilizes towards efficient multiuser scheduling.

To evaluate our network, we built a channel simulator entirely in MATLAB to carry out the link layer simulations.

Automated driving using out-of-band on-board sensor data

Vehicles equipped with increasing number of on-board sensors are progressively rolled out and are envisaged to make driving safer and automated. The number of on-board sensors currently on vehicles is at 100 units and is likely to double by 2020 [16]. Locally sensed data, hazard information can be shared with nearby vehicles via a road-side unit to realize smart cruise control systems [17]. Accurate, detailed maps, downloaded from road-side unit can complement sensor data to realize fully autonomous driving [18]. V2I communications can enable automated driving provided vehicles can exchange data with a nearby infrastructure over high speed links with data rates in excess of several Gbps.

ITS depend on the Dedicated Short-Range Communication (DSRC) standards traditionally for vehicular communications. DSRC standard such as IEEE 802.11p has a maximum of 75 MHz reserved in the 5.9 GHz for ITS use. Despite its PHY layer being robust to Doppler spread and low-latency, it suffers from high collision probabilities under medium to high loads due to its contention based random access. Moreover, the realistic maximum data rate does not exceed 6 Mbps [19]. This makes it unsuitable for reliable communication. The 3GPP's Long Term Evolution-Advanced (LTE-A) that specifies channel bandwidth up to 100 MHz has been suggested for use in vehicular communications [20]. But, the maximum data rate it supports is limited to 100 Mbps and end-to-end latencies exceed 100 ms [17]. Therefore, both DSRC and LTE-A fall short of meeting the communication constraints posed by the emerging ITS applications.

Recently commercialized millimeter Wave (mmWave) systems show promise in ensuring Gigabit-per-second throughput and latencies smaller than 10 ms [21]. This can largely be attributed to the availability of contiguous GHz-wide spectrum in the mmWave regime. But, an order of magnitude increase in carrier frequency and very high symbol rate more than 1 GS/s makes mmWave systems, more so in the vehicular context, particularly prone to poor propagation characteristics, hardware impairments, and Doppler induced channel's frequency selectivity.

In mmWave systems, to combat the high path loss, both the base station (BS) and mobile

stations (MSs) will employ highly directional beams realized using the large antenna arrays to provide sufficient received signal power. Due its high sensitivity to shadowing by obstacles, it is considered to be suitable for mostly short range (a few hundred meters) and point-to-point LOS communication [22]. The usage of highly directional antennas helps in achieving high-quality links since thermal noise dominates interference in mmWave links [23]. However, given the mobility of the vehicles and mobility-induced occlusions, frequent repointing of the beams is required and will cause misalignment of the beams resulting in loss of communication. Moreover, the beam training procedure is time consuming and represents a significant time overhead limiting the useful data rate. In that regard, recent work suggests combining out-of-band information from on-board automotive sensors, communication signals at sub-6 GHz and GPS signals for fast, accurate mmWave V2I beam alignment [24] in high mobility scenarios.

1.4 Thesis Contributions

Our link layer design approach for SDRs advances the state of the art and contributes to the research community in the following ways:

- **Standards compliant link layer:** We implement both the PHY-layer and MAC-layer protocols based on the IEEE 802.11b specifications [25], faithfully modeling the DATA and ACK packet structure. Further, the user can select either the classical DATA-ACK or RTS-CTS-DATA-ACK exchanges towards enabling the NAV virtual carrier sensing mechanism. The implemented MAC-layer also has advanced features such as modifying the back off window behavior and changing the channel sensing methods. This is the first time the 802.11 compliant link layer has been developed entirely in MATLAB with performance results reported. Our work provides a testbed to experiment with new MAC protocols starting from the fundamental IEEE 802.11 compliant standard.

- **State-action based design:** We model our system using a finite state machine (FSM) that transitions only on the clock cycles derived from the USRP clock, allowing for slot-time synchronized operations. In this manner, we eliminate the need for external clocks that would be necessary in a hardware-based design, or interrupts that may be preferable using a real-time operating system.

- **Design methodology using a common operating environment:** We use the Ettus Research Universal Software Radio Peripheral (USRP) hardware, a radio front end commonly used in

wireless research. As the basis for our software design, we use MATLAB R2015b and the Communications System Toolbox Support Package for USRP-based radio [26]. We use the MATLAB tools such as MATLAB Coder and the MEX interface to provide for acceleration and timing consistency in the execution of system blocks.

- **Full parameter flexibility:** Using a software-only approach and parameterizing the most important variables allows the user to reconfigure the system as needed to adapt to changes in its environment.

- **Publicly available:** Our software along with a GUI is released to the public for research purposes under the GNU Public License (GPL), and is available for download directly from GitHub [27] and MATLAB Central [28]. The modularity of our code makes it relatively easy to manage and will enable extensibility by the community.

Our work on mmWave and THz-assisted data mule paradigm has the following contributions:

- **Dynamic THz/mmWave Spectrum Switching:** We design a new mode selection protocol that allows the SDN controller to decide when one of these (mmWave or THz) physical layers should be preferentially chosen for a given SD-BS to vehicle link, and develop handoff techniques between these two access technologies.

- **Capacity modeling:** We analytically derive effective data upload rates as a function of channel characteristics of mmWave and THz channels, SD-BS locations, and vehicular paths and obtain bounds on how much data can be delivered between two end points within a permissible time threshold.

- **Vehicle scheduling:** We propose an optimal admission policy at the SDN controller for scheduling multiple vehicles for accessing a given SD-BS, to account for practical road congestion scenarios, considering the heterogeneity of the mmWave and THz links. Since the search for the optimal scheduling is a NP-hard problem, we design a computational-feasible greedy scheduling algorithm, exhibiting a polynomial-time complexity.

- **Simulation and case study:** We show the performance evaluation of our approach through simulations, as well as provide an example of a vehicle-assisted data backhauling considering the road topology of Boston city.

Our work on multi-user mmWave vehicle-to-infrastructure network has the following contributions:

- **Directional multi-user MAC protocol** Since only a finite number of beams and a finite time-frequency resource are available at the BS, not all vehicles requesting access can be concurrently served. We design a MAC protocol that helps the BS service the vehicles in a manner that resource efficient and fair. This is the first multi-user solution considering a fully hybrid beamforming scheme where the BS can receive concurrently in multiple beam directions.

- **Novel resource allocation scheme:** We present a scheme where the BS allocates the time-frequency resource to every MSs based on solving a rectangular bin packing problem. Each time-frequency resource block associated with every vehicle is represented by a smaller rectangle, whose edges are determined by the vehicle's data needs and the duration it will continue to be in LOS with the BS from the time it sent its request. The objective involves a classic combinatorial optimization problem required to minimize the unused time-frequency resource at the BS.

1.5 Novelty of the Contributions

- First work to take a software-only approach and to report 802.11 MAC layer results from over-the-air experimentation on USRP radios. The code base was developed entirely in MATLAB. There are no other MATLAB-based design/code available.

- First work to present an analytic description of the capacity resulting from preferentially switching between mmWave and THz bands. We designed a novel MAC protocol for vehicles equipped with transceivers capable of dynamic spectrum switching to achieve massive data transfers. Actual road maps and data center locations within Boston were used for performance evaluation.

- First work to design the radio frame structure using coherence bandwidth and coherence time specific to mmWave V2I communication. A time-frequency resource allocation scheme specific to mmWave vehicular channel was developed for the first time in multi-vehicle scenario.

1.6 Outline of the Dissertation

The thesis is organized as follows.

Chapter. 2 describes the design and operation of 802.11 functionally compliant link layer.

Chapter. 3 proposes and analyzes a novel approach of combining vehicular networks and software-defined network controlled switching between mmWave and THz access technologies.

Chapter. 4, presents a novel resource allocation scheme for realizing multi-user communication in mmWave V2I network, followed by the concluding remarks on the impact of this thesis towards future link layer design.

Chapter 2

Systems Implementation of 802.11 WiFi Networks

Software defined radios (SDRs) allow fine-grained control of their operation by executing the processing steps in user-accessible program code [29]. This technology forms the building block for applications needing high levels of reconfigurability, such as access points that support multiple wireless standards, or for systems like cognitive radios that incorporate situational intelligence to evolve with the radio frequency (RF) environment [30]. For example, in SDRs, the network designer can tune basic elements, such as modulation, spectrum spreading, scrambling, and encoding through software functions, instead of relying on static hardware, thereby allowing unprecedented access to all aspects of the radio operation.

This chapter details our approach to realize a SDR platform using commonly available tools. We believe that true and repeatable systems-level research is only possible when a commonly used processing environment is used in conjunction with affordable SDR hardware. This motivates our choices for basing our work on MATLAB software and Ettus USRP® N210 hardware [31]. Our approach introduces a novel methodology for an implementation starting at the USRP hardware driver (UHD) and building progressively up the protocol stack. To facilitate quick deployment, it includes an initialization script for the setting and tuning of the reconfigurable parameters at the physical layer based on the specific channel measurements at the chosen experimental site. Importantly, it complies with the processing definitions in the IEEE 802.11b specification, though hardware limitations increase the time to completion of the entire transmission/reception cycle compared to an off-the-shelf hardware-only Network Interface Card.

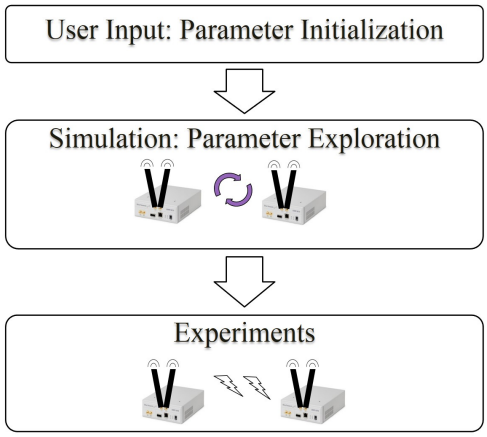


Figure 2.1: System Architecture

The rest of this chapter is organized as follows. In Sec. 2.1, we present the system architecture. We discuss related work on SDR using heterogeneous systems and software platforms in Sec. 2.2. In Sec. 2.3, we describe the slot-time synchronized operations around which the state machines for the designated transmitter and receiver are modeled, and we identify the common system blocks. We describe the algorithms implemented for RFFE and preamble detection in the PHY Layer, followed by a discussion on parameter selection and same-frequency channel operation in Sec. 2.4. The MAC layer design and key algorithms required to implement the CSMA/CA protocol, such as energy detection and random backoff, are described in Sec. 2.5. The experimental setup involving the USRP N210 platform and MathWorks products is given in Section 2.6. In Sec. 2.7, we undertake a comprehensive performance evaluation of the two node and three node system and establish through the experimental results that the system exhibits fairness.

2.1 System Architecture Overview

The operational steps that architect our system are shown in Fig. 2.1. In a given SDR pair, we identify clearly the transmitting and receiving node by using the terms designated transmitter (DTx) and designated receiver (DRx). This terminology helps avoid ambiguity in describing a bi-directional transceiver link, where the transmitter must send out its DATA packet and then switch to a receiver role to get the acknowledgement (ACK). Thus, in the discussion ahead, the DTx alternates between its transmit and receive functions, and the DRx alternates between receive and transmit functions.

In the initialization step, the system is preset with recommended parameters and lets the

user modify a number of parameters for the entire transceiver chain. The user then, in a simulation-only environment, initiates a parameter exploration stage, where all the *nodes* are virtual and are contained within the same computer. The DTx and DRx codes are executed with the user-supplied parameters as constants, and the code cycles through possible variations in the settings of processing blocks as well as entire algorithms, each time identifying the performance that results from these settings.

From this data set, the user is presented with a feasible set of parameter settings. These parameter settings result in less than 5% packet loss at the receiver. This represents the *best case* scenario, for it should be noted that further channel outages will be introduced by the actual wireless channel. Once the user selects one of the possible feasible configurations returned by the search, the code is ready for driving the USRPs for over-the-air experiments.

We adopt the IEEE 802.11b PHY and MAC layer packet structure specifications in our implementation [25] [1]. Our approach collects all the bits in the packet in multiples of 8 octets, which forms one USRP frame. This makes it easy for us to work with the MATLAB system objects (specialized objects required for streaming, henceforth referred to as objects) and with PHY and MAC header fields in the DATA/ACK packet that happen to have sizes that are multiples of 8 octets. Multiple USRP frames will compose the standard-compliant 802.11b packet.

We use differential binary phase shift keying (DBPSK), as the differential component enables us to recover a binary sequence from the phase angles of the received signal at any phase offset, without compensating for phase. In addition, DBPSK requires only coarse frequency offset compensation, without any closed-loop techniques. If residual frequency offset is much less than DBPSK symbol rate, then the bit error rate (BER) approaches theoretical values [32].

2.2 Related Work

2.2.1 SDR Software Platforms

Specialized software is needed to effectively work with the SDR systems and perform the signal processing tasks needed to instantiate wireless communications, such as modulation, preamble detection, encoding, and filtering. GNU Radio is one of the most widely used SDR programs, owing to the fact that it's open source, hardware-independent, and modifiable [33]. Its GUI, GNU Radio Companion, allows the user to build block diagrams to represent complex encoding and decoding schemes. Modules are built in C++, ordering of components performed in Python, and connections

are made using SWIG. Built-in modules allow the user to perform various types of modulation (e.g. GMSK, PSK, QAM, OFDM) and error-correcting codes (e.g. Reed Solomon, Viterbi, turbo). The Software Communications Architecture (SCA) is another open-source, HW-independent framework that models SDR components using data flow diagrams. It is also written using C++ and Python, but intra-block message-passing is accomplished using Common Object Request Broker Architecture (CORBA) middleware. Different software blocks are graphically represented using Unified Modeling Language (UML). The OSSIE software effects an SDR using the SCA framework for interaction with the USRP board [34]. OSSIE provides a GUI to enable the designer to create new waveforms, add new signal processing and modulation routines, and generate the C++/Python code for SCA-CORBA interactions.

2.2.2 SDR on Heterogeneous Systems

There are existing SDR projects implemented on heterogeneous systems that make use of a combination of hardware components to handle computing tasks, including digital signal processors (DSPs), application-specific integrated circuits (ASICs), and field-programmable gate arrays (FPGAs). [35] describes an SoC design for placing transceiver components, including RF receivers at 2 GHz and 5 GHz, a voltage controlled oscillator (VCO), and a baseband filter. [36] proposes a hardware architecture for an embedded software modulation/demodulation (modem) platform, implementing IEEE 802.11a PHY using the Altera Stratix II FPGA and S3C2410 ARM processor. [37] realizes BX501 components on an ASIC and hardware modules for MAC-layer control on FPGA in Verilog.

In addition, there are SDR projects that are implemented in both hardware and software on a platform that comprises both processor and FPGA, and this often includes many custom-made components. WARP is scalable, extensible programmable wireless platform produced by Rice University to prototype advanced wireless networks [38]. It combines a MAX2829 RF transceiver, high-performance programmable hardware Xilinx Virtex-4 FPGA board, and an open-source repository of reference designs and support materials. This platform has been used to build, among many other things, a full duplex IEEE 802.11 network with OFDM and a MAC protocol [39], and a distributed energy-conserving cooperation MAC protocol for MIMO performance improvements [40]. USC SDR presents a wireless platform to remove bottlenecks from current SDR architectures [41]. It combines Xilinx VC707 PCI FPGA development boards with self-sufficient radio front-end daughterboards to make a MIMO testbed, using the FPGA Mezzanine Card (FMC) connection. Real-time

SW architecture allows user programs to perform signal processing tasks, PHY- and MAC-layer algorithms. The Sora soft-radio stack combines a Radio Control Board (RCB) with a multi-core CPU. The RCB that consists of a Virtex-5 FPGA, PCIe-x8 interface, and 256 MB of DDR2 SDRAM [42]. Microsoft Research built the SoftWiFi Demo radio system to interoperate with 802.11a/b/g NICs, and it uses a company-proprietary language for SDR description.

There are other SDR projects that are implemented using Xilinx Zynq SoC, utilizing both the PS/ARM processor and PL/FPGA fabric. Iris uses XML description to link together components to form a full radio system [43]. Components are run within an engine, which could be either a PS processor core or PL logic fabric. It's tested using OFDM for video transmission. GReasy presents a GNU radio version for Xilinx Zynq, using Tflow to instantly program FPGA fabric [44]. [45] uses Zynq SoC to implement digital pre-distortion algorithm (DPD), which mitigates the effects of power amplifier (PA) nonlinearity in wireless transmitters, something required for 3G/4G base stations. This uses Vivado HLS to design the PL component and receives up to 7X speedup from HW acceleration. [46] proposes a scalable cluster of Zynq ZC702 boards, controlled by a Zedboard that acts as a task mapper to partition data flows across the Zynq FPGAs and ARM cores. tFlow rapid reconfiguration software was used to build FPGA images from a library of pre-built modules.

[47] describes an SDR-based testbed that implements a full-duplex OFDM physical layer and a CSMA link layer using MATLAB R2013a, MATLAB Coder on USRP-N210 and USRP2 hardware. The IEEE 802.11a based PHY layer, incorporates timing recovery, frequency recovery, frequency equalization, and error checking. The CSMA link layer involves energy detection based carrier sensing and stop-and-wait ARQ. It outlines some strategies in establishing bidirectional communications. However, this approach involves additional development efforts to improve speed and enable full-duplex operation.

The above platforms make for capable choices in terms of performance. However, our choice of the operating environment was motivated by the price point, which is why we chose to use the combination of USRP N210 hardware and MATLAB software towards link layer implementation. So far there has been little support for MATLAB in the existing SDRs and, in this regard, our framework allows for quick development of new higher layer protocol design. In addition, our software-only infrastructure allows for full flexibility of parameter choices, an option not available to many other SDR platforms.

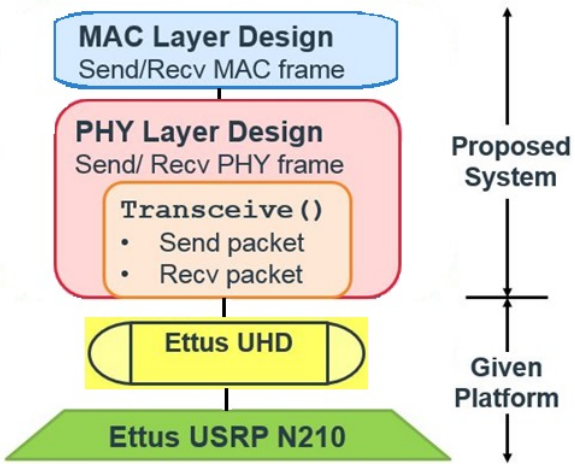


Figure 2.2: System Methodology

2.3 State-action based System Design

Our approach involves first designing a number of (i) state diagrams to reflect the logical and time-dependent operational steps of our system, and (ii) block diagrams to reflect the sequential order of operations. Furthermore, we structure the MATLAB code in a way that enables slot-time synchronized operations. For the implementation, we use MATLAB Coder to generate the MEX functions for the USRP objects on an Ubuntu 64-bit platform that serves as the host computer for the USRPs.

Since the underlying code in a MEX function is written in C, it is generally faster than the interpreted MATLAB. The speed-up in performance can vary depending on the application. In our case, we preferred the MEX interface because it can enforce a consistent processing time per frame. The interpreted MATLAB, unlike the MEX, lacks this ability because it exhibits significant deviation from the desired timing. In addition, time-sensitive operations such as frequency offset compensation, show speed improvement using MEX.

Our system design builds upon an already-defined platform, the USRP, produced by a well-known platform supplier, Ettus Research [31]. The communication between the USRP and host computer is established in MATLAB using the Communications System Toolbox (CST) USRP Radio support package, which acts as a wrapper for the Ettus USRP Hardware Driver (UHD) drivers. Identifying the manner in which the RF samples are transported between the USRP and a calling function defines the manner in which we must build the physical (PHY) layer, as illustrated in Fig. 2.2.

The UHD transfer of a frame of samples to a transmit buffer is performed as soon as it is requested while the UHD retrieval of a frame from a receive buffer has to wait until the next rising edge of a clock cycle before trying to retrieve again. The most common undesirable behaviors that can occur are *underflow* and *overflow*. *Underflow* occurs when the radio requests for a frame of data from the transmit buffer, but the host is not yet ready to provide it. *Overflow* occurs when the receive buffer becomes full and buffered data must be overwritten.

In this regard, we define real-time operation over the course of an entire DATA-ACK packet exchange using equation (2.1) below:

$$t_{receive} \leq t_{radio} \tag{2.1}$$

where t_{radio} is the frame time stipulated by the USRP radio's analog-to-digital converter (ADC) and $t_{receive}$ is the average time to recover any given frame, which includes the time to retrieve a frame from the receive buffer, process the retrieved frame to decode it into the corresponding bits, and other memory and conditional operations.

Essentially, we operate in real-time if we meet the timing deadline set forth by equation (2.1). Such an operation will guarantee a stable, basic bi-directional link that shows no sign of any undesirable system behavior, such as buffer underflow or buffer overflow. A MAC protocol that effectively schedules packet transmissions reduces the potential for packet collisions and buffer overflow, thereby decreasing packet errors.

2.3.1 Slot-time synchronized operations

Any IEEE 802.11-based wireless transceiver implementation must have the ability to perform operations based on some slot-based timing. Performing such slot-time synchronized operations will let us realize time-sensitive functions, for example, make a node wait for a backoff (BO) duration before sending a DATA packet.

Interpreted MATLAB or any other software that runs on the host computer may have trouble performing such operations in this manner, even by actively waiting. For this reason, we rely on the USRP for our timing. Using the value for USRP interpolation/decimation defined in Section 2.4.3.1, we can calculate the slot time. Then, we write our while loop in the main program so that it calls the transceive function once per loop, running helper functions to prepare data to transmit or process received data based on the active state, as shown in the program code in Listing 2.1.

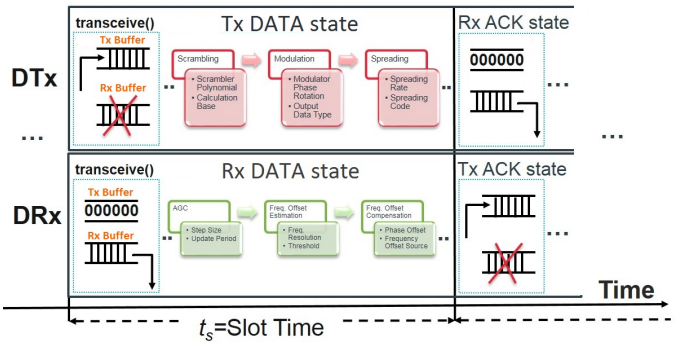


Figure 2.3: Transceive Function Behavior as Defined by Operational State

```
while ~endOfTransmission
  if (state==Tx)
    data2Tx = processData2Tx();
  end
  dataRxd = transceive(data2Tx);
  if (state==Rx)
    processRxdData(dataRxd);
  end
end
```

Listing 2.1: Main program calls transceive function

At the heart of the transceiver model is the *transceive* function, as shown in Listing 2.2. By design, *transceive* is called at a constant time interval that we define as a slot time. At each slot time, *transceive* sends and receives a fixed number of samples, which we refer to as a *USRP frame*.

We define a slot time as the smallest unit of time in which our SDR can make a decision. In our design, the frame time is the minimum time our system takes to make a decision and hence, we equate it to the slot time. In this regard, our transceive function performs two actions: it gets a frame from, and puts a frame into the USRP buffers at fixed time intervals [32]. A data frame is sent or received every slot time and further, the functions we define for processing the received data frame or preparing a new data frame to transmit are intended to complete in less than a slot time to ensure timing accuracy. In practice, we recognize that the processing time for certain frames may exceed the radio time, t_{radio}, but the recovery time, $t_{receive}$, converges to the radio time.

When a node (either DTx or DRx) enters a transmit state (refer to Fig. 2.3), it transmits the samples in the transmit buffer and ignores all samples in the receive buffer. On the other hand, when a node enters a receive state, it retrieves samples from the receive buffer for processing and puts

zeroes in the transmit buffer. This way, we make sure that the samples in the transmit and receive buffer are current and relevant.

```
function dr = transceive(ft, d2s)
persistent hrx htx;
% Initialize received data variables
dr = complex(zeros(nspf,1));
ns = 0;
% Initialize system objects once
if isempty(hrx)
  hrx = ...; htx = ...;
end
% Flag to release system objects
if ft
  release(hrx); release(htx);
else
  step(htx,d2s);
  while (ns == 0)
    [dr,ns] = step(hrx);
  end
end
```

Listing 2.2: Transceive function code

The step method of the transmitter object operates in a blocking way as it returns only after the radio accepts the frame to be transmitted. On the other hand, the step method of the receiver object returns right away, hence it is non-blocking.

The step call of receiver object will return 0 as length of the received frame if there is not enough data in the radio. Once the radio collects enough data, the next step call returns a non-zero length value and the valid data. Since we know the sample rate of the data and the number of samples in a frame, we can calculate how long it takes to get one frame of data from the radio. The while loop blocks the transceive function until a frame of data is received. Therefore, we can use the call duration of this function as our clock source.

2.3.2 Designated Transmitter State Machine

In implementing the carrier sense multiple access with collision avoidance (CSMA/CA)-based protocol in the link layer, we identify 4 main states for the DTx, as shown in Fig. 2.4. Table 2.1 identifies the blocks in each substate and is described in detail in Section 2.3.4.

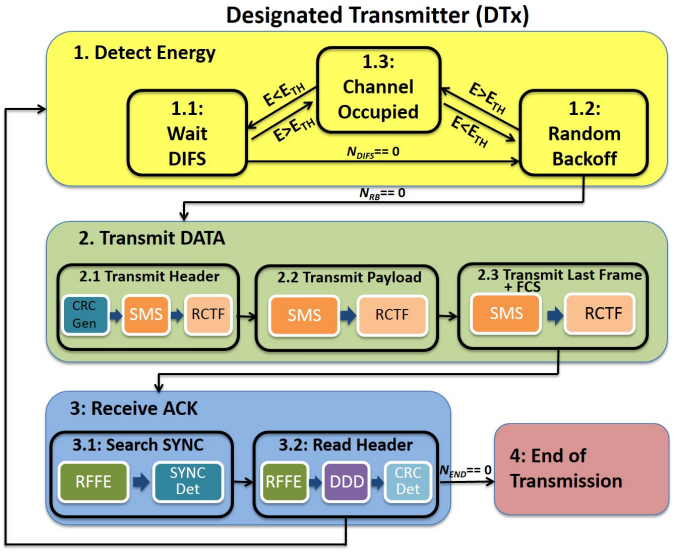


Figure 2.4: States for the Designated Transmitter (DTx)

Table 2.1: Substate Operation Combinations

Block	Block Components
SMSRC	Scrambling, Modulation, Spreading, and Raised Cosine Transmit Filter (RCTF)
RFFE	Radio Frequency Front End: includes Automatic Gain Control (AGC), Coarse Frequency Offset Estimation (CFOE), Frequency Offset Compensation (FOC), and Raised Cosine Receive Filter (RCRF)
PD	Preamble/SYNC Detection: Find SYNC in Rx'd USRP frames
DDD	Despreading, Demodulation, and Descrambling

2.3.2.1 Detect Energy

At the start, a new USRP frame arrives, and gets stored in a receive buffer. The DTx begins to continually sense energy in the channel and decides to transition either into a backoff state or to a transmit state depending on whether or not the channel is busy. It first waits for a DCF interframe spacing (DIFS) duration and then waits for a random amount of time that is chosen uniformly from a progressively increasing time interval. Only when the channel is free does the DTx decrement the chosen random backoff time; otherwise, it stalls. Only when the backoff time counts down to zero

does the DTx attempt to transmit.

2.3.2.2 Transmit DATA

Upon entering this state, the DTx prepares the DATA packet and then, by calling the transceive function continually, places it in the transmit buffer of the USRP which then gets transmitted over the air. After transmitting the DATA packet, two possibilities exist. The transmission is successful with the reception of an ACK, or the transmission is not successful due to packet collision with another DTx.

2.3.2.3 Receive ACK

As soon as the DATA packet is transmitted, the DTx moves into the Receive ACK state, searching and decoding the Physical Layer Convergence Procedure (PLCP) header in the received ACK. If that is successful, the frame control and the address fields are read-out from the subsequent MAC header and checked for accuracy. The DTx then progresses to transmit a new frame and repeats the above mentioned sequence of steps until the last frame is successfully transmitted. On the other hand, if no ACK is received, the packet is considered lost and the DTx backs-off for an increased random backoff time and re-attempts transmission.

2.3.2.4 End Of Transmission

When there are no more DATA packets left to be transmitted, the DTx reaches the end of transmission (EOT) state.

2.3.3 Designated Receiver State Machine

Similarly, we identify 3 main states for the DRx as shown in Fig. 2.5. Unlike the DTx, the DRx does not perform energy detection.

2.3.3.1 Receive DATA

When the DRx successfully detects the Preamble and the Start Frame Delimiter (SFD), it decodes the PHY and MAC header and then progresses to extract the payload. When extracting the last set of payload bits, Frame Check Sequence (FCS) is obtained and checked.

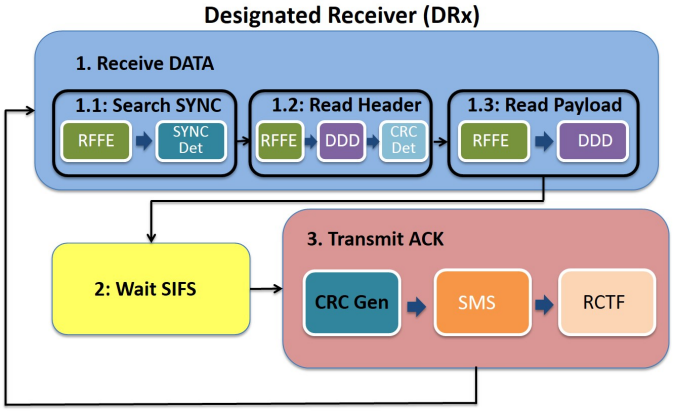


Figure 2.5: States for the Designated Receiver (DRx)

2.3.3.2 Wait SIFS

The DRx waits for a fixed interval of time, referred to as Short Inter-frame Space (SIFS), before sending an ACK packet post reception of the DATA packet.

2.3.3.3 Transmit ACK

The DRx sends out an ACK addressed to the DTx when it successfully retrieves all the payload bits.

2.3.4 System Blocks

Within each of the substates in the FSM diagrams (Figs. 2.4 and 2.5), there are sequential operations that need to be performed. In order to simplify the logic of which operations must be performed in each state, we define a number of *blocks* to comprise the most common operations, as shown in Table 2.1. Identifying the grouping of blocks with the related substates helps better organize and restructure the implemented code.

In each substate of DTx state 2 (Tx) and DRx state 2 (Tx ACK), SMSRC is performed prior to each *transceive* (send and receive operation). In DTx substate 3.1 and DRx substate 1.1, RFFE and PD are performed after each transceive. In DTx substate 3.2 and DRx substates 1.2, RFFE and DDD are performed after each transceive.

2.4 PHY Layer Algorithms

2.4.1 RF Front End Algorithms

The components in the RFFE block recover a signal prior to preamble detection. These include the automatic gain control (AGC), frequency offset estimation and compensation, and raised cosine filtering. The ordering of these components is an important consideration, and through exhaustive simulations, we found the preceding order to be ideal. The AGC algorithm counters attenuation by raising the envelope of the received signal to the desired level. We chose to use the MATLAB `comm.AGC` object [48]. To accurately estimate the frequency offset between the receiver and the transmitter, we chose to use the `comm.PSKCoarseFrequencyEstimator` object, which uses an FFT-based-based method, based on equation (2.2), and finds the frequency that maximizes the FFT of the squared signal:

$$f_{offset} = \arg\max_{f} \mathcal{F}\{x^2\} \tag{2.2}$$

where x is the signal, $\mathcal{F}$ denotes the Fast Fourier Transform (FFT), and f_{offset} is the frequency offset.

2.4.1.1 Speeding up the RFFE block

From our initial experiments, we know that a frequency resolution (on the order of 1-10 Hz) is necessary in order to do preamble detection accurately. Setting such a low frequency resolution takes too long to execute with a sample rate of 200 kHz, or 200,000 samples per sec. For this reason, we decided to decimate the signal by a factor of 22 (the RCRF factor times the spreading rate) before CFOE, which is, in essence, an FFT. After decimation, we experimented with raising the CFOE's frequency resolution by an order of magnitude to 10-100 Hz, and determined that it is accurate up to 100 Hz and meets the timing guidelines set by radio time.

We employ a FIR Decimator step, as shown in Listing 2.3, that enables us achieve an order of magnitude reduction in RFFE block execution time. In essence, we are able to get enough frequency estimation accuracy with reduced sample rate (hence the use of decimation) and 100 Hz frequency resolution, which requires much less processing power than full frame higher resolution estimates.

```
(1) dsp.FIRDecimator('DecimationFactor',22);
(2) comm.PSKCoarseFrequencyEstimator(
        'Algorithm','FFT-based', ...
        'FrequencyResolution',cef,...
        'ModulationOrder',2,...
        'SampleRate',(2e5/22));
```

Listing 2.3: RFFE Decimation Method

2.4.2 Preamble Detection Algorithms

The IEEE 802.11b preamble is a sequence of all one bits that undergoes scrambling. Since the scrambling phase is not known, and the received signal is correlated to the zero phase scrambled sequence, the maximum correlation position may not be the synchronization position. Therefore, the standard provides Start Frame Delimiter (SFD), to fine tune the synchronization time.

Preamble detection (PD) is performed in two stages. In the first stage, we perform a cross-correlation of the received complex data after raised cosine filtering with the expected real preamble to get an estimate of where the preamble starts, giving the so called synchronization delay. Finally, in the second stage, we look for the SFD immediately after the preamble in the descrambled bit stream. If it is not in the expected place, we perform a cross-correlation on a window of descrambled frame samples to the left and right to further fine-tune the synchronization delay.

2.4.2.1 Optimization of Preamble Detection

Detecting the Preamble fast and with high accuracy is critical to the speed at which the nodes can reliably exchange DATA/ACK packets. In one implementation, we exploit the property of the cross-correlation of two real signals in the frequency domain to compute the same (i.e. the point-wise product of the Fourier transform of the two signals), followed by an inverse Fourier transform resulting in the cross-correlation of the two signals. Since one of the signals is the expected preamble, its Fourier transform can be pre-computed and loaded into the workspace during run-time. We experimented with several MathWorks utilities to compute cross-correlation faster
(e.g. `dsp.CrossCorrelation` object, `xcorr` function).
We determined the version of `dsp.Crosscorrelator('method', 'fastest')` compiled using MEX to be the fastest among all the candidate methods for computing cross-correlation with increasing signal lengths, as shown in Fig. 2.6. It is important to note that although we operate with

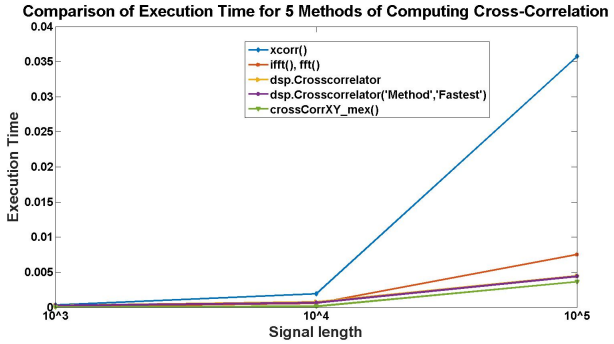


Figure 2.6: Comparison of Execution Time for 5 Methods of Computing Cross-Correlation

Table 2.2: Important Parameters

Param	Block	Description	Range	Tunable
R_i, R_d	USRP	USRP Interpolation Decimation Factor	500	No
L_f	USRP	USRP Frame Length	64 bits	No
L_p	Frame	#Octets per 802.11b Packet Payload	0-2312	Yes
K	RFFE	AGC Max Power Gain	30-60	Yes
N	RFFE	AGC Adaptation Step Size	0.01-0.5	Yes
Δf	RFFE	Frequency Resolution	1-100 Hz	Yes

signal lengths on the order of 10^3, preamble detection is a frequent operation, so savings in time add up quickly.

We declare packet detection only if the second stage finds a perfect match for the SFD. This approach greatly minimizes false packet detections.

2.4.3 Parameter Selection

The initialization step described in Section 2.1 lets us carefully choose a number of design parameters (see table 2.2).

2.4.3.1 Constant Parameters for USRP & IEEE 802.11b Frame

We recognize parameters that cannot change during packet transmission/reception and have to be fixed. The number of octets in the payload per IEEE 802.11b packet should be maximized to

decrease the header overhead. In that case, a large frame size is preferred as it reduces the percentage of overhead processing. On the other hand, the frame size should be minimized to make quick decisions with a small number of samples or bits, unlike a large frame size which increases the frame time, thereby reducing the resolution of time ticks for the system. We chose frame length of 1408 as a well balanced compromise between these two requirements. For this reason, the frame length is left fixed.

The USRP N210 analog-to-digital converter (ADC) operates at a fixed rate of 100 MHz. The USRP interpolation-decimation rates control the rate of transmitting and receiving frames. For example, setting interpolation rate, R_i, and decimation rate, R_d, to 500 ensures that the ADC and DAC convert a sample every 5 μs, as shown in equation (2.3).

$$\begin{aligned} t_{sample} &= R_i/(100Msamples/sec) \\ &= 500/10^8 \\ &= 5 \times 10^{-6} sec/sample \end{aligned} \tag{2.3}$$

Setting frame length, L_f, to 1408 samples means that a frame is retrieved by the transceive function every 7.04 ms, as shown in equation (2.4).

$$\begin{aligned} t_{radio} &= L_f \times (R_i/100Msamples/sec) \\ &= 1408 \times (500/10^8) \\ &= 7.04 \times 10^{-3} sec/frame \end{aligned} \tag{2.4}$$

Even though our system may take more than 7.04 ms to process a frame every once in a while, the buffers in the USRP receiver prevents the system from overrunning (or lose samples) and the system, on average, stays real-time.

2.4.3.2 Tunable Parameters for RFFE Block

Tunable parameters can change during transception. For example, the AGC adaptation step size controls the convergence speed of a received signal's envelope to the desired level. In other words, it governs the speed of convergence. The frequency offset estimation component's frequency resolution setting is an important design consideration as it is inversely proportional to the FFT length. A lower frequency resolution gives more accurate offset estimates, but with increased computational time.

2.4.4 Same-Frequency Channel Operation

In a multi-node setting, it is advantageous to operate the transmit and receive links, at the DTx and DRx, in the same band of frequencies. Thus, we set both DTx and DRx to operate at the same center frequency. Unlike different-frequency channel operation, this eliminates the need for repeated switching of transmit and receive center frequencies when transitioning among the energy detection, transmit, and receive states. In addition, it makes for an easier implementation of medium access and contention resolution.

From our initial experiments, we learned that the receive-only port, *RF2*, of the USRP leaks about 7 dBm into the transmit & receive port, *RF1*. The effect of this leakage causes the DTx to detect the preamble in its own DATA packet while it is waiting for an ACK. We added logic to ensure that the DTx rejects its own DATA packet as soon as it reads the MAC header and does not find the expected ACK frame control sequence.

2.5 MAC Layer Design

We first implement the CSMA/CA protocol that allows the nodes to sense the channel and attempt to transmit packets only when the channel is idle to avoid packet collisions. Then, we modify this base implementation with the standards-specific functions, as described below.

2.5.1 MAC Overview

Our MAC layer employs the Distributed Coordination Function (DCF) strategy incorporating the CSMA/CA mechanism as it is described in the IEEE 802.11 specification [1]. Our implementation incorporates the key features of CSMA/CA, namely, 1) carrier sensing via energy detection, 2) DCF interframe spacing (DIFS) duration, and 3) exponential random backoff. An illustration of the overall steps of the operation is shown in Fig. 2.7 and Fig. 2.8.

2.5.1.1 Energy Detection

Channel occupancy can be identified by detecting RF energy in the channel. Energy in the channel is computed using equation (2.5).

$$Energy = \sum_{n=1}^{n=N} |x(n)|^2 \tag{2.5}$$

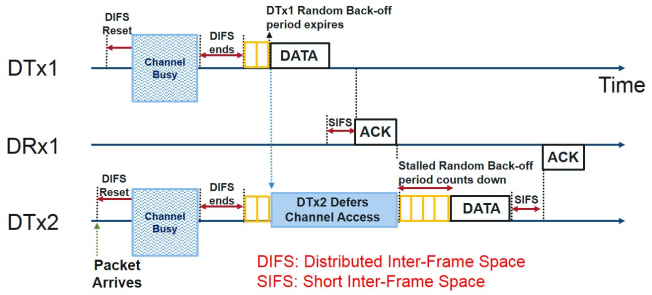


Figure 2.7: CSMA/CA/ACK Timeline Chart - Energy Detection

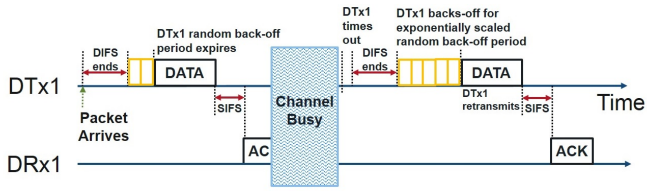


Figure 2.8: CSMA/CA/ACK Timeline Chart - Exponential Random backoff and Retransmission

In our implementation, $x(n)$ represents the samples in the USRP frame retrieved from the receive buffer of the USRP.

2.5.1.2 DIFS Period

The standard specifies that when a packet is prepared by the DTx and ready to be sent to the intended DRx, the DTx must actively listen to the channel for a fixed specified amount of time known as the DIFS period. If during this period, the DTx senses RF signal energy from other transmitting devices (i.e. when the channel is found busy), it defers the transmission and enters a *Channel Occupied* state. In this state, the DTx stays idle as long as the ambient RF energy is above a specified threshold. When the energy drops below the threshold (i.e. the medium is sensed to be free), the DTx resets the DIFS duration and starts counting down again.

2.5.1.3 Binary Exponential Random Backoff

This method of random backoff is used to schedule retransmissions after collisions. Essentially the retransmissions are delayed by an amount of time determined by a minimum contention window, c_{min}, and the number of attempts to retransmit the DATA packet. With this increased number of retransmit attempts, the delay can increase exponentially.

When the DIFS duration runs out, the DTx transitions to the exponential random backoff state wherein it generates a random backoff delay uniformly chosen in the range [0, W-1] where W is

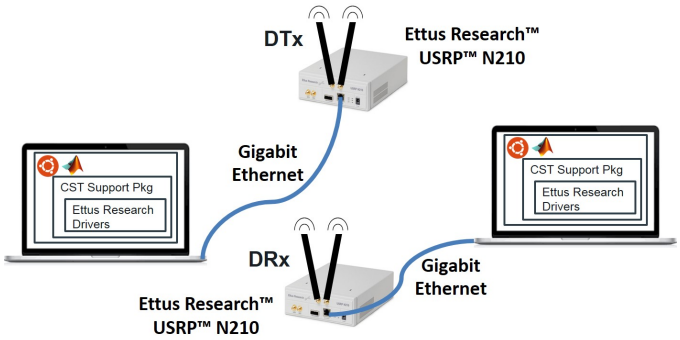


Figure 2.9: Transceiver Hardware Setup

called the contention window (CW).

In correspondence with the IEEE 802.11 standard, time is slotted using a basic time unit which is the time needed to detect the transmission of a packet from any other station. In our implementation, t_{radio} represents the basic time unit for the system, within which we can detect another DTx transmitting.

As an example, after k collisions, a random number of slot-times is chosen at random from $[0, 2^k\text{-}1]$ as described in equation (2.6).

$$Random\ Back\text{-}off\ Delay = randi[0, 2^k\text{-}1] \times t_{radio} \tag{2.6}$$

The MATLAB `randi` function picks an integer uniformly at random from the specified interval. In our implementation, we have the option to truncate the exponentiation with a fixed number of retransmits so as to have a ceiling for the Random backoff Delay.

2.6 Experimental Setup

We use the USRP N210 platform [31], as it allows us to define the parameters listed in Section 2.4.3.1, connect to a PC host using a gigabit Ethernet cable, and to program it using MATLAB [26]. We use the Ubuntu OS, with send and receive buffer sizes for queues set to ensure that there is enough kernel memory set aside for the network Rx/Tx buffers. We also set the maximum real-time priority for the `usrp` group to give high thread scheduling priority. This change is made by adding a line to the file `\etc\security\limits.conf` that sets the `rtprio` property for the `@usrp` group to 50. The overall setup is shown in Fig. 2.9.

2.6.1 Communications System Toolbox USRP Support Package

We use the Communications System Toolbox objects for our design [49]. We used the `comm.AGC` object and the PSK coarse frequency offset estimator that allows us to work with FFT-based options. These objects facilitate easy generation of C code using MATLAB Coder. Here, the `comm.SDRuTransmitter` object puts a frame on the USRP transmit buffer, and `comm.SDRuReceiver` gets a frame from the USRP receive buffer. However, this approach has some disadvantages, such as a requirement for fixed frame length and single-threaded `step` methods.

2.6.2 MATLAB Coder

A number of steps must be taken to make the MATLAB code ready for C code generation using MATLAB Coder. All variables that do not change over the course of the program execution are given a static size and type (including real or complex). All objects are declared as persistent variables as they cannot be passed into MEX functions. The first call to each function tests whether the persistent variable is empty, and initializes each object if true. The transceive and RFFE function code are designed in this manner.

2.7 Experiments and Results

We choose to evaluate our system using a number of experiments. First, we time the reception of DATA packets at the DRx. Next, we time the RFFE block using both interpreted MATLAB and MEX. We then perform a two node experiment, measuring bi-directional link latency and packet error rate. We then profile execution time in the transmitting states. Finally, we perform a three node experiment, measuring previous metrics and goodput.

In the three node experiment, we address the fairness in our system. Considering two bi-directional links emerging from two DTxs but incident on a DRx helped us to design (within hardware constraints) and demonstrate a stable bi-directional link and allowed us to test the fairness enabled by the MAC protocol in the most simplified way, thereby eliminating the need for further multi-node scenarios. Performing more scenarios would require setting up and performing experiments involving multiple nodes and host machines, and would take a large amount of effort. Such an effort would not have helped us in attaining our goal of fairness assessment. In addition, we can presume that an increase in the number of DTx nodes would exhibit less fairness because it increases the likelihood of collisions. In this situation, nodes that would collide would also choose to wait for increased backoff

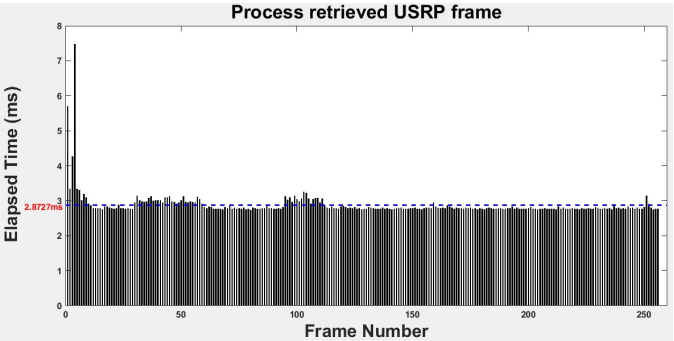


Figure 2.10: Process Time per USRP frame at DRx

periods, which would give other nodes an increased opportunity for transmissions. Additional tests would not be necessary to confirm this hypothesis.

2.7.1 Timing DATA Packet Reception at DRx

At the DRx, after preamble detection, the elapsed time to process each retrieved USRP frame corresponding to an entire DATA packet is shown in Fig. 2.10. The dotted line represents the average of all the frame processing times towards a DATA packet reception. The DTx sends out a DATA packet that is made up of 258 USRP frames. After recovering the header bits, the DRx retrieves the payload, which is 250.5 USRP frames (2004 octets). Since the Preamble is 128 bits long, it corresponds to 2 USRP frames. Hence, we account for the reception of (258 - 2) = 256 USRP frames in the DATA packet.

The time to process any given frame usually falls below the desired frame time, t_{radio}, and is fairly constant at 2.87 ms. The first set of frames have a higher processing time because they consist of the MAC header information that must be resolved (e.g. frame control, MAC address).

2.7.2 RFFE Block Timing

The timing of the RFFE block for various values of the frequency resolution parameter in interpreted MATLAB and C code compiled into MEX is shown in Fig. 2.11. The addition of a FIR decimation step in the RFFE block reduces the sampling rate of the input for the subsequent coarse frequency offset estimation (CFOE). This reduction helps in increasing the frequency resolution, currently set at 100 Hz, which is the key parameter in controlling the execution time of CFOE. Further, we benefit from the improved accuracy of CFOE in that it corrects the signal so well that the later preamble detection block produces the correct synchronization delay to detect the start of

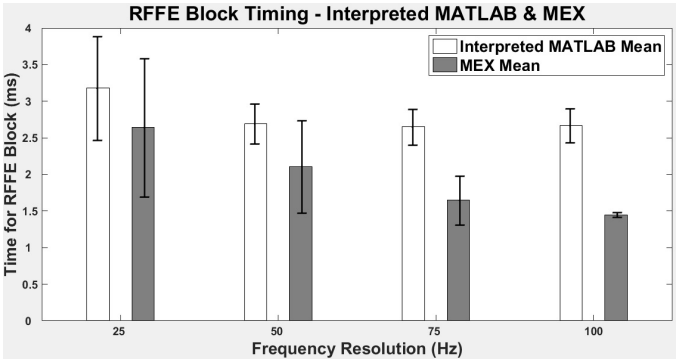


Figure 2.11: RFFE block timing using interpreted MATLAB and MEX

DATA/ACK packet. The results clearly establish that average execution time for the RFFE block decreases with increase in frequency resolution. The reason for this is that CFOE uses progressively smaller FFT lengths. As before, the average execution time using MEX is generally smaller than using interpreted MATLAB. Also, the standard deviation for MEX results is always significantly less. Hence, MEX is a better option for the purpose of enforcing consistent RFFE execution times, which is required for slot-time synchronized operations.

2.7.3 Two Node Performance (1 DTx and 1 DRx)

Link layer contention resolution and other MAC layer functions depends on the ability to reliably generate alternating DATA-ACK packets between the sender and receiver. In this regard, determining the performance of this basic link is important.

Packet error rate (PER) and bi-directional link latency are key performance indicators of the two node system. Of particular interest is the performance of the system when the transmit power level of the DTx is decreased below standard levels. The DTx was set up to send IEEE 802.11b compliant packets each with a large payload of random binary bits (2012 octets). The DRx receives the packet, checks for the correctness of the header information and acknowledges the receipt of the DATA packet by transmitting an ACK. The experiment was designed to be statistically significant, and hence, 100 packets were transmitted for each of the 5 different transmit gain settings. The results were averaged over 5 runs.

The experimental setup involved two host computers, both running MATLAB R2015b on a Ubuntu OS environment, each interfaced via the Ethernet cable to a USRP N210. The devices are configured to be DTx and DRx respectively and are kept about a meter apart.

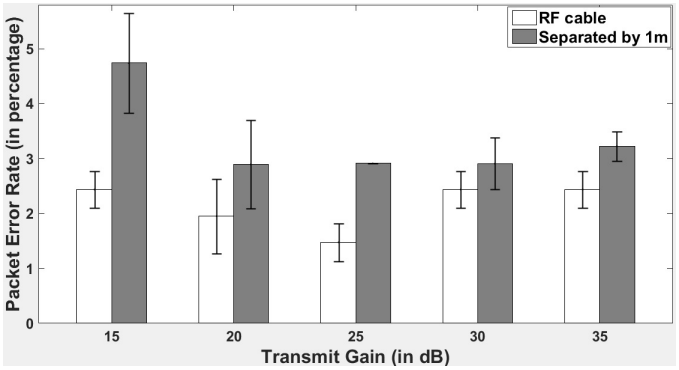


Figure 2.12: Two Node Performance: Packet Error Rate

2.7.3.1 Packet Error Rate

A packet is in error if the ACK for the same is not received in time by the DTx. This could mean that either the packet could not be decoded properly by the DRx or that the ACK was corrupted or lost while in transit to the DTx. An ideal system must recover quickly from such errors and, best trade-off PER and bi-directional link latency. PER is measured on average in percentage reflecting how many packets might be received in error for every 100 packets sent.

2.7.3.2 Bi-directional Link Latency

Bi-directional link latency is the average time taken by the DTx between sending a DATA packet and receiving the corresponding ACK packet. The bi-directional link latency includes any delay resulting from retransmissions accounting either for loss of DATA packet or ACK packet. Note that since the MAC layer code runs during the course of the experiment, the bi-directional link latency includes the DIFS duration and the random backoff period both set at 20 ms. The MAC layer functionality however is largely dormant in the 2 node case due to the lack of contention. Bi-directional link latency is averaged for a packet in seconds.

In the two node system, increasing DIFS and backoff time practically has no effect on the packet error rate due to lack of contention. However, increasing DIFS and backoff time also increases link latency by the same amounts. It should be noted that in the specifications, DIFS and contention window slot time are both fixed constants.

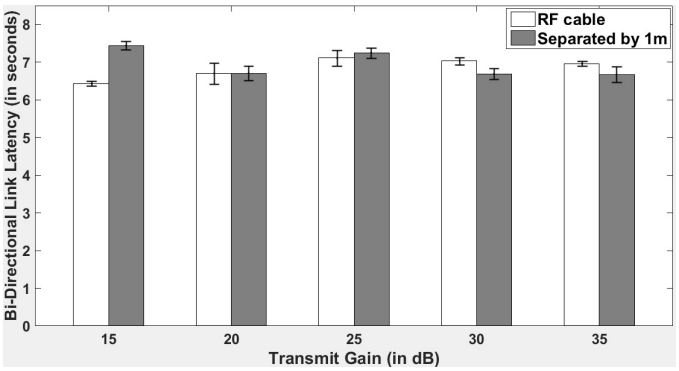


Figure 2.13: Two Node Performance: Bi-directional Link Latency

2.7.4 Profile of Time Elapsed in DTx States

At the DTx, we measured the time elapsed in each state for a DATA-ACK packet exchange. The stacked plots shown in Fig. 2.14 and Fig. 2.15 show the breakdown of the time spent in each substate. The plot at the top shows the small contributors to the overall processing time, and the one at the bottom shows the large contributors. Both the plots are part of the same DATA-ACK packet exchange and are separated for clarity. Note that (1) the time spent in the MAC portion of the code includes the time elapsed to detect energy in the channel continually together with the DIFS and random backoff duration, and (2) the time taken to send the IEEE 802.11b DATA packet includes the time to prepare the packet.

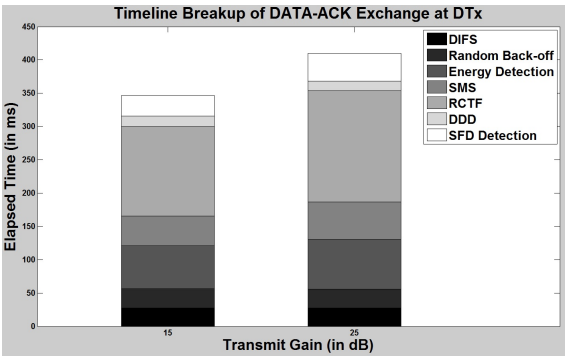


Figure 2.14: Timeline Breakup of DATA-ACK Packet Exchange at DTx

From Fig. 2.12 and Fig. 2.13, we can infer that the 2 node experiments show that the system guarantees a consistent $\leq 5\%$ packet error rate and approximately 7 seconds of bi-directional link latency (DATA-ACK packet exchange inclusive of the MAC functions) over a wide range of transmit gains (15-30 dB). Importantly, varying the distance between the 2 nodes does not significantly affect performance. Even moving the 2 nodes farther apart while still in line-of-sight (e.g. by 15 meters),

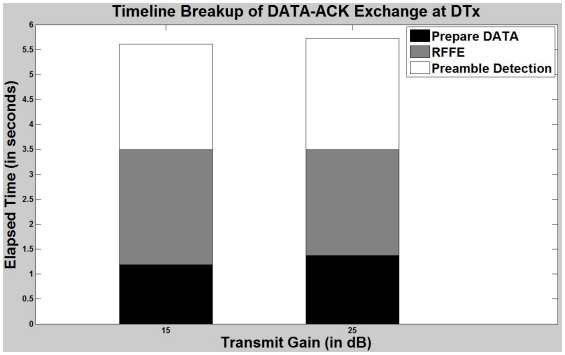


Figure 2.15: Timeline Breakup of DATA-ACK Packet Exchange at DTx

the PER and bi-directional link latency stayed consistent. However, the presence of many metallic surfaces, such as in our lab setting, give rise to multi-path reflections that can be strong and result in packet errors. The fact that the performance was significantly better when the nodes were connected by RF cables confirms the case.

Keeping the packet sizes identical (DATA and ACK are 2072 octets and 40 octets long respectively), the standard off-the-shelf devices, operating at standard specified timings, the link latency $L_{std-link}$ (neglecting media contention, backoff times, and retransmissions) can be computed using Equation 2.7. TxDATA and TxACK represent the elapsed time (in microseconds) to transmit a DATA packet and an ACK packet (at 1Mbps) respectively.

$$\begin{aligned} L_{std-link} &= DIFS + TxDATA + SIFS + TxACK \\ &= 50\mu s + (2072 \times 8)\mu s + 10\mu s + (40 \times 8)\mu s \\ &= 16956\mu s = 16.956ms \end{aligned} \tag{2.7}$$

Comparing this to t_{radio} in equation (2.4), we see that the link latency is in the same order as our slot time. Owing to hardware constraints, packet exchanges in standard devices are in the order of milliseconds while exchanges in this system are in the order of seconds. However, we argue that this is acceptable because our system adds the feature of software definition, which requires additional time for execution.

2.7.5 Three Node Experimental Setup (2 DTxs and 1 DRx)

Given that without the MAC layer, the DATA/ACK packet collisions and the link latencies will be unacceptably high, we performed experiments to assess the MAC performance with a set of 3 USRPs (three nodes: 2 DTxs and 1 DRx). To that end, we implemented MAC functions to

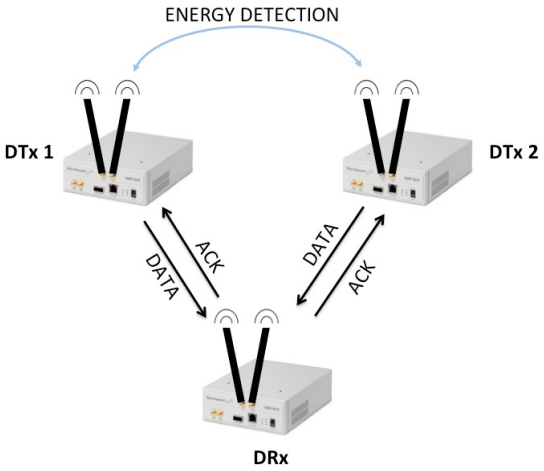


Figure 2.16: Three Node System with 2 DTxs and 1 DRx

distinguish the two links and fine-tuned the MAC/PHY parameters of the system. We expect to see increased bi-directional link latency and PER as the DTxs contend to gain access to the channel leading to packets collisions and subsequent retransmits.

In our 2 node experiments, we confirmed that for a wide range of transmit gains, the performance remains consistent. We now have two independent links incident on one shared DRx, and hence, we do not expect to see much difference in the performance of the two links when varying the transmit gains here in the 3 node case. Instead, we measured bi-directional Link Latency and Packet Error Rate for DATA-ACK packet exchange in the two links as shown in Fig. 2.16 by varying the payload size in the DATA packet. Essentially, the experiments let us compare the individual performances of the two links and further establish the MAC layer's role in enforcing fairness among the DTxs in accessing the channel.

2.7.5.1 Implemented MAC functions

The MAC header format for DATA and ACK shown in Fig. 2.17 and Fig. 2.18 respectively will aid in discussion of the MAC layer functions [1].

The DRx determines the DTx address from the MAC header of the received DATA packet and sends out an ACK addressed to that DTx. Furthermore, the DRx can reject DATA packets not addressed to it. Note that steps right from preamble detection, SFD detection, all the way up to reading into the IP address of the DTx from the MAC header, are carried out at the DRx, preceding the rejection of that DATA packet. On the other hand, the DTxs can determine the DRx from the

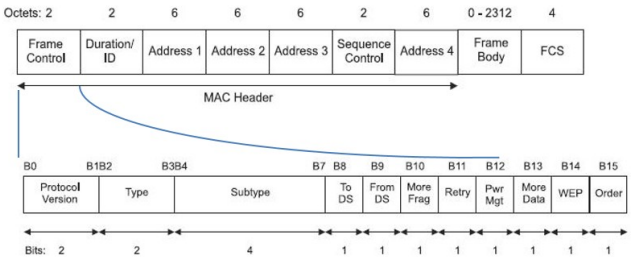


Figure 2.17: MAC Header - DATA packet [1]

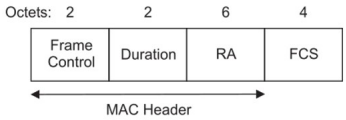


Figure 2.18: MAC Header - ACK packet [1]

MAC header of the received ACK and can go on to either accept or reject the ACK based on the IP Address. Previously, we had the DTx re-transmitting DATA packet only towards lost ACKs. Clearly, these are the MAC functions necessary for scaling up the system, enabled by reading into the MAC header of the DATA/ACK packet.

2.7.5.2 MAC parameters

We learned from our initial set of experiments that the DATA/ACK packet processing in the host machine takes significantly more time compared to time taken in transmitting a DATA packet. This is expected as most SDRs use a host computer for processing. Also, the SIFS duration, set in the order of microseconds in commercial products, imposes a time constraint in most SDRs that is difficult to achieve. The reason is that the latency for the signal to move back and forth from the radio to the host exceeds the SIFS duration requirements. The standard specifies the constants as follows: Slot-time = 20 μs, SIFS = 10 μs, DIFS = SIFS + 2 x Slot-time = 50 μs.

The experiments helped us fine-tune the DIFS duration (which the standard specifies be greater than SIFS), random backoff duration, and ACK timeout duration towards fewer packet collisions. As a result, we performed our experiments with DIFS duration, minimum contention window, and ACK timeout duration set at 0.75, 0.5, and 5.0 seconds, respectively.

2.7.5.3 Picking the Energy Threshold

Three node performance relies heavily on the energy detection step at both the DTxs. Accuracy of energy detection is critical and it requires the energy threshold be carefully picked at both the DTxs, enabling each DTx to back off as soon as they sense another DTx transmit, and subsequently transmit at the right instants of time, thereby keeping the packet errors and bi-directional link latency to a desired minimum. Additionally, it enforces fairness towards channel access among the DTxs.

The receive gain set at the DTx and the inter-node distances (1 meter in our experiments) affect the magnitude of the energy threshold. A value close to and slightly above the *noise floor* set as the energy threshold will not work as intended, as a power-cycle of the USRP changes it. Also, an energy threshold set at a large value might not allow the DTxs to sense each other transmitting due to rapidly fluctuating RF power output despite the AGC. Therefore, each DTx may not backoff at the right instants, leading to collisions at the DRx. However, by picking a small enough energy threshold, which is enough to detect signal energy over channel noise, we could make each DTx sensitive enough to sense the other DTx transmitting and backoff fairly well, thereby reducing packet retransmissions.

2.7.6 Three Node Performance: Experimental Results

Packet error rate and bi-directional link latency for DATA-ACK packet exchanges in the two links varying the payload size in the DATA packet are shown in Fig. 2.19 and Fig. 2.20, respectively. Four different payload sizes, 500, 1000, 1500, and 2000 octets, were used for the experiment to measure 3 node performance.

Smaller payload sizes correspond to smaller packets and decreased time that the DTx is occupying the channel whereas larger payload sizes increases the likelihood of packet collisions. The link latency and the packet error rate in the latter is bound to increase as larger packets incur higher processing delay at the DRx and more collisions necessitating increased packet retransmits.

2.7.6.1 Goodput

Goodput, a performance measure used in computer networks, is the rate at which useful information bits traverse a link. Goodput can be measured using equation (2.8),

$$Goodput = \frac{Total\ payload\ bits\ correctly\ decoded}{Average\ Bi\text{-}directional\ Link\ Latency} \tag{2.8}$$

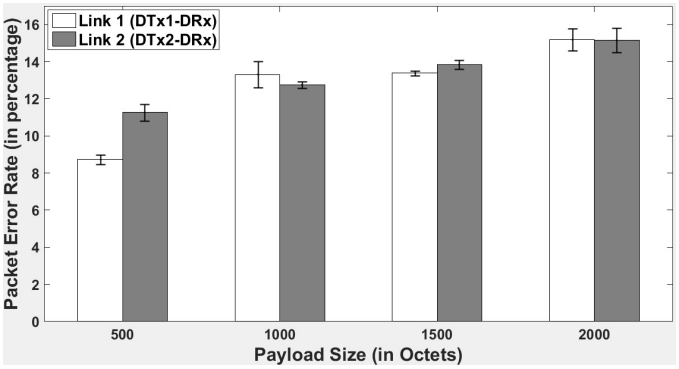


Figure 2.19: Three Node Performance - Packet Error Rate of the Links

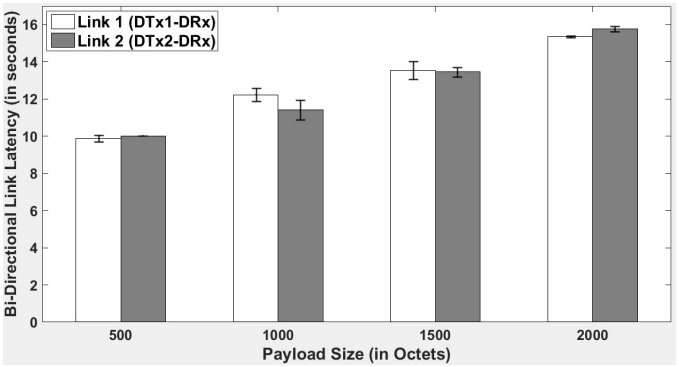


Figure 2.20: Three Node Performance - Bi-directional Link Latencies

The average Goodput of the two bi-directional links computed using (2.8) are shown in Table 2.3. Notice that the goodput increases with the payload size. The reason for this is that the combined PHY and MAC header occupies a decreased fraction of the entire DATA packet as the payload size increases.

Table 2.3: Average Goodput for Varying Payload Sizes

Payload Size (#Octets)	**Link 1 Goodput** (Kbps)	**Link 2 Goodput** (Kbps)
500	0.41	0.40
1004	0.66	0.70
1500	0.89	0.89
2004	1.05	1.02

In the three node system, when there is a symmetric increase in DIFS and backoff time at the two DTxs, then the system will remain fair with reduced contention, resulting in fewer packet errors. However, the goodput decreases as link latency increases. Also note that the standard specifies the DIFS and the contention window slot time be fixed constants.

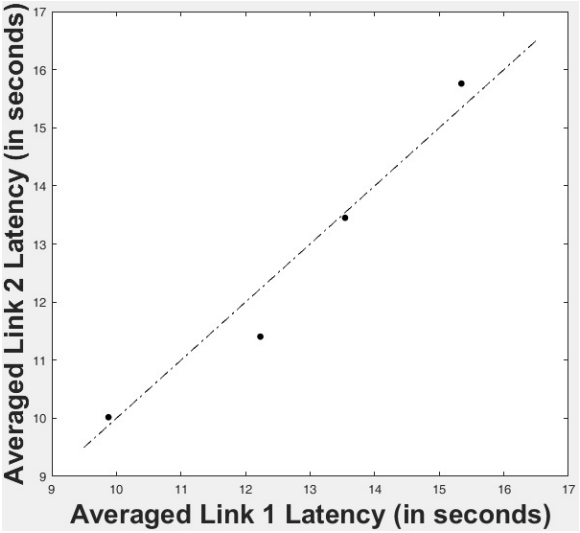


Figure 2.21: MAC Layer Fairness - Averaged Link Latencies

2.7.6.2 Fairness

The line shown in Fig. 2.21 is representative of an ideal system, in which the two DTxs access the channel equally often, such that their bi-directional link latencies are identical. Fairness is an important feature for the system to possess, and is brought about by the MAC protocol.

Notice that the latencies of the two links deviate by only a small amount from the ideal line for varying payload sizes. This result establishes the role and efficacy of the MAC layer in enabling and enforcing fairness among the two DTxs when accessing the common channel.

2.8 Virtual Carrier Sensing - RTS/CTS Signaling

In addition to the 802.11 DCF, the code base implements the optional virtual carrier sensing in CSMA/CA with the IEEE 802.11 RTS/CTS exchange, thereby bringing the implementation closer to being fully compliant with IEEE 802.11 MAC standard. The 802.11 standard specifies RTS/CTS signaling to address the hidden terminal problem. The exchanged RTS and CTS packets are standard compliant.

The DTx/DRx state machines have been modified to run the RTS-CTS-DATA-ACK over-the-air exchange (Refer to Fig. 2.22 and Fig. 2.23). The code is written in such a way that allows the user to pick from the two state machines so as to have either a DATA-ACK exchange or a RTS-CTS-DATA-ACK (RTS-to-ACK) exchange. The user can choose to either run the RTS-to-ACK exchange (default option) or the DATA-ACK exchange by setting a *vcs* flag at the DTx and DRx. The

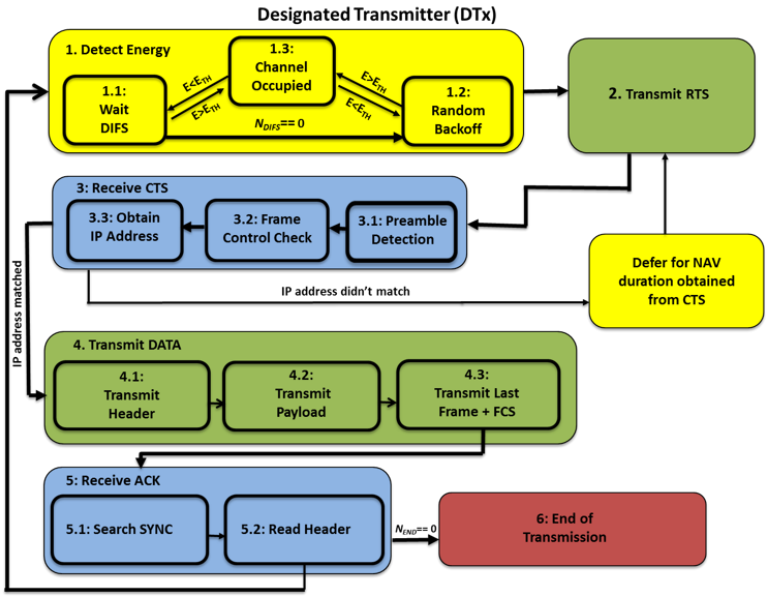


Figure 2.22: States for the Designated Transmitter (DTx)

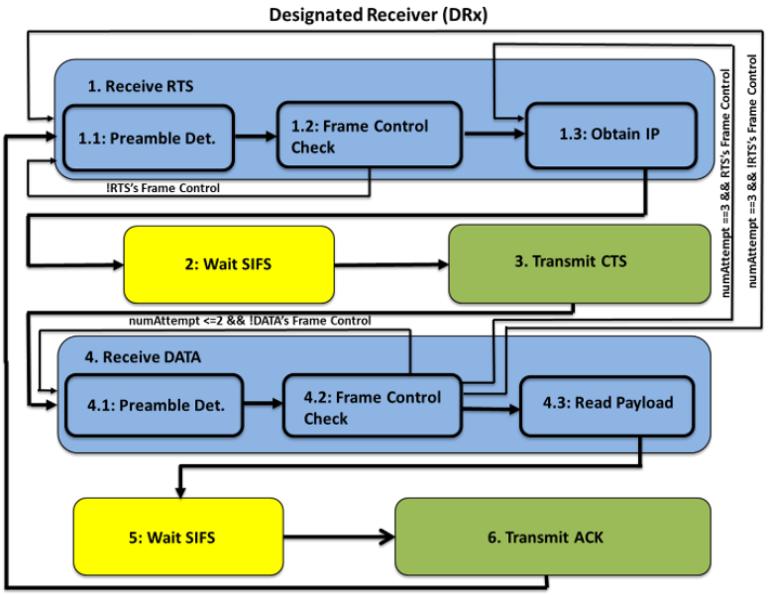


Figure 2.23: States for the Designated Receiver (DRx)

user also has the option to choose between Binary Exponential Back-off and Binary Linear Back-off algorithm to space out repeated retransmissions.

Virtual Carrier Sensing required us implement the logic to have the overhearing DTx back-off for the amount of time specified in the duration field in the RTS packet and test the implementation in a three-node setting. The DTx backs-off for Network Allocation Vector (NAV) specified in the

```
@ 8.79: 802.11b RTS Packet Transmitted.
CTS's Preamble Detected!
Returning To Detect Preamble..
CTS's Preamble Detected!
CTS's Frame-Control Readout:0001110011111111
CTS Addresed to DTx: 192.168.10.3
@28.53: 802.11b CTS Packet Received.

@30.32: 802.11b DATA Packet #1 Transmitted.
ACK's Preamble Detected!
ACK's Frame-Control Readout:0001110111111111
ACK Addresed to DTx: 192.168.10.3
@50.15: 802.11b ACK Packet #1 Received.

@ 2.32: 802.11b RTS Packet Transmitted.
CTS's Preamble Detected!
CTS's Frame-Control Readout:0001110011111111
CTS Addresed to DTx: 192.168.10.3
@22.12: 802.11b CTS Packet Received.

@23.92: 802.11b DATA Packet #2 Transmitted.
ACK's Preamble Detected!
Returning To Detect Preamble..
ACK's Preamble Detected!
ACK's Frame-Control Readout:0001110111111111
ACK Addresed to DTx: 192.168.10.3
@43.73: 802.11b ACK Packet #2 Received.

@ 2.31: 802.11b RTS Packet Transmitted.
CTS's Preamble Detected!
CTS's Frame-Control Readout:0001110011111111
CTS Addresed to DTx: 192.168.10.3
@22.12: 802.11b CTS Packet Received.

@23.92: 802.11b DATA Packet #3 Transmitted.
ACK's Preamble Detected!
ACK's Frame-Control Readout:0001110111111111
ACK Addresed to DTx: 192.168.10.3
@43.73: 802.11b ACK Packet #3 Received.

@ 2.31: 802.11b RTS Packet Transmitted.
CTS's Preamble Detected!
CTS's Frame-Control Readout:0001110011111111
CTS Addresed to DTx: 192.168.10.3
@22.12: 802.11b CTS Packet Received.

@23.92: 802.11b DATA Packet #4 Transmitted.
ACK's Preamble Detected!
ACK's Frame-Control Readout:0001110111111111
ACK Addresed to DTx: 192.168.10.3
@43.72: 802.11b ACK Packet #4 Received.

@ 2.31: 802.11b RTS Packet Transmitted.
CTS's Preamble Detected!
CTS's Frame-Control Readout:0001110011111111
CTS Addresed to DTx: 192.168.10.3
@22.12: 802.11b CTS Packet Received.
```

Figure 2.24: Screen log at DTx

```
DATA Packet Received!
DATA Packet's Frame Control Readout:0010000000000000
To Address read out from DATA: 192.168.20.2
From Address read out from DATA: 192.168.10.3
Ack Packet Transmitted for DATA Packet 1

RTS Packet Received!
RTS Packet's Frame Control Readout:0001101100000000
To Address read out from RTS: 192.168.20.2
From Address read out from RTS: 192.168.10.3
CTS Packet Transmitted for RTS Packet

DATA Packet Received!
DATA Packet's Frame Control Readout:0010000000000000
To Address read out from DATA: 192.168.20.2
From Address read out from DATA: 192.168.10.3
Ack Packet Transmitted for DATA Packet 2

RTS Packet Received!
RTS Packet's Frame Control Readout:0001101100000000
To Address read out from RTS: 192.168.20.2
From Address read out from RTS: 192.168.10.3
CTS Packet Transmitted for RTS Packet

DATA Packet Received!
DATA Packet's Frame Control Readout:0010000000000000
To Address read out from DATA: 192.168.20.2
From Address read out from DATA: 192.168.10.3
Ack Packet Transmitted for DATA Packet 3

RTS Packet Received!
RTS Packet's Frame Control Readout:0001101100000000
To Address read out from RTS: 192.168.20.2
From Address read out from RTS: 192.168.10.3
CTS Packet Transmitted for RTS Packet

DATA Packet Received!
DATA Packet's Frame Control Readout:0010000000000000
To Address read out from DATA: 192.168.20.2
From Address read out from DATA: 192.168.10.3
Ack Packet Transmitted for DATA Packet 4

RTS Packet Received!
RTS Packet's Frame Control Readout:0001101100000000
To Address read out from RTS: 192.168.20.2
From Address read out from RTS: 192.168.10.3
CTS Packet Transmitted for RTS Packet

DATA Packet Received!
DATA Packet's Frame Control Readout:0010000000000000
To Address read out from DATA: 192.168.20.2
From Address read out from DATA: 192.168.10.3
Ack Packet Transmitted for DATA Packet 5
```

Figure 2.25: Screen log at DRx

duration field in the CTS packet (variable *VCSSlot*).

When the DTx times out (upon waiting for an ACK) post sending a DATA packet, instead of retransmitting DATA, retransmits the RTS, essentially restarting the protocol. We noticed that this requirement demanded by the standard adds significant overhead to the latency in the order of a few seconds. Moreover, the size of the contention window is increased only for RTS/CTS loss and not in the case of DATA loss. The above requirement in turn forces the DRx to return to **Receive RTS** state whenever it fails to detect the DATA packet (say, for example, frame control does not check out, packet loss).

We tested the virtual carrier sensing in the two node node setting and it is stable. We observed a bidirectional link latency (RTS-to-ACK) averaging 11.5 seconds. In the screen logs, Fig. 2.24 and Fig. 2.25, obtained from the DTx and DRx host machines, observe that the frame control

bits read out from the RTS/CTS/DATA/ACK packets are different and as specified in the standard.

2.9 GUI for the Testbed Software

The testbed code is well documented and includes an user-friendly GUI guiding the user to conduct over-the-air experiments. The GUI makes it very easy for the user to reconfigure the important PHY/MAC parameters as required during run-time. The *initparameters.m* file has been entirely done away with as the updated GUI code now includes all the default parameter settings. Also, the test suite code gets called from the GUI code making the GUI ready to be used for over-the-air experimentation. The user can run the *GUIMain.m* script (once having all the extracted files in the same directory) to invoke the GUI. The GUI has been extensively tested and found to be stable.

The GUI overwrites the default variable settings with the user inputs (for variables displayed in the GUI). Fig. 2.26 and Fig. 2.27 display the tabs corresponding to the PHY and MAC parameters selected respectively. The tabs list the important parameters, along with the recommended ranges, sorted according to the frequency of use. The user is highly encouraged to set the value of the parameter variables in the recommended range as it reflects the feasible settings in which the system exhibits a stable operation.

The GUI can be used to carry out either of the two tasks: One, perform a demo involving transfer of images, and two, compute performance metrics for the over-the-air experiments. The purpose of demo is for the user to verify the correctness in the working of the system wherein the DTx(s) and DRx participate in exchange of image(s). The demo runs with the preset parameters and has the screen log, virtual carrier sensing settings turned off by default. The demo confirms the correct working of the system when the DRx can successfully discern and receive the image(s) without error from the DTx(s). Since the GUI allows the user compute important link layer performance metrics, it enables quick prototyping of new MAC protocols.

The layout of the GUI is designed keeping in mind the needs of the user. The *Help* button on top right of the GUI, when clicked, opens up a document in PDF that specifies the selection criteria for each parameter along with the default values. The *Screen Log* radio button turns on or off the verbose text that gets printed on to the command window when the code runs. The screen log is useful when debugging or testing the code. The *Mode* radio button decides the mode of operation of the connected node as either a DTx or a DRx. Once the user has set all the desired PHY and MAC parameters and is ready to run the code, the user can click the *START* button on the DTx(s) and DRx to begin the experiment.

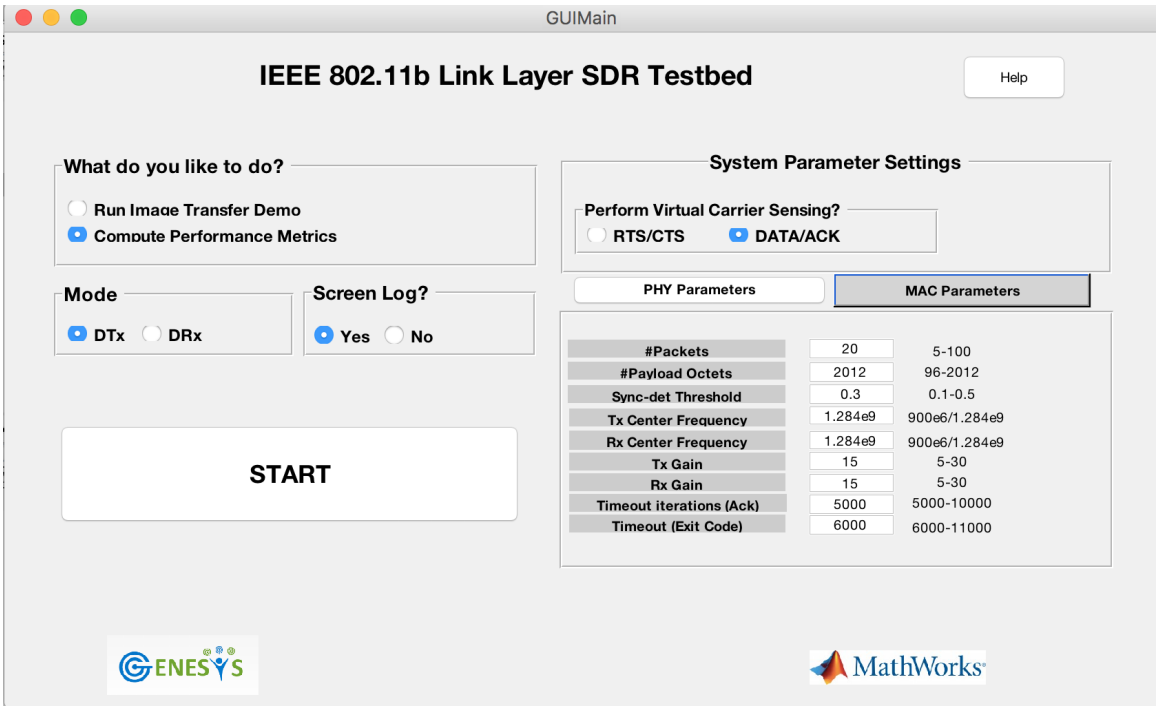


Figure 2.26: GUI with the important PHY parameters tab selected

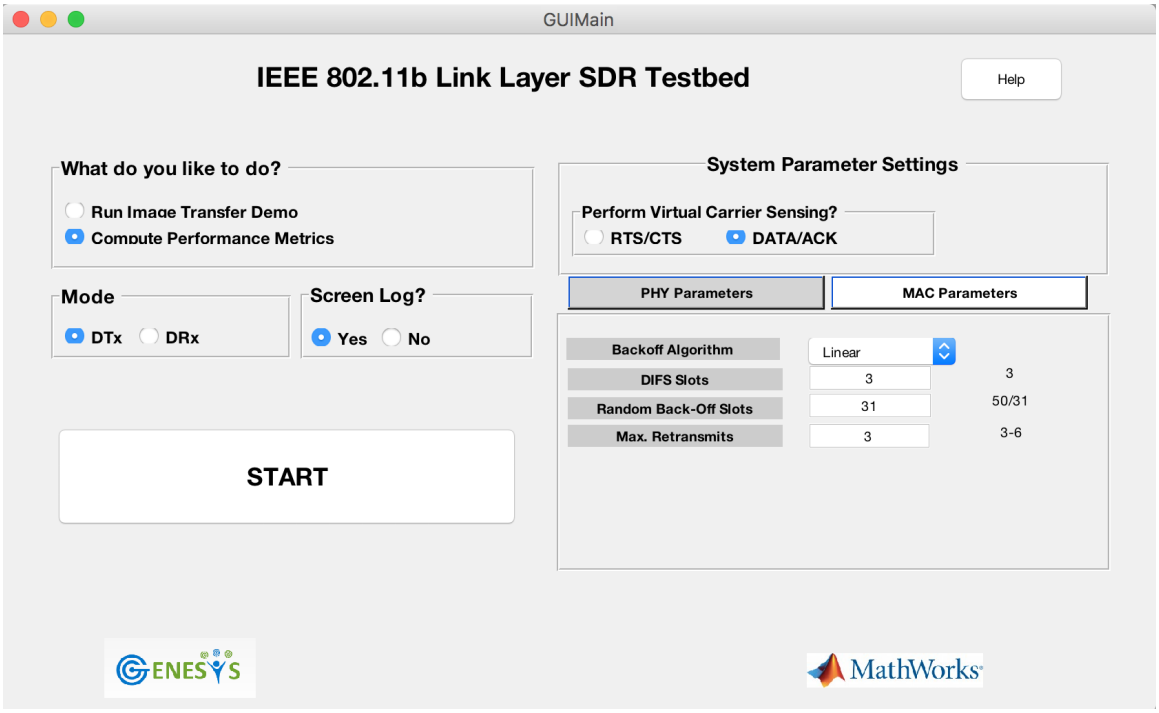


Figure 2.27: GUI with the important MAC parameters tab selected

Chapter 3

Software-Defined Network Controlled Spectrum Switching

When both types of wireless access become possible, there are several non-trivial tradeoffs that play a role in the SDN controller deciding which one of the two should be selected. The mmWave allows communication to commence at a greater separation distance, and thus can result in longer connected durations if there is relative motion between the nodes of the link. On the other hand, data exchange in the THz range may incur additional time for the node pair to be aligned in close proximity, but then it quickly ramps up by leveraging massive levels of bandwidth in such frequencies. There are additional considerations in this access selection problem, including the need for accommodating the channel-induced BER, which is unique for the choice of spectrum, and the amount of backlogged data to be delivered.

The *infrastructure* refers to installed roadside software-defined base station (SD-BS) typical of small cells that operate under the directive of a SDN controller. Unlike the traditional cellular network, where base stations are spaced out in a hexagonal grid pattern, SD-BS are opportunistically placed and their locations can largely be random.

As shown in Fig. 3.1, vehicles connect to SD-BS 1 for very short access times (in the order of seconds) during their motion. Considering the example of data backhauling, they may download the desired data at that location, and then upload the data via target tower 2 when proximity conditions allow.

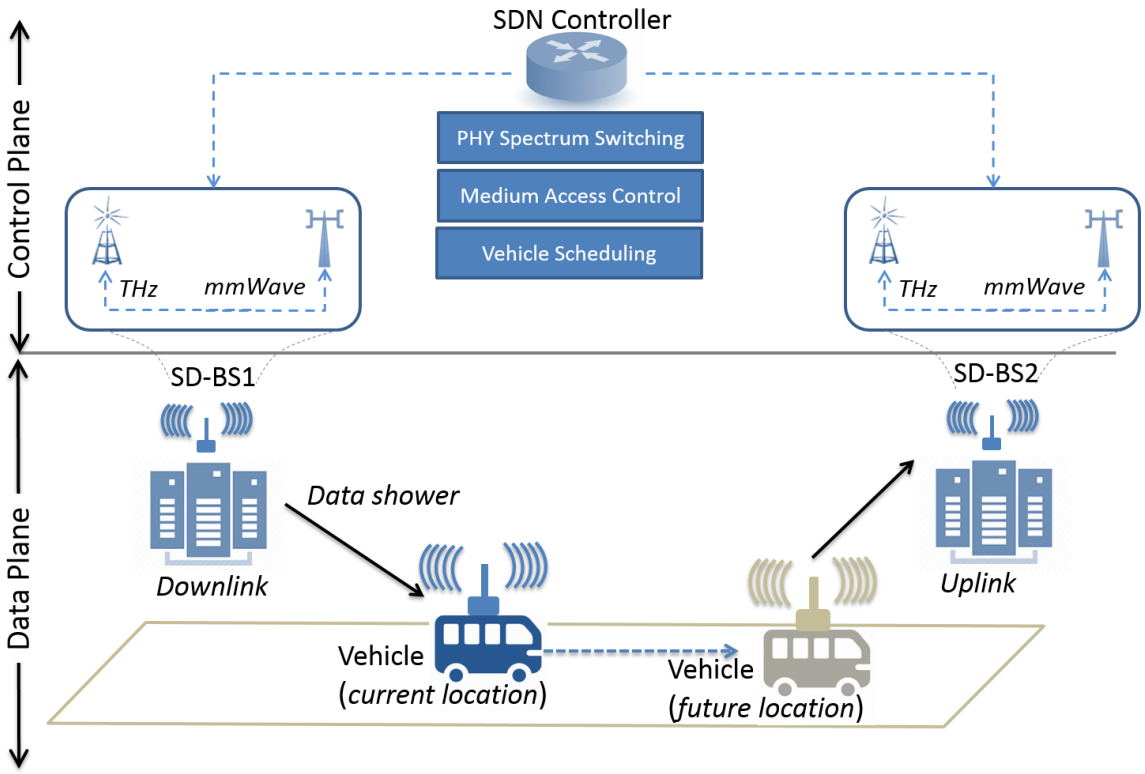


Figure 3.1: Network architecture for SDN controlled mmWave/THz connections.

3.1 Background and Architectural Assumptions

In this section, we first describe the main propagation characteristics of the mmWave and the THz bands used in the next sections of the paper, and the underlying architectural assumptions.

3.1.1 THz Channel Model

The signal propagation in the THz-band is mainly affected by molecular absorption, which results in both molecular absorption loss and molecular absorption noise [50–52]. In particular, the molecular absorption defines several transmission windows along the frequency scale with varying widths that are, to some extent, defined by the molecular composition of the medium.

The THz channel transfer function $H_{\text{THz}}(f,d)$ consists of a spreading loss function and a molecular absorption loss function given by [50, 53]:

$$H_{\text{THz}}(f,d) = \frac{c}{4\pi f d} e^{-\frac{k(f)d}{2}} e^{-j2\pi f \tau_{\text{LOS}}}, \tag{3.1}$$

where c denotes the speed of light, d stands for the distance between the transmitter and the receiver, and $\tau_{\text{LoS}} = d/c$ equals to the time-of-arrival of the line of sight (LoS) propagation. $k(f)$ is the frequency-dependent medium absorption coefficient that depends on the molecular composition of the transmission medium, i.e., the type and concentration of molecules found in the channel. Additional details for computing $k(f)$ and its effects on the THz propagation are reported in [50]. As in [50, 53], in this paper we do not account for Non-Line-of-Sight (NLoS) transmissions in the THz band due to the lack of experimental characterization. The few NLoS channel models existing to date [52] are mainly focused on the lower end of the THz band, i.e., 0.06 to 1 THz. We note that, by neglecting the NLoS opportunities, we underestimate the data shower in the THz band, i.e., we derive a lower bound on the achievable capacity in THz band. Moreover, we highlight that, by separately considering the NLoS propagation and outage event only for the mmWave communications, we incorporate in our model the fact that mmWave links are more robust than the THz links.

The molecular absorption determines not only the attenuation characteristics of the THz medium but also the noise. As described in [50, 53], the noise can be modeled as additive, colored Gaussian. In our work, we denote the distance-dependent noise power spectral density (p.s.d.) as $S_n(f, d)$. This model indicates that the THz channel is highly frequency-selective, and, in addition, the molecular absorption noise is non-white. Thus, the capacity can be obtained by dividing the total bandwidth B_{THz} into many narrow sub-bands of width Δf_i and summing the individual capacities [50, 52]. In fact, if the sub-band width is small enough, the channel appears as frequency-nonselective and the noise p.s.d. can be considered locally flat. Thus, by denoting with N_B the number of sub-bands and with $f_i, i \in \{1, \ldots, N_B\}$ the center frequency of the i-th sub-band, the resulting capacity in bits/s is given by:

$$\mathcal{C}_{\text{THz}}^{\text{LOS}}(d) = \sum_{i=1}^{N_B} \Delta f_i \log \left(1 + \frac{|H_{\text{THz}}(f_i, d)|^2 P_i}{\Delta f_i S_n(f_i, d)}\right) \tag{3.2}$$

where P_i is the power associated to the i-th sub-band accounting for the antenna directional gains, under the constraint $\sum_{i=1}^{N_B} P_i \leq P_s$ with P_s denoting the overall power, and $H_{\text{THz}}(f, d)$ is reported in (3.1). From (3.2), as pointed out in [50], the THz channel capacity depends on the frequency f_i of the electromagnetic wave, the transmission distance d, the molecular composition of the channel through $H_{\text{THz}}(f, d)$ and $S_n(f, d)$, and the powers P_i.

3.1.2 Architectural Assumptions

Software Defined Network (SDN)-based paradigm [54] is needed for seamless communication brought about by efficient resource sharing, thereby achieving high spectral efficiency, when involving multiple different wireless technologies, namely, LTE, mmWave and THz, and providing support for mobility. The software definition enhances rapid prototyping and reconfiguring of protocols thereby allowing for flexible processing on the hardware at runtime. The SDN control plane implementing the centralized PHY/MAC functions enables the physical layer switching, running MAC layer chunk size determination algorithms and the medium access scheduling for multiple vehicles.

The network architecture involves the SDN controller, providing the necessary abstraction to applications, moving vehicle and, a SD-BS that have three different connectivity options: (i) classical LTE bands used only for control packets when data communication occurs in mmWave band, (ii) mmWave transceivers used for data primarily, but in a secondary role, for sending control packets when THz channels are used for data, and (iii) short distance THz transceivers that may be used for one directional data transfers only, at a given time.

SD-BS can perform in-band signaling of real-time control messages, network status to the controller which in turn can feed back the control policies that best optimize for high link utilization [55] via standard interfaces like OpenFlow [56]. Since OpenFlow is capable of providing a uniform interface for different wireless standards it enables user mobility when moving across SD-BSs that support multiple wireless standards.

In addition, the mules are equipped with caches able to fetch big amount of data. This is a very reasonable assumption since the available memory capacity is considered the fastest growing and yet untapped network resource today due to the continuous progress of the storage technology.
• *Localization:* As the tower communication antennas are fixed and the vehicles today are generally equipped with GPS technology accurate to about a meter, we assume that there is full knowledge about the geolocation of both the mule and the tower antennas. Thus, the start/stop times for communication can be set accurately through beacons transmitted via currently existing and classical 802.11p/WAVE standards [57, 58]. Various techniques for tracking the sender/receiver during an ongoing communication have been proposed in THz channels, where a narrow-beam turns progressively thereby avoiding the need for frequent re-synchronization [51]. [59] uses out-of-band mmWave radar to aid beam alignment which significantly helps in reducing the beamsteering complexity. We account for the beamsteering complexity in the resulting overhead time. Finally,

Doppler shift arising in the vehicular speeds of interest may also be discounted as directional and steerable antennas can mitigate the impact of relative motion [60].

• *Need for mode switching:* We incorporate the fact that mmWave links are more robust than the THz links by separately considering the non line of sight (NLoS) propagation and outage event for the former only. Thus, the THz links can be in two different states: a LoS path is available or there is an outage. For the mmWave links, recent work suggests that the states of LoS, NLoS and outage are distinct [5]. Furthermore, experimental studies have demonstrated that the outage probability is small enough to be neglected, when the relative distance between the sender-receiver nodes is less than 200m [61]. Hence, when the distance between the mule and the tower is smaller than this threshold but greater than what is possible over THz link, the SDN controller prefers the mmWave link. Given the relative robustness of the mmWave link and high susceptibility of errors arising from NLoS in THz, the former can also be used as a separate control channel to return packet reception acknowledgments from the receiver to the sender that are communicating data in the THz channel. Our medium access protocol design assumes that both the mmWave and THz transceivers, albeit individually half-duplex and operating on entirely different frequency spectrum, can together be used to create a full duplex link.

• *Noise-limited communications:* It is worthwhile to note that, given the highly directional nature of the mmWave and THz access technologies, the ensuing communication is not interference-limited; rather it is noise-limited. Hence, the concept of medium access protocol refers to the selection and configuration of the mmWave and THz communication modes, so that the maximum data transfer can be achieved along with the assurance of an error recovery capability. This is discussed in detail in the next section.

3.2 Dynamic Spectrum Switching and Medium Access Protocol

The protocol described in this section is concerned with the selection and configuration of the mmWave and THz modes of communication at the SDN controller, so that (i) the maximum data transfer can be achieved, and (ii) error recovery can be assured.

3.2.1 Distance-dependent spectrum switching

Let the maximum distances between a pair of nodes at which communication becomes possible for the mmWave and the THz channels be given by $d_{\text{th}}^{\text{mm}}$ and $d_{\text{th}}^{\text{THz}}$, respectively. As discussed

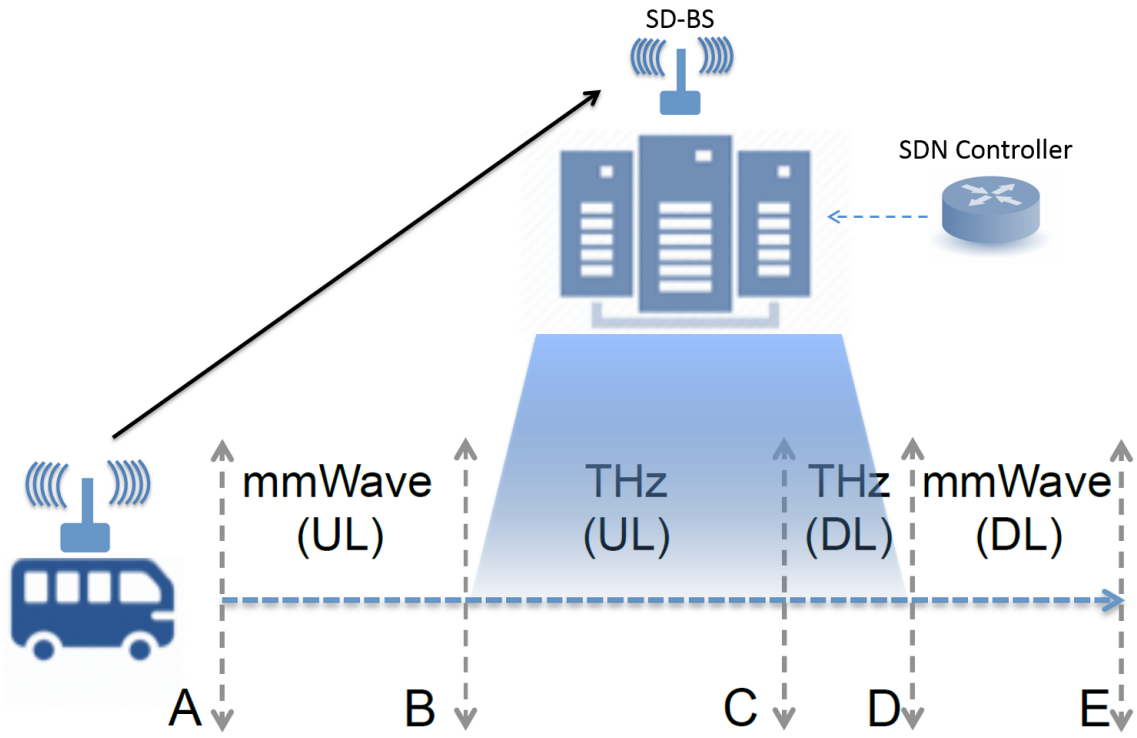


Figure 3.2: Selecting durations for uplink (A-C) and downlink (C-E)

in Sec. 4.1, $d_{\text{th}}^{\text{mm}} >> d_{\text{th}}^{\text{THz}}$. As THz bands allow transmission rates of several orders of magnitude higher than mmWave, we propose to use this mode whenever possible. Thus, the communicating node pair *always* switches to THz communication when the separation distance is less than $d_{\text{th}}^{\text{THz}}$, and to mmWave band when $d_{\text{th}}^{\text{THz}} \leq d \leq d_{\text{th}}^{\text{mm}}$. For example, in Fig. 3.2, the vehicle is moving from left to right, and in the process, reaching closer to the tower before pulling away again. The THz communication is only possible between B-D, and mmWave may be used both in A-B and D-E portions of the journey.

3.2.2 Uplink/downlink optimization

The overall data transfer between two physically separate towers requires downlink to the vehicle, the movement of the vehicle to the next location, followed by period of uplink. The vehicle repeats this cycle as it moves successively between the two infrastructure locations. As shown in Fig. 3.2, we divide the interaction time of the vehicle with a tower into distinct uplink (UL) followed by downlink (DL) phases. The ratio of the time taken to complete these two phases is not fixed; rather it is negotiated on the classical LTE channel ahead of the vehicle's arrival in the vicinity of the

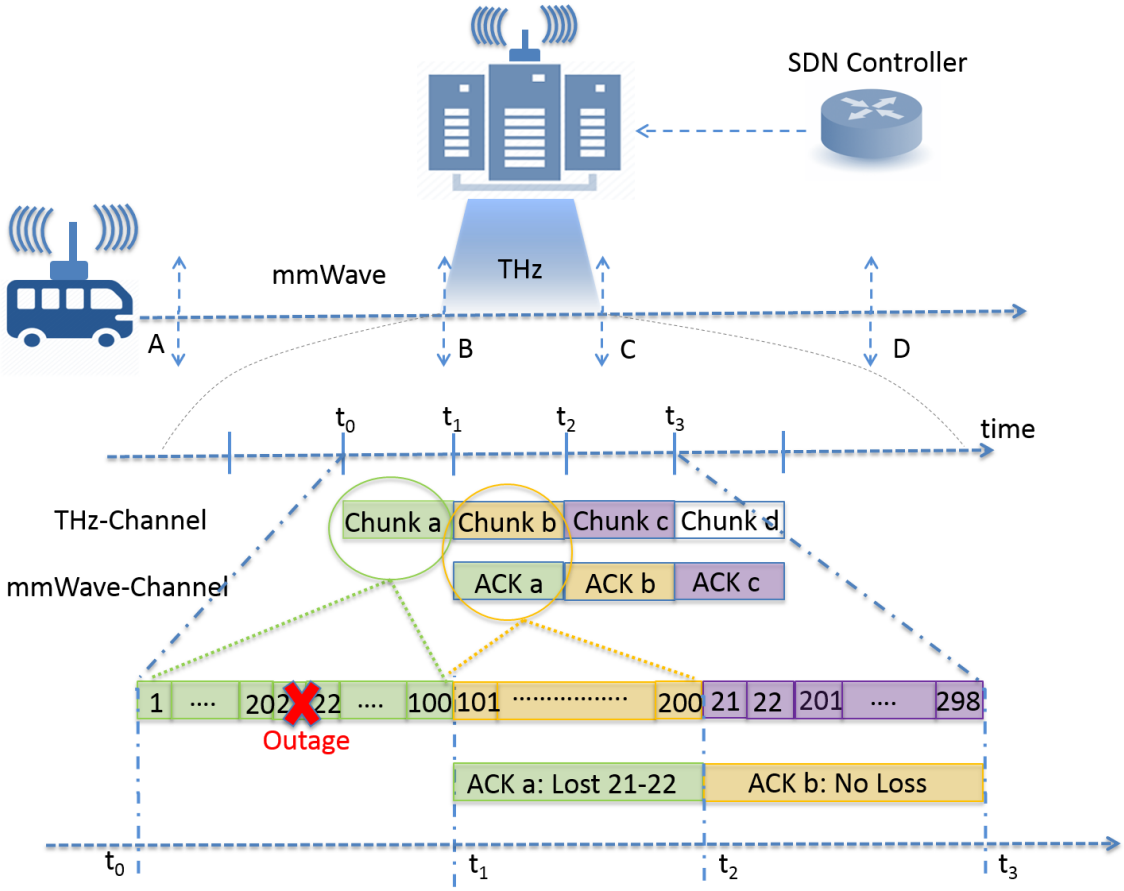


Figure 3.3: Protocol overview for the uplink phase when the distance between the mule and the tower antennas is smaller than the THz threshold. The data chunks are labeled with literals, whereas the numbers represent the packet IDs. A similar procedure applies when mmWave is used for data communication.

tower. This depends upon the path geometry-specific duration available for completing both the net UL/DL phases and the amount of backlogged data in either direction.

While a time-division like allocation for the UL and the DL phases allows us to clearly present the proposed framework, we note that the following analysis is neither affected by the directionality of the data shower, nor the assumption of time as a resource unit. Thus, our analytic derivations of capacity are valid for frequency-division or code-division resource allocations as well.

3.2.3 Throughput maximization, packet aggregation, and error recovery

According to the distance-dependent mode switching described in Section 3.2.1, the node pair always selects the best performing mode for data communication. To maximize the achievable throughput, we delegate the reverse path acknowledgments (ACKs) for the second-best option available at a given distance. Reversing the communication direction (for the ACK) introduces many challenges in completing a new round of beam training and alignment, synchronization etc. So instead, we retain the unidirectional flow of data in our scenario and delegate the slower and more reliable access technology for the returning ACKs. Specifically, when the THz link is active for one-way communication from sender to receiver, the mmWave link is used to report the ACKs from the receiver to the sender. When the mmWave is used for data communication, then the LTE link is used for ACKs reporting. Given that the transmission rate for data in each case is several order of magnitudes higher than that for the ACKs, the latter must be cumulative. We aggregate multiple data packets into a unit called as a *data chunk*, and each ACK cumulatively validates the packets within the chunk. Our design saturates both the access technologies as ACKs are smaller, but for each mmWave ACK, there are at least an order of magnitude more data packets sent in the forward THz channel. The size of a chunk needs to be chosen so that both the forward (i.e., data) and the reverse (i.e., ACK) channels remain saturated. In summary, data packets are sent continuously without any gaps, and they are periodically validated with cumulative ACKs received through the reverse channel to allow efficient error recovery. In fact, when some packets of a data chunk are received with errors, these errors are notified back to the sender through second-best performing channel so that the sender can selectively re-transmit the lost data, but this time in the best-performing channel. As shown in Fig. 3.3, once the ACK is received through the reverse channel at the sender side, the lost packets are identified and re-transmitted within the next data chunk[1], by prepending them to the new data. As a use-case, errors within the THz communication range are notified to the sender by using the mmWave band, allowing the sender to re-transmit in the active THz band. A similar process is used when ACKs are sent over LTE and data communication occurs over mmWave. In Fig. 3.3, two packets with IDs 21 and 22 belonging to the first chunk, say chunk a transmitted at time t_0, are lost due to an outage event. The sender becomes aware of such a packet lost at time t_2, upon the reception of the corresponding ACK a. Hence, it re-transmits these two packets with the third chunk. Missing ACKs are handled in a conventional manner, i.e., the entire packet train (i.e., entire chunk)

[1] Although the ACK processing delay could require that the lost packets will be re-transmitted at a some time slot in the future, we omit these particulars from Figure 3.3 for the sake of simplicity.

represented by that ACK will need to be re-sent in the forward channel. We assume constant chunk durations in this work, and propose to investigate the effect of dissimilar and derive optimal chunk intervals in future investigations.

3.3 Capacity Modeling

In this section, we theoretically derive the effective channel capacity achievable through the proposed protocol, by exploiting both the mmWave and THz communications. In particular, the theoretical analysis accounts for the impact of relative distances and channel propagation conditions, noise and signal power.

3.3.1 Capacity Formulation

We first introduce some definitions that will be used in the following analysis. Specifically, let us denote with $\mathcal{R}_{mm} \triangleq (d_{\text{th}}^{\text{THz}}, d_{\text{th}}^{\text{mm}}]$ the distance interval in which a mmWave communication is established. Similarly, we denote $\mathcal{R}_{THz} \triangleq (0, d_{\text{th}}^{\text{THz}}]$ as the distance interval in which a THz communication is established.

Definition 1. $P(d)_{\text{mm}}^{\text{LoS}}$ denotes the probability of having a LoS connection between the transmitter and the receiver in the mmWave band, when their relative distance is d. $P(d)_{\text{mm}}^{\text{NLoS}}$ denotes instead the probability of having a NLoS connection between the transmitter and the receiver in the mmWave band, when their relative distance is d.

Clearly such probabilities depend also on the geography of the considered network area, including building density and other natural/man made structures.

Definition 2. $P(d)_{\text{THz}}^{\text{LoS}}$ denotes the probability of having a LoS connection between the transmitter and the receiver in the THz band, when their relative distance is d.

We recall that the LTE interface is only used for ACKs and the mmWave link is used for data whenever the distance is $d_{\text{th}}^{\text{THz}} < d \leq d_{\text{th}}^{\text{mm}}$. Also, the mmWave interface is used for ACKs and the THz link for data whenever inter-node distance is $d \leq d_{\text{th}}^{\text{THz}}$. Hence, at a given relative distance d, the capacity $\mathcal{C}(d)$ available for transmitting data is given by:

$$\begin{aligned}\mathcal{C}(d) = &\left[\mathcal{C}(d)_{\text{mm}}^{\text{LoS}} P(d)_{\text{mm}}^{\text{LoS}} + \mathcal{C}(d)_{\text{mm}}^{\text{NLoS}} P(d)_{\text{mm}}^{\text{NLoS}}\right] 1_{\mathcal{R}_{mm}}(d) + \\ &+ \left[\mathcal{C}(d)_{\text{THz}}^{\text{LoS}} P(d)_{\text{THz}}^{\text{LoS}}\right] 1_{\mathcal{R}_{THz}}(d)\end{aligned} \tag{3.3}$$

where $P(d)^{\text{LoS}}_{\text{mm}}$, $P(d)^{\text{NLoS}}_{\text{mm}}$ and $P(d)^{\text{LoS}}_{\text{THz}}$ are defined in Definitions 1 and 2, respectively, and $1_{\mathcal{R}_{mm}}(d)$ is the indicator function of $\mathcal{R}_{mm}$ given by:

$$1_{\mathcal{R}_{mm}}(d) = \begin{cases} 1, & d \in \mathcal{R}_{mm} \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

Similarly, $1_{\mathcal{R}_{THz}}(d)$ is the indicator function of $\mathcal{R}_{THz}$. In (3.3), $\mathcal{C}(d)$ represents the available channel capacity at a given distance d, qualified further with appropriate subscripts (mm, THz) depending upon which access mode is used, and superscripts (LoS, NLoS) depending upon which of these propagation conditions exist.

We stress that equation (3.3) is valid regardless of the adopted models for the channel capacities and the probabilities of having LoS and NLoS paths. In the following, we expand (3.3) further by considering some specific models for the channel capacity and the LoS and NLoS probabilities.

3.3.2 Case I - mmWave links

First, regarding the mmWave capacity, when LoS link is available at a given relative distance d, we adopt the Shannon model used in [61]:

$$\mathcal{C}(d)^{\text{LoS}}_{\text{mm}} = B_{\text{mm}} \log\left(1 + \gamma_{\text{mm}}(d)\right), \tag{3.5}$$

where $\gamma_{\text{mm}}(d)$ denotes the average SNR, accounting for the directional antenna gains, observed at the distance d in the mmWave spectrum of width B_{mm}.

Second, when the NLoS link is available at a given relative distance d, we adopt the widely-used model that scales the LoS SNR with a factor Δ [61]:

$$\mathcal{C}(d)^{\text{NLoS}}_{\text{mm}} = B_{\text{mm}} \log\left(1 + \frac{\gamma_{\text{mm}}(d)}{\Delta}\right) \tag{3.6}$$

Regarding the LoS and NLoS probabilities for mmWave communications, we adopt the models proposed in [5, 61], since they were validated through experimental data. Specifically:

$$P(d)^{\text{LoS}}_{\text{mm}} = \left(1 - P(d)^{\mathcal{O}}_{\text{mm}}\right) e^{a_{\text{LoS}} d} \tag{3.7}$$

$$P(d)^{\text{NLoS}}_{\text{mm}} = 1 - P(d)^{\mathcal{O}}_{\text{mm}} - P(d)^{\text{LoS}}_{\text{mm}} \tag{3.8}$$

where $P(d)^{\mathcal{O}}_{\text{mm}}$ denotes the outage probability that can be computed as [5, 61]:

$$P(d)^{\mathcal{O}}_{\text{mmW}} = \max\left(0, 1 - e^{-a_{\mathcal{O}} d + b_{\mathcal{O}}}\right), \tag{3.9}$$

In (3.8) and (3.9), a_{LoS}, $a_{\mathcal{O}}$ and $b_{\mathcal{O}}$ are values empirically derived [5, 61].

3.3.3 Case II - THz links

Differently from mmWave communications, experimental data validating an outage distribution model is not available for the THz band. Hence, by adopting a similar approach described in [5], we assume an exponential distribution[2] for the outage in THz band as a function of the average SNR $\gamma_{\text{THz}}(d)$ at the distance d [62]:

$$P(d)^{\mathcal{O}}_{\text{THz}} \triangleq 1 - P(d)^{\text{LOS}}_{\text{THz}} = 1 - e^{-\gamma_{\text{th-THz}}/\gamma_{\text{THz}}(d)} \tag{3.10}$$

where $\gamma_{\text{th-THz}}$ denotes the minimum SNR required for establishing the THz link that depends also on the sensitivity of the receiver [62]. Using the THz channel model described in Section 4.1.1, $\gamma_{\text{THz}}(d)$ can be evaluated as[3]:

$$\begin{aligned}\gamma_{\text{THz}}(d) &= \frac{\int_{B_{\text{THz}}} S_t(f)|H_{\text{THz}}(f,d)|^2 df}{\int_{B_{\text{THz}}} S_n(f,d) df} = \\ &= \sum_{i=1}^{N_B} \frac{|H_{\text{THz}}(f_i,d)|^2 S_t(f_i)}{S_n(f_i,d)}\end{aligned} \tag{3.11}$$

Finally, regarding the THz capacity $\mathcal{C}(d)^{\text{LoS}}_{\text{THz}}$, it has been analyzed in Section 4.1.1 and its expression is given in (3.2).

3.3.3.1 Data Shower Bulk

Through the analysis described in the previous sections, we derived a closed-form expression for the channel capacity for a given distance d when our network design is adopted. Using these results, we derive the maximum average number of data bits exchanged between the transmitter and the receiver in Proposition 1. We refer to this average number as *data shower bulk*. We also provide in Corollary 1 a closed-form expression for the *data shower bulk* under the hypothesis of constant-speed straight trajectory. Before we proceed with this analysis, we list some preliminary definitions.

Definition 3. ϵ_s^{mm} denotes the time spent at the start of the mmWave communication to synchronize the transmitter and the receiver. This time is needed to calibrate the transceivers at a finer granular level, as observed in Section 4.1, despite the assumption of steerable antennas. Similarly, ϵ_s^{THz} denotes the time spent at the start of the THz communication to synchronize the transmitter and the

[2]This assumption is not restrictive, since the results derived within the paper continue to hold by simply adopting a different outage probability model.

[3]In Section 4.1.1 we assumed an ideal low-pass receiver filter.

receiver at a finer granular level. Finally, ϵ_{tr} denotes the time spent in switching from transmitting mode to receiving mode and vice versa.

The time ϵ_{tr} takes into account not only the effective time for mode selection, but also a guard time to handle possible burst errors arising from the previous phase. The setting of such a parameter is beyond the scope of this paper, but does raise interesting design possibilities that we intend to explore in our future work. We observe that the optimization of such a parameter should account for the allocation strategy chosen for the UL and the DL phase, as well as the length of the packet chunks which in turns depend on both the channel conditions and the delay propagation.

Definition 4. t_{in} and t_{out} denote the starting and the ending time of a contact event, respectively, i.e., the first and the last time instant in which the transmitter and the receiver could establish and sustain either a mmWave or THz link in a one-way journey.

Proposition 1. *The data shower bulk transferred by adopting the proposed architecture is given by:*

$$\overline{n} = \int_{t_{in}}^{t_{out}} \mathcal{C}(d(t))dt \tag{3.12}$$

where $d(t)$ denotes the transmitter-receiver relative distance at time t and the capacity $\mathcal{C}(d(t))$ is given in (3.3)*.*

Proof. See Appendix A □

Remark 1. We note that the time interval $[t_{\text{in}}, t_{\text{out}}]$ can be characterized by a sequence of time-separated contact periods, as a consequence of the mule moving repetitively in and out of the communication range due to the street topology constraints. Nevertheless, the time instants belonging to the considered time interval at which the distances $\{d(t)\}$ do not range in $\{\mathcal{R}_{mm} \bigcup \mathcal{R}_{THz}\}$ do not contribute to the transferred bits $\overline{n}$, since the capacity $\mathcal{C}(d(t))$ is null according to (3.3).

Remark 2. The data shower bulk $\overline{n}$ derived in Proposition 1 constitutes an upper bound of the layer-2 throughput achievable by adopting the proposed architecture. In fact, (3.12) does not account for the synchronization overhead associated with the times ϵ_s^{mm} and ϵ_s^{THz}, as well as the switching overhead associated with the time $\epsilon_{tr}^{\text{THz}}$. Furthermore, the throughput depends on a number of physical-realization parameters, such as the adopted channel code, the adopted modulation technique, as well as the synchronization techniques and the mode switching procedure.

In the following we derive in Corollary 1 a strict bound for the data shower bulk, under the hypothesis of uniform straight movement from A to E as depicted in Fig. 3.2. To this aim, let us

denote with the α the angle formed by: (i) the distance $d_{\text{th}}^{\text{mm}}$ between the mule with the tower at time t_{in}, and (ii) the direction of the movement.

Corollary 1. *The data shower bulk transferred by adopting the proposed architecture under the hypothesis of constant-speed straight-trajectory with average speed v is given by:*

$$\overline{n} = \left(\frac{2d_{th}^{mm} \cos \alpha}{v} - \epsilon_s^{mm} - \epsilon_s^{THz} - \epsilon_{tr} \right) \frac{\int_{d_{min}}^{d_{th}^{mm}} \mathcal{C}(\eta) d\eta}{d_{th}^{mm} - d_{min}} \tag{3.13}$$

where d_{min} is the minimum distance between the antennas of the mule and the BS during the movement, $\mathcal{C}(d)$ is given in (3.3) *and ϵ_s^{mm}, ϵ_s^{THz}, ϵ_{tr}^{THz} are defined in Definition 3.*

Proof. See Appendix B □

The data shower bulk $\overline{n}$ derived in Corollary 1 constitutes a stricter bound than (3.12). In fact, in (3.13) we explicit some time overhead through ϵ_s^{mm}, ϵ_s^{THz}, $\epsilon_{tr}^{\text{THz}}$.

3.4 Multi-vehicle Scheduling

The work so far covers the capacity formulation for a single vehicle exchanging data with one roadside infrastructure location. However, multiple vehicles $\mathcal{V} = \{1, \ldots, V\}$ may also pass through the same region concurrently. This requires the SDN controller scheduling them at different time instants (there is only one mmWave/THz transceiver at the roadside location) so that all their cumulative bandwidth needs are satisfied. The scheduling time is dependent also on the location of the vehicles at that instant, which in turn influences whether the mmWave or the THz link is active.

Considering that the entire time horizon is composed of slots of duration T, and let D_v with $v \in \mathcal{V}$ denote the number of bits uploaded/downloaded to/from the infrastructure tower that is bounded by $\bar{n}$ derived earlier in (3.12). Further, let the number of vehicles in $\mathcal{V}$ that are close enough to the tower so that a communication (either mmWave or THz) link can be established in a given time slot k be given as $\mathcal{V}^k \subseteq \mathcal{V}$. Thus,

$$\mathcal{V}^k = \{v \in \mathcal{V} : d_v^k \in \mathcal{R}_{\text{mm}} \cup \mathcal{R}_{\text{THz}}\} \tag{3.14}$$

with d_v^k denoting the maximum distance of the v-th vehicle from the tower during time slot k, and $\mathcal{R}_{\text{mm}}$ and $\mathcal{R}_{\text{THz}}$ defined in Sec. 3.3.1. Note this implies that a vehicle belongs to $\mathcal{V}^k$ if and only if its distances in the entire time slot belong to $\mathcal{R}_{\text{mm}} \cup \mathcal{R}_{\text{THz}}$. In the following, for the sake of simplicity and without loss of generality, we assume $\bigcup_k \mathcal{V}^k = \mathcal{V}$.

3.4.1 Scheduling Problem Formulation

We devise an optimization problem to select which slots must be assigned to each vehicle $v \in \mathcal{V}$ with the objective of maximizing the total number of bits exchanged between the infrastructure tower and the vehicles within the time horizon under the following constraints:

i) the total number of bits exchanged with the v-th vehicle within a one-way journey does not exceed D_v;

ii) at most one vehicle is scheduled during each time slot.

Constraint (i) avoids sub-optimal scheduling, i.e., it avoids assigning a time slot to a vehicle that already completed its communication needs (represented by D_v). Constraint (ii) accounts for the THz/mmWave mode selection described in Sec. 4.2. Although two technologies (mmWave/THz) can be concurrently used in a given time slot, only one vehicle can be scheduled in each time slot, since we exploit the second-best technology for reverse path acknowledgments.

By denoting with N the total number of exchanged bits and with N_v, $v \in \mathcal{V}$, the number of bits exchanged with the v-th vehicle within a one-way journey, we can reformulate the considered problem as follows:

$$\text{given } k_s \leq k_e: \quad \mathcal{V}^{k_s-1} = \mathcal{V}^{k_e+1} = \emptyset \tag{3.15}$$

$$\mathcal{V}^k \neq \emptyset, \forall\, k \in \mathcal{K} \tag{3.16}$$

$$\underset{\substack{\{\phi_v^k\}_{k\in\mathcal{K},}\\ v\in\mathcal{V}}}{\text{maximize}} \quad N \tag{3.17}$$

$$\text{subject to} \quad N_v \leq D_v, \forall\, v \in \mathcal{V} \tag{3.18}$$

$$\sum_{v\in\mathcal{V}^k} \phi_v^k = 1, \forall\, k \in \mathcal{K} \tag{3.19}$$

with $\mathcal{K} \triangleq \{k_s, \ldots, k_e\}$ denoting the set of time slots and ϕ_v^k denoting the indicator function mapping each vehicle with a time slot, i.e., $\phi_v^k = 1$ if the v-th vehicle is scheduled within the $k-$th time slot and $\phi_v^k = 0$ otherwise.

(3.16) guarantees that, during each time slot of the considered time horizon, there exists at least one vehicle in connection with the tower. In fact, an *empty* time slot represents a separation between different journeys, which need to be individually optimized due to the finite cache sizes. This is accounted for in (3.15).

We derive in Proposition 2, the closed-form expression of N and N_v, by accounting for the time spent by the tower and the vehicle to establish a physical link. This time depends on vehicle-specific parameters, such as the adopted antenna beamforming algorithm. To abstract the scheduling problem from underlying dependencies, we accumulate all such coordination overheads within $T_v^o > 0$, which we refer to as *overhead time*.

Proposition 2. *The total number of bits N exchanged between the tower and the vehicles is equal to*

$$N = \sum_{v \in \mathcal{V}} N_v \tag{3.20}$$

where

$$N_v = \sum_{k \in \mathcal{K}} \phi_v^k n_v^k \tag{3.21}$$

and

$$n_v^k = \int_{(k-1)T + \chi_v^k T_v^O}^{kT} C_v(d_v(t))dt \tag{3.22}$$

$$\chi_v^k = \begin{cases} 1 & \textit{if } \phi_v^k - \phi_v^{k-1} = 1 \\ 0 & \textit{otherwise} \end{cases} \tag{3.23}$$

with $C_v(d_v(t))$ given in (3.3) and $\phi_v^0 \triangleq 0$.

Proof. See Appendix C □

Remark 3. The scheduling problem is NP-hard, since: i) the variables ϕ_v^k denoting the scheduling-state of the v-th vehicle at time slot k have integer values (actually, binary); ii) the presence of the overhead time T_O within the integral in (3.22). In fact, the time complexity grows with the total number of possible solutions, i.e., $\mathcal{O}\left(V^K\right)$, where $V = |\mathcal{V}|$ is the number of vehicles and $K = |\mathcal{K}|$ is the number of time slots. As an example, when $V = 4$ and $K = 20$, it results $V^K = 4^{20} \simeq 10^{12}$. Hence, we design a greedy scheduling algorithm (see Algorithm 1), which has polynomial-time complexity.

Algorithm 1 Greedy Scheduling Algorithm

$\phi_v^k = 0, \forall\, k \in \mathcal{K} \wedge v \in \mathcal{V}$
for all $k \in \mathcal{K}$ **do**
  if $|\mathcal{V}^k| == 1$ **then**
    // Only 1 vehicle in contact
    $v = \mathcal{V}^k[1]$
    // with $A[n]$ denoting the n-th element of array A
    $\phi_v^k = 1$
    $\mathcal{K} = \mathcal{K} \setminus \{k\}$
    $D_v = D_v - \tilde{n}_v^k$
    if $D_v \leq 0$ **then**
      $\mathcal{V}^k = \mathcal{V}^k \setminus \{v\}\ \forall\, k \in \mathcal{K}$
    end if
  end if
end for
for all $k \in \mathcal{K}$ **do**
  if $\mathcal{V}^k == \emptyset$ **then**
    $\mathcal{K} = \mathcal{K} \setminus \{k\}$
    // Remove empty slot
  end if
end for
while $\mathcal{K} \neq \emptyset$ **do**
  $k_t, v_t \leftarrow$ Algorithm 2
  $\phi_{v_t}^{k_t} = 1$
  $D_{v_t} = D_{v_t} - \tilde{n}_{v_t}^{k_t}$
  $\mathcal{K} = \mathcal{K} \setminus \{k_t\}$
  if $D_{v_t} \leq 0$ **then**
    for all $k \in \mathcal{K} : v \in \mathcal{V}^k$ **do**
      $\mathcal{V}^k = \mathcal{V}^k \setminus \{v_t\}$
      if $\mathcal{V}^k == \emptyset$ **then**
        $\mathcal{K} = \mathcal{K} \setminus \{k\}$
      end if
    end for
  end if
end while

Algorithm 2 Inner Procedure

1: $\mathcal{K}_t = \arg\max_{k \in \mathcal{K}} \left\{ \max_{v \in \mathcal{V}^k} \{\tilde{n}_v^k\} \right\}$
2: **if** $|\mathcal{K}_t| == 1$ **then**
3: $k_t = \mathcal{K}_t[1]$
4: $v_t = \arg\max_{v \in \mathcal{V}^{k_t}} \{\tilde{n}_v^{k_t}\}$
// if argmax returns multiple items, pick one at random
5: **else**
6: **for all** $i = 1 : |\mathcal{K}_t|$ **do**
7: $\mathcal{V}_t[i] = \arg\max_{v \in \mathcal{V}^{\mathcal{K}_t[i]}} \{\tilde{n}_v^{\mathcal{K}_t[i]}\}$
8: **end for**
9: $i_t = \arg\min_{i=1,\ldots,|\mathcal{K}_t|} \left\{ \sum_{v \in \mathcal{V}^{\mathcal{K}_t[i]} \setminus \{\mathcal{V}_t[i]\}} \{\tilde{n}_v^{\mathcal{K}_t[i]}\} \right\}$
10: $k_t = \mathcal{K}_t[i_t]$
11: $v_t = \mathcal{V}_t[i_t]$
12: **end if**
13: **return** k_t, v_t

3.4.2 Explanation of Algorithm 1

The greedy algorithm works by first (lines 2-13) computing the sets of slots during which only one vehicle can establish a communication link with the tower. For each such slot, a given vehicle is scheduled if the constraint described in equation (3.18) is satisfied (lines 8-11). As soon as a vehicle completes its communication needs, it is excluded (line 10) from all the remaining slots. As a consequence, a slot may become *empty*, i.e. the number of vehicles to be scheduled could become zero. Lines-14-19 remove such empty slots from $\mathcal{K}$.

In lines 20-33, the algorithm schedules vehicle v_t at time slot k_t, if this choice maximizes the number of exchangeable bits without accounting for the scheduling overhead, i.e.:

$$\phi_{v_t}^{k_t} = 1 \iff \tilde{n}_{v_t}^{k_t} = \max_{k \in \mathcal{K}} \left\{ \max_{v \in \mathcal{V}^k} \{\tilde{n}_v^k\} \right\} \tag{3.24}$$

where

$$\tilde{n}_v^k = \int_{(k-1)T}^{kT} C_v(d_v(t))dt \tag{3.25}$$

Specifically, at line 21 k_t and v_t are computed through Algorithm 2, and the remaining lines schedule the vehicle and satisfy the constraint given in (3.18).

Table 3.1: Parameter Setting

mmWave parameter	Value
f_c: carrier frequency	73 GHz
Δf_c: uplink/downlink shared bandwidth	1 GHz
α: path-loss intercept least squares fit	LoS: 69.8, NLoS: 82.7
β: path-loss slope least squares fit	LoS: 2 - NLoS: 2.69
P_{tx}: transmit power	30 dBm
G: directional antenna gain	27 dB
Noise power	-87 dBm
Noise figure	5 dB
$d_{\text{th}}^{\text{mm}}$: operational distance	200 m
$1/a_{\text{LoS}}$: LoS state probability parameter	37 m
$1/a_{\text{O}}$: outage state probability parameter	45.5 m
$1/b_{\text{O}}$: outage state probability parameter	3.3

THz parameter	Value
$k(f)$: frequency-dependent coefficient	$[2 \cdot 10^{-6} - 3 \cdot 10^{1}]cm^{-1}$
f_c: carrier frequency	0.85 THz
Δf_c: uplink/downlink shared bandwidth	0.1 THz
P_{tx}: transmit power	$0 - 20$ dBm
G: directional antenna gain	27 dB
$d_{\text{th}}^{\text{THz}}$: operational distance	10 m

We note that Algorithm 2, through lines 5-12, accounts for the case in which multiple feasible choices for maximizing the number $\tilde{n}_v^k$ of transferred bits is possible, i.e., there exists multiple time slots in which the same maximum $\tilde{n}_{v_t}^{k_t}$ is achieved. In such a case, line 9 selects the time slot in which the lowest communication opportunities (i.e., the lowest average number of exchangeable bits) are available to the remaining vehicles.

Finally, we note that the constraint given in equation (3.19) is satisfied with lines 7 and 24. Hence, Algorithm 1 computes a valid (admissible) solution for the considered scheduling problem.

Remark 4. The time complexity of the greedy algorithm given in Algorithm 1 is $\mathcal{O}(V \cdot K^2)$. Specifically, Algorithm 1 exhibits a polynomial complexity, which grows quadratically with the number of time slots and linearly with the number of vehicles. Clearly, this is an attractive feature since it assures the computational practicability of the algorithm. With reference to the example given in Remark 3, it results $V \cdot K^2 \simeq 2 \cdot 10^3 \ll 10^{12}$.

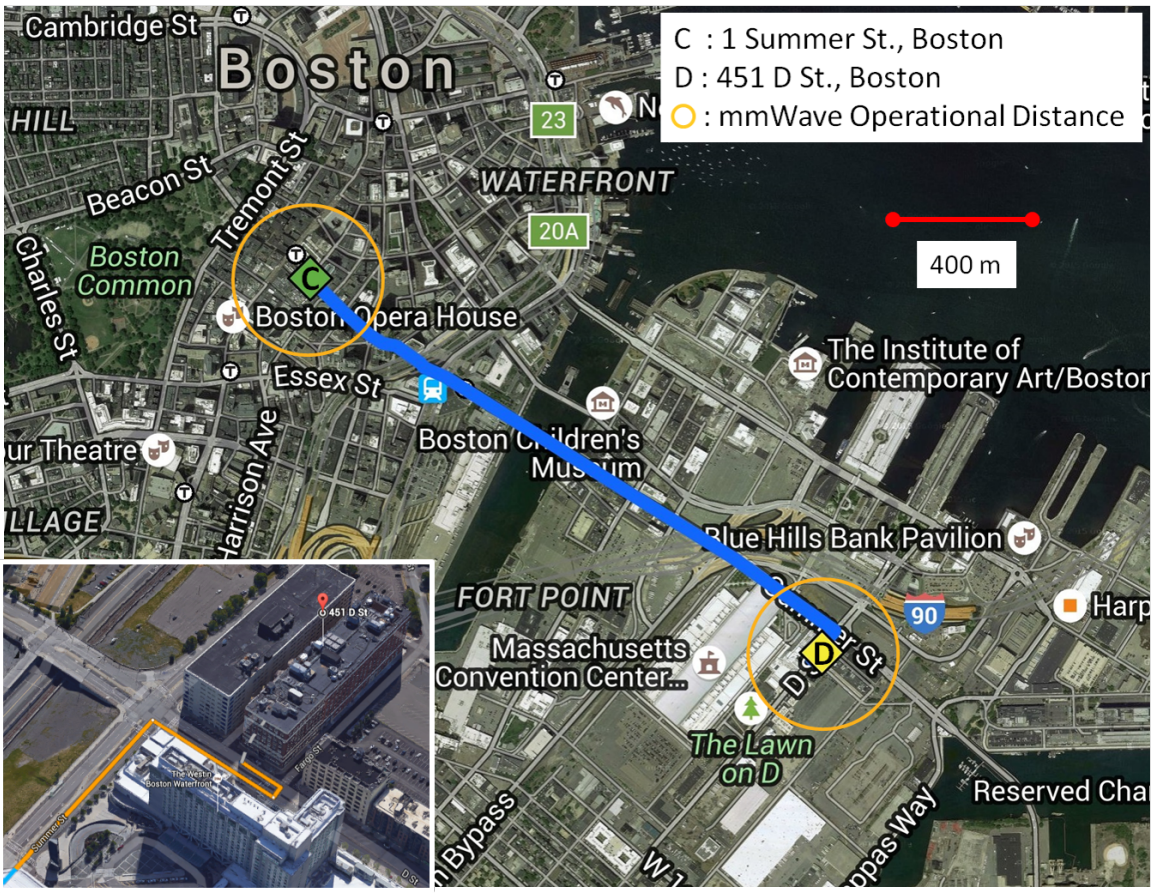


Figure 3.4: Google maps showing the suggested route for a vehicle moving from *1 Summer Street* to *451 D St.*. The end-to-end distance is roughly 1.2 miles and the estimated travel time is about 7 minutes, depending on the traffic conditions. The yellow circle represents the mmWave operational distance.

3.5 Data Exchange Evaluation

In this section, we evaluate the achievable capacity using an example scenario of V2I communication enabling data center traffic backhauling [10].

Specifically, we first introduce the adopted scenario in Section 3.5.1. Then,we assess the capacity as a function of the distance for both the mmWave and the THz links in Section 4.5.1. In Section 3.5.3, we derive the data shower bulk as a function of the minimum distance between the transmitter and receiver antennas, along with the effective data transfer rates for data centers located in Boston. Finally, in Section 3.5.4, we assess the benefits of adopting the proposed multiple-vehicle scheduling algorithm.

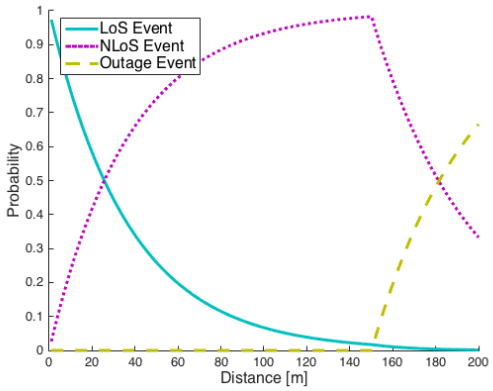


Figure 3.5: Empirical LoS, NLoS and outage probabilities for a mmWave link at 73GHz as a function of the separation distance between transmitter and receiver.

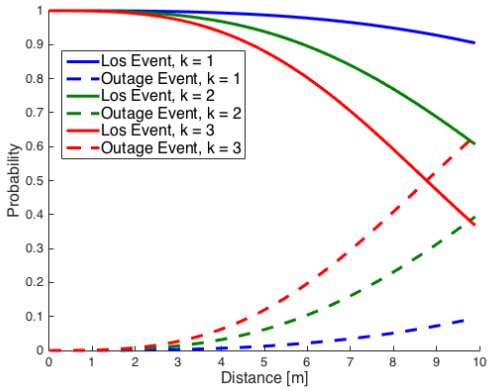


Figure 3.6: LoS and Outage probabilities for a THz link at 0.85THz with 0dBm transmitted power as a function of the separation distance between transmitter and receiver.

3.5.1 Network Scenario

To assess the achievable amount of exchanged data for backhauling under realistic conditions, we consider the actual positions of existing data centers located in Boston city [63].

Out of 22 available data centers, we choose two centers located in downtown Boston as typical use case: the first is located at *1 Summer Street*, owned by *XO Communications*, and the second is located at *451 D St.*, owned by *Markley Group LLC*. Through Google Maps, we obtain the suggested vehicular route between the two considered data centers, shown in Fig. 3.4. The vehicle route length is roughly 1.2 miles long with an estimated travel time ranging between 7 and 19 minutes. The inline picture shows the zoomed in view of the route near the first center. This is to indicate that the journey does account for the constraints arising from buildings and lanes.

From Google maps directions, basing on the antenna positions, we can estimate the distance between transmitter and receiver as function of time. We emulate a vehicle-assisted deployment where antennas are placed on vehicle rooftops and streetlight poles closest to the chosen data center, respectively. The rationale for this choice is twofold: i) the corresponding antenna heights agree with those used in mmWave channel measurements [61] allowing so us to adopt the corresponding experimental mmWave channel model; ii) the antenna positioning ensure that the THz link is not affected by outage events caused by pedestrians or vehicles blocking the LoS path.

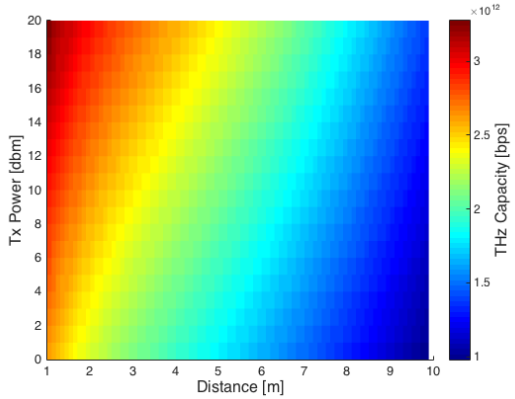


Figure 3.7: Capacity for a THz link at 0.85THz as a function of the transmitted power and the separation distance between transmitter and receiver. Outage events have been considered.

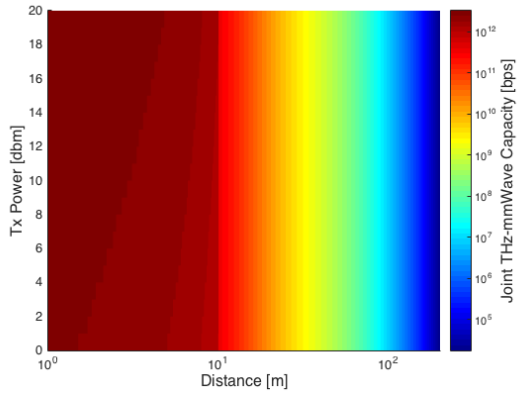


Figure 3.8: Capacity achievable by adopting the proposed THz/mmWave mode selection, as a function of the transmitted power in the THz band and the separation distance between transmitter and receiver.

3.5.2 Channel Modeling

The values for all the relevant parameters, used in this section, are summarized in Table 3.1. Their values are set according to previous works [5, 61] and [50, 52], as detailed below.

Regarding the mmWave communications, to provide a realistic estimation of the channel capacity, we use the experimental values of the mmWave channel parameters measured in [5, 61] for both the LoS and NLoS propagation conditions, when the carrier frequency is 73 GHz and the bandwidth is 1 GHz. As expected, the path loss for NLoS propagation conditions is significantly higher than the one in LOS propagation conditions. For the sake of clarity, in Fig. 3.5 we report the experimental values of the LoS, NLoS and outage probabilities given in equations (3.7), (3.8) and (3.9), respectively [5, 61]. According to the experiments, the probability of having an outage event is null for distances smaller than 150m, but it increases up to 0.7 for distances around 200m.

Regarding the THz communication, we adopt an accurate channel modeling by accounting for the molecular absorption characterizing USA high latitude locations at sea level during summer available in the Hitran Database [64] as done in the seminal work in [50]. Accordingly, the total path-loss $|H_{\text{THz}}(f, d)|^2$, given in (3.1), is a function of both the distance and the frequency. We account for some unique findings in relation the THz bands from the previous works [50, 52], i.e., the path loss in the THz band not only depends on the transmission distance and the system frequency, but also on the composition of the transmission medium at a molecular level through $k(f)$. Specifically,

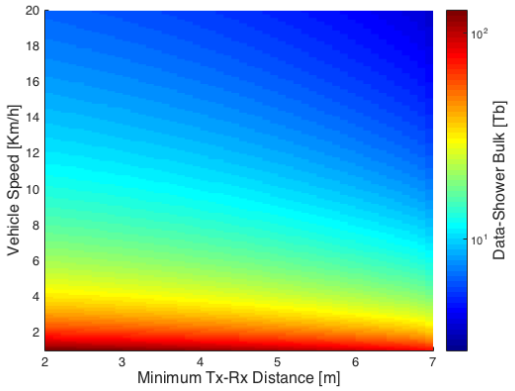


Figure 3.9: Data Shower Bulk as a function of the minimum separation distance d_{min} between the transmitter and the receiver and the average mule velocity. Single-way journey between the vehicle, moving with constant-speed along a straight-trajectory, and the tower.

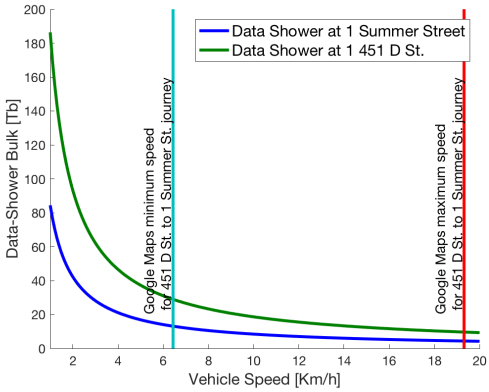


Figure 3.10: Data Shower Bulk as a function of the average mule velocity. Single-way journey between two towers located at *451 D St.* and *1 Summer Street* and owned by *Markley Group LLC* and *XO Communications*, respectively, through the route suggested by Google Maps.

we observe that: i) the path loss increases with both the distance and the frequency; ii) several peaks of attenuation can be observed due to the molecular absorption loss controlled by $k(f)$; iii) the molecular absorption defines several transmission windows along the frequency scale with varying widths that are defined by the molecular composition of the medium. It is clear that the spectrum range [0.8-0.9] THz represents a suitable band for THz communications up to 10 meters, and the results derived in the following assume the use of such a band. Clearly, larger bands can be exploited by adopting distance-based modulation techniques and the results derived in the following continue to hold.

In Fig. 3.6, we report the values of the LoS and outage probabilities for the THz communications obtained according to the model (3.10). The simulation setting is as follows: the transmitted power is 0dBm and the minimum SNR $\gamma_{\text{th-THz}}$ required for establishing the THz link is given by:

$$\gamma_{\text{th-THz}} = k\gamma_{\text{THz}}(10) \tag{3.26}$$

with k ranging from 0.1 to 1, i.e., with $\gamma_{\text{th-THz}}$ being a fraction of the average SNR measured at a distance equal to 10m.

The rationale for this model is twofold: i) it allows us to abstract from the particulars of the THz transceiver, such as its sensitivity or noise figure; ii) it sounds reasonable to assume that the

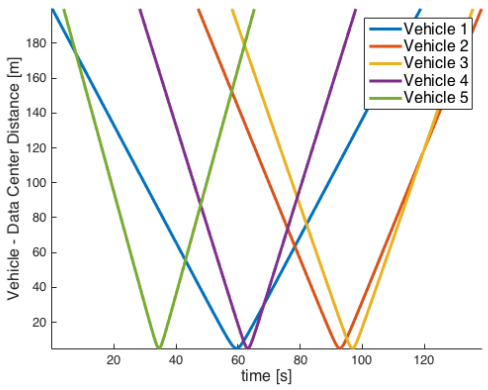


Figure 3.11: Vehicle to data center distance as function of time for a single Monte Carlo realization. Minimum separation distance during closest approach is roughly 5 m. Used as input to generate Figure 3.12.

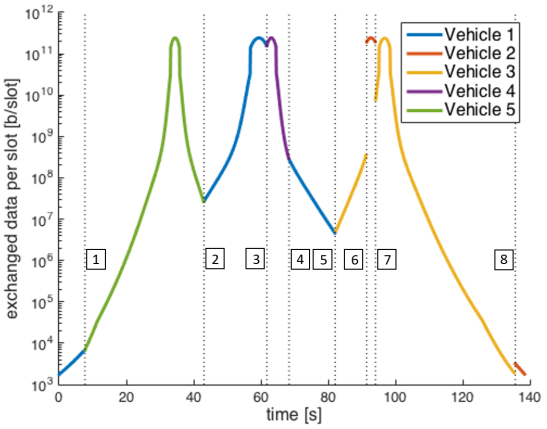


Figure 3.12: Amount of exchanged data in every time slot by adopting the proposed greedy scheduling algorithm (Algorithm 1). Each switch is identified by the dotted vertical line with an associated index.

minimum SNR required for establishing a THz link is related to the SNR measured at the maximum distance at which the THz link could be established. We observe that for a transmitted power of 0dBm, at 10m we measure an outage probability of roughly 0.6 for $k = 1$.

3.5.3 Data Shower Performance Analysis

In Fig. 3.7, we report the THz capacity as function of the transmitted power and the separation distance between the transmitter and the receiver, by accounting for the outage loss as in (3.10). Although we under-estimate the achievable THz capacity given in (3.2) by limiting our attention to a single spectral window, we note that the achievable capacity is greater than 1 Tbps for every values of the considered transmitted power at the maximum distance of 10m. Furthermore, in presence of a LoS connection the achievable capacity at 10m roughly increases of 1.5 times.

In Fig. 3.8, we report the distance-dependent capacity available by adopting the proposed protocol, derived in (3.3), as a function of the distance and the THz transmitted power. Within the considered distance range $[1, 200]$m, the achievable capacity varies of several orders of magnitude, ranging from Tbps to Mbps for distances around 200m. This result is reasonable, since: i) in urban scenarios, the probability of a mmWave LoS connection decreases significantly as the distance increases, due to the building outage effects, and ii) mmWave NLoS path loss is particularly severe, with values exceeding 200dB for distances greater than 100m. Nevertheless, we note that the

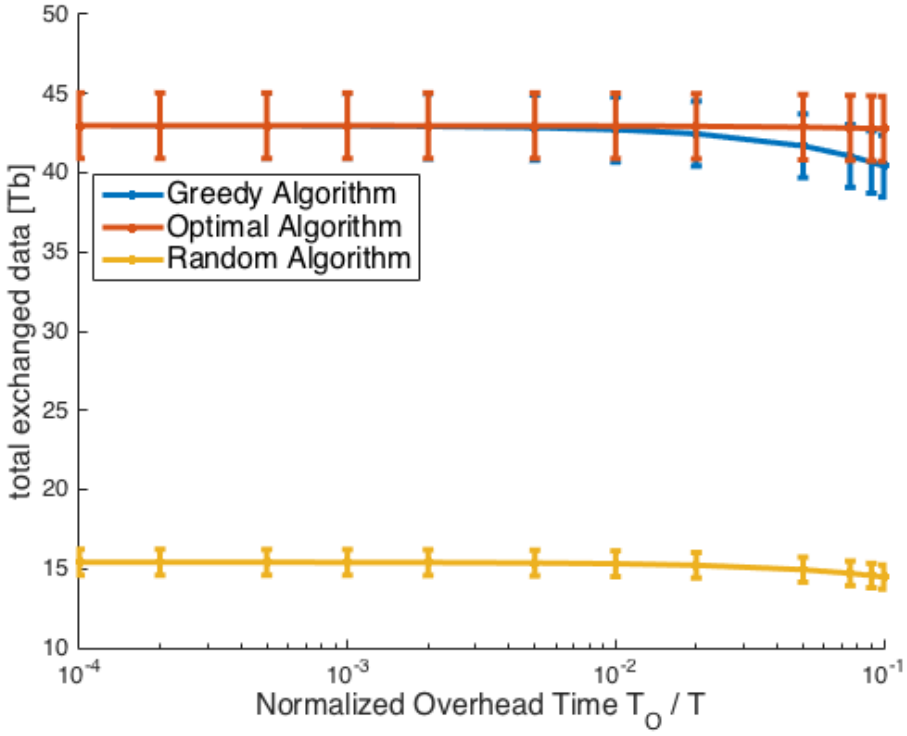


Figure 3.13: Comparing the total exchanged data transferred with the three scheduling approaches.

available capacity exceeds the Gbps and the Tbps for distances in the order of 10 meters or less, respectively, for every value of the transmitted power.

In Figure 3.9, we show the data capacity derived in (3.13) as a function of both the minimum distance $d_{\min}$ between the transmitter and the receiver, and the average mule velocity. Specifically, we show that the amount of bits that can be transferred in a single-way journey between the vehicle, moving with constant-speed along a straight-trajectory, and the data center by adopting the proposed THz/mmWave mode selection. For a fair comparison, we assume that the transmitted powers of the mmWave and the THz links differs by at least 10dBm, i.e., we assume a Tx power of 30dBm and 20dBm for the mmWave and THz links, respectively. We adopt the same transmitter power value for mmWave communications used in the real-world experiments described in [61]. For THz communications, we consider levels up to 20dBm to account for the latest results experimentally achieved in submillimeter literature [65, 66]. We note that the data rate increases as the average velocity decreases, having the mule spending more time in the range in which a mmWave/THz communication is possible. Hence, by controlling the velocity of the mule, an impressive transfer of information can be easily achieved. In particular, we observe that at the reasonable minimum

distance of 4m, we are able to transfer an amount of information exceeding one Terabit with a single journey in the worst case, i.e., when the average mule velocity is 10Km/h. Even more impressive, when the average velocity is roughly 2 Km/h, the amount of information exceeds 100 Terabit with a single journey for every considered minimum distance. These results suggest that by using the proposed mmWave/THz switching protocol, we can exchange much higher amount of data compared to what can be achieved with classical wired or wireless technologies.

In Fig. 3.10, we quantify the data transfer volume as function of the mule velocity for the real journey traced in Fig. 3.4, with the mule reaching the (existing) tower located *451 D St.* starting from the (existing) tower located at *1 Summer Street*. The distance between the mule and data center as a function of time has been obtained from the journey route suggested by Google Maps. Specifically, the minimum distances between the transmitter and the receiver are 5.02 and 5.03m, respectively. The minimum and the maximum average speed, obtained through the Google Map estimation of the journey time are reported within the figure. We observe that the results shown in Fig. 3.10 confirms that the data exchange of around 100 Terabit is possible with a single journey for each data center.

3.5.4 Scheduling Performance Analysis

To assess the performance of the proposed scheduling procedure, we consider multiple vehicles traveling along a two-lane road with constant velocity by first approaching and subsequently moving away from the Infrastructure.

The closest distance of approach to the data center is around 5 m as in Figure 3.10, and the vehicles enter within the mmWave operational range at random times. The vehicle velocities are picked uniformly at random within the range [3-7]m/s, modeling so a typical urban scenario. Hence, the vehicles are characterized by different contact times.

In the first experiment, we consider 5 vehicles to be scheduled. Fig. 3.11 shows the distances between the vehicles and the data center as a function of time for a single Monte Carlo run. The time horizon is roughly two minutes, corresponding to 1387 slots. Given the variable arrival and contact times, the naive approach of scheduling vehicles on first-come-first-serve basis (even with all the other settings held identical) will clearly be sub-optimal.

This is confirmed by Fig. 3.12, which shows the amount of data exchanged by each vehicle in each time slot by adopting the greedy scheduling algorithm given in Algorithm 1. Fig. 3.12 is obtained by considering the same Monte Carlo realization depicted in Fig. 3.11 and with D_v

uniformly distributed in $[5, 15]$ Tb. A switch between scheduled vehicles happens at the time instants depicted with the dotted vertical lines. A total of 8 switches is observed in the entire time horizon. The contiguous set of slots assigned to a vehicle can be identified by the color. We note that three vehicles, i.e., vehicle 1, 2 and 3, are served in non-contiguous time slots so that the overall throughput can be maximized. Clearly, as pointed out in Sec. 3.4, the greedy algorithm does not assure always the optimal solution, since it does not account for the time overhead in vehicle scheduling. Finally, we note that the peaks in the figure are indicative of the time slots during which the vehicles are in the THz operational range.

To substantiate the performance of the proposed greedy algorithm, in Fig. 3.13 we compare it with the random scheduling and the optimal scheduling designed in Section 3.4. Specifically, we implement the optimal scheduling through the exhaustive search of the solution maximizing the total number N of exchanged bits, among the admissible solutions satisfying constraints (3.18) and (3.19). For a fair comparison, we implement the random scheduling by selecting uniformly at random one solution among the admissible ones. To assure practical time and memory complexity for the optimal and the random algorithms, we limit the number of vehicles to two.

Fig. 3.13 shows the average total exchanged data as a function of the normalized overhead time, i.e., T_O/T, for 1000 Monte Carlo runs. The vertical bars denote the 95% confidence intervals. The relative performances of three algorithms in comparison is quite apparent. Importantly, the greedy algorithm exhibits excellent performance overall and an optimal performance for overhead time lower than $10^{-2}T$. Intuitively, this can reasoned in the following way: Greedily assigning the set of vehicles that can complete its data transfer needs at the earliest, and subsequently removing those vehicles that have successfully completed the data exchange from any further assignment is a clever strategy for it rapidly makes progress in reducing both the overall backlogged data from all vehicles, and the overhead time that will be needed in the future within the time horizon. Since the greedy approach prioritizes the completion of data needs of each vehicle over the overhead cost, the performance deteriorates slightly from that of the exhaustive search for large overhead time relative to the slot time.

Not surprisingly, the random algorithm performs poorly as it disregards the variable amount of backlogged data on each vehicle. Moreover, random assignment on every slot implies a poor utilization of high bandwidth as it essentially keeps switching across vehicles and accumulates all the coordination overhead time. This impairment in performance becomes all the more apparent and severe when the normalized overhead time approaches unity – the overhead time occupies the entire slot time thereby leaving no time for data exchange.

Chapter 4

Resource Allocation Scheme for Multi-User mmWave V2I Network

4.1 Background

4.1.1 Issues Specific to mmWave V2I Communication

Millimeter wave communications differs from microwave in a number of ways: (i) different channel models, (ii) new hardware constraints due to the high operating frequency and bandwidths, and (iii) large arrays employed at both the transmitter and receiver.

• There is a lack of accurate channel models, as well as extensive measurement studies that characterize the vehicular scenarios [16].

• Outdoor millimeter wave communications suffer from high propagation and penetration losses. For this reason, frequency bands centered at 28 GHz, 38 GHz, 73 GHz, and in the 81–86 GHz range [67] have gained considerable interest as the losses are relatively less.

• In mmWave urban environments, penetration through just one wall incur losses $\sim$30 dB, particularly for buildings structures with steel concrete and energy saving windows. This implies propagation that involves penetrating through the urban buildings is not a relevant effect [68]

• Gaseous absorption, in particular, is insignificant for the urban cellular deployments, where base station spacing's might be on the order of 200 m [69].

• The use of directional antennas makes the mmWave links quite sensitive to occlusions/blockages resulting from concrete buildings, foliage, pedestrians.

• The availability of strong LOS/NLOS links is shown to be highly location and orientation specific

implying the strongly site specific nature of mm-wave link [70]. This suggests paying close attention to the geometry and building materials of the specific urban environment at a potential mm-wave BS location.

• To overcome the severe shadowing and path loss at millimeter-wave, amplifiers need to generate enough power by operating in the saturation region. As a result, the received signal is likely to be severely affected by nonlinearities [67].

• mmWave communications can be impacted significantly by CFO (due to clock frequency mismatch) and phase noise (due to imperfections in the local oscillators) as oscillators at millimeter-wave frequencies are not as accurate compared to those at microwave frequencies [71]. This can become important when dealing with the Doppler spread arising due to the user mobility. Doppler spread may introduce different frequency shifts [72]. Carrier synchronization algorithms that are usually designed to track only one frequency offset will find it difficult to eliminate the frequency spread caused by Doppler.

• In mmWave, given the need for quick reconfiguration and the reduced coherence due to high mobility, the beamswitching technique may be preferred over beamsteering [73, 74]. In this approach the beam patterns that can cover the entire region are pre-configured and the final tuning concerns only the selection of one of these patterns.

4.1.2 Hybrid beamforming

Analog beamforming is not suitable for multi-user mmWave systems for it does not scale with the number of users. On the other hand, digital beamforming requires as many radio-frequency (RF) chains as the antennas used. This is prohibitive due to the high cost and power consumption at mmWave. Hybrid beamforming reduces the complexity of digital beamforming while improving the performance of analog beamforming [75]. Hybrid analog-digital beamforming is increasingly preferred in multi-user systems as it considers both digital and analog precoder/combiners and allows for more designing freedom than analog beamforming. As a result, few radio-frequency (RF) chains can drive a large number of antennas making for a feasible architecture in mm-Wave as it best trades off complexity (energy losses) and flexibility. The multi-user association phase will benefit from a fully hybrid precoding/combining scheme wherein both BS and MSs may form multiple beams.

Unlike analog only precoding/combining that allows only one beam per time, hybrid precoding allows for multiple beams thanks to spatial multiplexing which is useful in multi-user scenario [76]. Considering the case that the BS allows multiple streams and communicates with each

MS via only one stream, we have the number of streams $N_S = U$. It is reasonable to assume that the number of beams is equal to the number of users since spatial multiplexing gain of a multi-user hybrid precoding system, with $U \leq N_{RF}$, is limited by $min(N_{RF}, U)$, where N_{RF} is the number of RF chains and U is the number of users [77].

4.2 Related Work

In the V2I context, progress has been made in tackling technical challenges relating to accurate channel models, reduction in beam alignment overhead, Doppler compensation mechanisms.

4.2.1 User Association Phase

The aim of the user association phase is to robustly align beams and quickly attain connectivity, a procedure to be successfully completed prior to communication. Unlike microwave, initial access in mmWave is very challenging due to the need for beam alignment [78]. Significant delay can be incurred by wastefully testing useless beam combinations at both ends. This phase necessitates the BS periodically transmit synchronization signals while the users scan for the presence of these signals to detect the base station, and learn the timing and direction of arrivals.

[79] proposes a directional cell discovery procedure where the mm-Wave BS uses omnidirectional and random directional transmissions, and the MS performs either analog, digital or hybrid beamforming. The use of omnidirectional transmission of synchronization signals at the BS during user association will reduce coverage, while random directional transmission will result in large delays. Moreover, the procedure employs either random beamformers or no beamforming for the transmission of synchronization signals. Clearly, more sophisticated beamforming or even deterministic search patterns are likely to prove superior.

We account for the above shortcomings in the design of our user association phase in our MAC protocol.

4.2.2 Directional MAC Protocols

Carrier sense based protocols like the CSMA/CA is ill-suited for contention resolution in mm-Wave networks as Clear Channel Assessment (CCA) will be impaired due to the high directionality of the beams. [80] provides a modified CSMA/CA protocol, aimed at current mm-Wave standards, that addresses the inefficiency of prolonged back-off time (CSMA/CA) by the

inclusion of a collision notification signal. However, they make simplistic impractical assumptions towards resolving multiple RTS that collided at the BS.

We identify this as one of the critical MAC layer design aspects.

4.2.3 Coherence Time and Coherence Bandwidth

4.2.3.1 Coherence Time (T_C)

High Doppler, arising from mobility of the MS, results in a reduced channel coherence time, introducing time-selective fading. The Doppler spread is directly proportional to the carrier frequency. This means that for the mmWave radios, that operate at an order of magnitude higher frequency compared to microwave bands, it will be 10 to 30 times higher. The classical result is that the coherence time is inversely proportional to the maximum Doppler frequency, $T_C \sim 1/f_D$ (where $f_D = \frac{vf}{c}$). This however is not accurate for mm-Wave systems employing directional antennas. The authors in [81] show that the directional reception leads to smaller Doppler spread resulting in a larger coherence time and provide a mathematical relation between coherence time, in LOS conditions, and small beam-width θ of the MS:

$$T_C(\theta) = \frac{D_\lambda}{f_D \sin(\alpha_{LOS})} \cos^{-1}(2\theta^2 \log R + 1) \text{ where } D_\lambda = D/\lambda \tag{4.1}$$

The description and typical values of the parameters involved is given below in the table.

Parameters	Description	Typical Value
θ	Vehicle antennas' beamwidth	5°
v	Max. receiver speed	30 m/s
f_c	Carrier frequency	60 GHz
λ	Carrier wavelength	5 mm
f_D	Max. Doppler	v/λ
D	Tx-Rx distance	50 m
α_{LOS}	Angle of arrival of the LOS path	$0^\circ - 90^\circ$
μ_r	Pointing angle	$0^\circ - 90^\circ$
R	Target correlation	0.5

They further show that the chosen range of θ must feasible in practice and makes for beams that aren't too narrow or too wide, so that both the pointing error and the Doppler do not limit performance.

Moreover, [81] describes a beam coherence time, T_B, with regards to the constant need for beam realignment owing to the mobility of the vehicle. The authors define T_B as the duration after which the signal power drops by half due to beam misalignment. In predominantly LOS conditions, it is shown that for realistic parameter settings, $T_B \gg T_C$ (by an order of magnitude),

implying that beam realignment needn't be performed in the interval, T_C, during which the vehicle is communicating.

4.2.3.2 Coherence Bandwidth (B_C)

The authors in [82] provide a best description model,

$$B_C = \alpha \cdot e^{-\beta \cdot RMS-Delay-Spread} \tag{4.2}$$

for the relationship between coherence bandwidth and root-mean-square (rms) delay spread in an indoor setting as there is no clear relationship between those two parameters. Moreover, they inform us of the random behavior of delay spread with receiver position for channels dominated by a strong direct path. The measurement campaign [83] carried out at 62.4 GHz in suburban street setting with wide-beam-width antennas show that the coherence bandwidth is highly variable with the location of the MS relative to the BS. Therefore, measuring the correlation coefficient between the signal envelopes over several frequency spacings are insufficient to characterize the frequency correlation function of the channel. Coherence bandwidth (B_C) is shown in [84] to depend on the street layout, frequency, antenna heights, and street width. In LOS propagation, however, the multipath components are sparse and, in contrast to 28 GHz, measurements at 73 GHz found fewer, but stronger multipath components (Refer to Table V in [85]). Delay spread [86] is generally much lower due to the usage of directional antennas. Measurement campaign [87] carried out in urban low-rise environments in LOS environments, determined that using beamwidth of 10° at 28 GHz, RMS delay spreads (over all TX-RX location combinations) are less than 25 ns (with $\mu \sim 8$ ns). The experimental study [88] carried out at 55 GHz with narrow beamwidth antennas in an urban mobile setting estimates the coherence bandwidth to be greater than 66 MHz in a large number of cases and the smallest value of coherence bandwidth to be about 20 MHz. Also, the authors point out the usefulness of estimating the delay spread from the measured coherence bandwidth,

$$B_C = \frac{1}{2\pi \cdot Delay - Spread} \tag{4.3}$$

The above estimate despite being insensitive to the shape of the delay distribution (considering a simple geometrical model of the channel) is very in much in agreement with the value of B_C produced from a smooth exponential distribution. From the observations in [86], [87], and [88] discussed here, we can specify the range of coherence bandwidths:

$$20MHz \leq B_C \leq 132MHz$$

We can model the B_C as random samples drawn from a probability distribution P_{B_C} with the following property:

$$P_{B_C}(20MHz \leq B_C \leq 132MHz) = 1 \tag{4.4}$$

$$P_{B_C}(B_C \geq 66MHz) = 0.95 \tag{4.5}$$

4.3 Proposed Design Approach

We consider a fully hybrid precoding/combining scheme for the discovery phase. In our approach, the BS initiates the association with MSs by sending a synchronization message on multiple beams. This helps speed up the the association with the vehicles, while the process gain of the beam coding compensates for the reduced received mmWave signal power. With its concurrent beams, the BS then receives the multiple collision-free RTS requests. The association step is complete once the BS discovers all the contending MSs by exhaustively scanning the possible beam sectors.

A time-frequency resource allocation scheme carried out post completion of the user association phase, is developed. The protocol is utilized by the BS for efficient multiuser uplink/-downlink scheduling. The protocol is specifically designed considering the bandwidth needs and time the vehicle will continue to be in LOS. Every MS is assigned a portion of the time and an independent set of consecutive frequency subcarriers, thereby allowing for heavy multiplexing of MSs. The fundamental problems we tackle here are: (i) how to decide the bounds for each such rectangular resource block, T_k and B_k, and (ii) how to pack these resource blocks tightly within the broader systems-defined constraints of T_{tot} (time) and B_{tot} (frequency). The problem is formulated as a rectangle bin packing problem, where the optimized packing is determined by considering the objective that minimizes the unused areas, i.e. the resources not allocated to the users and therefore wasted.

4.4 Multi-user Directional Medium Access Protocol

The BS must be able to serve, within its operating range, multiple MSs which are likely to be in random road locations. Let the hybrid beamforming BS can serve a few beam directions, say N, concurrently. So, in order to keep the intial beam steering complexity to a minimum, its best the BS adopts beam switching towards user association. Let M beam directions (we will refer to the same as simply beams henceforth) be required at the BS to cover all the possible M beam sectors where the

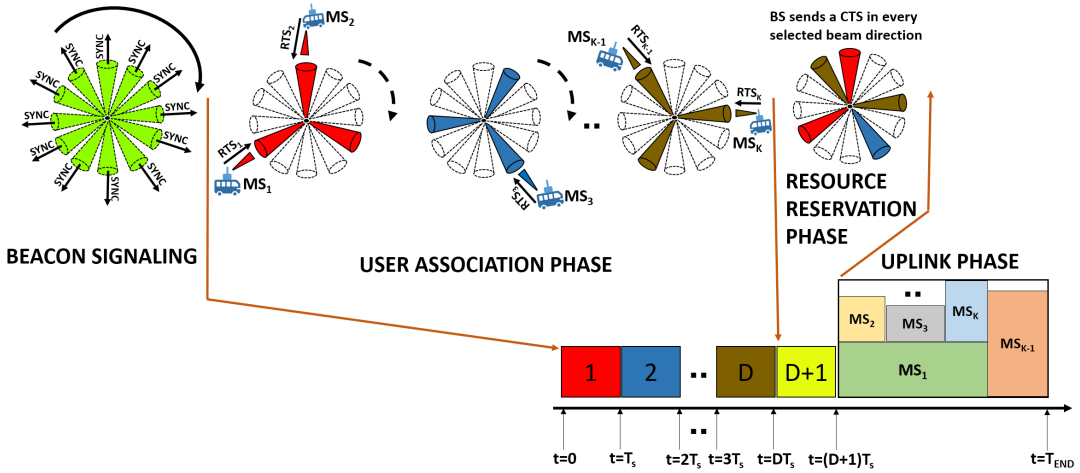


Figure 4.1: Directional MAC Protocol operating in three phases. Lock step switching of random start, fixed orientation beam patterns at BS.

MSs will be present. We assume that no more than N beams can simultaneously be active among the entire set of M beams.

Beam switching with the N beams can happen in one of the following ways:

1. random orientation of beam directions and switching of beams occurring in lock steps.
2. fixed orientation of the beams with random start and switching of beams occurring in lock steps.
3. fixed orientation of the beams with a random switch of beams.

Using concurrent beams to establish connection between BS and MSs may require mitigating inter-beam interference. So, we select beams that are maximally spaced apart. Given the likelihood of selecting very close beams in option 1) and an increased beam switching complexity in 3), we prefer the option 2).

The BS in the association phase picks m_{UA} (very few) beam directions that are maximally spaced out with beamwidth, θ_{UA}, that may relatively be wider than the beamwidth, θ_{RA}, used in the resource allocation. This allows the BS to scan angular sectors quickly and thereby keep the user association overhead to a minimum.

Let us say that the BS serves no more than N users in the uplink phase and D be the maximum

number of switches required for the BS to scan the beam sectors. We have,

$$m_{UA} \leq N \ll M \tag{4.6}$$

$$\theta_{UA} \geq \theta_{RA} \tag{4.7}$$

$$D = \lceil M/m_{UA} \rceil \tag{4.8}$$

The lock step switching time is represented as T_S (in seconds).

As shown in Fig. 4.1, the proposed protocol operates in four distinct phases. Next, we will describe the protocol operation:

Beacon Signaling Phase:

1. BS transmits a synchronization signal S in all the beam directions one after the other, say, in a clockwise manner.
2. MS(s) receive and decode S to synchronize with the BS. Specifically, they perform channel estimation and estimate the necessary beamforming direction.

User Association Phase:

1. At t = 0, the MS(s) which successfully completed the previous step, transmit a RTS to the BS in the same beam sector.
2. The BS receives and decodes the RTS. When successful, the BS and the corresponding MS are now associated.
3. The BS stops switching the fixed orientation beam pattern only when the associated vehicles reach N; even if all the D beam directions are exhausted.

Resource Reservation Phase:

1. When the number of associated MSs reaches N, the BS sends out the CTS along the selected beam directions.
2. The corresponding MS decode the CTS to become known of the exact time-frequency resource it is scheduled for access.

Uplink Phase:

1. Every associated MS_k establishes uplink with the BS in its allocated RSB_k.

2. At t = t_{END}, uplink is complete.

3. BS restarts the Beacon Signaling Phase.

4.5 Resource Allocation for Multi-User mmWave Vehicular Communications

Given the stringent beam-steering requirements and site-specific nature of connectivity in mm-Wave, the requests of time-frequency resource from the MSs will be asymmetric.
In this section, the time-frequency resource allocation at the BS for the mobile outdoor mm-Wave small-cell communication is discussed. The resource allocation is then mathematically formulated as a two dimensional rectangular bin packing problem. Determining the optimized packing to the stated problem involves three key steps:

1. Picking for every instance of the problem, the boundaries of the rectangular bin and the smaller rectangular pieces using the relevant models. The width and height of the rectangles together are referred to as the boundaries.

2. Identifying the set of packing constraints with respect to orientation, unused areas etc.

3. Using a computationally-feasible algorithm to determine the optimized packing satisfying the above constraints.

We will discuss these steps in detail in the subsequent sections.

4.5.1 Radio Frame Design

Let $T_{C,k} := T_{C,k}(v, \theta)$ represent the coherence time experienced by the mobile MS_k moving at speed v and steering a beam of width θ and $B_{C,k}$ represent the coherence bandwidth experienced by the MS_k.
We will refer to the smallest discrete unit of time-frequency resource that the BS can handle as the Minimum Resource Unit (MRU). The MRU can be represented as a tuple (T, Δ f) and visualized as one small blue box with black boundaries in Fig. 4.2 & Fig. 4.3. Let T_C^{min} and B_C^{min} be the minimum possible coherence time and coherence bandwidth. We can compute the Pilot Spacing (PS)

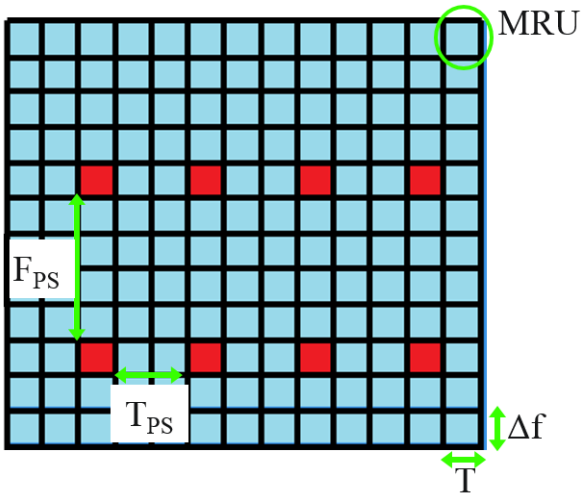


Figure 4.2: Radio Frame Showing the Important PHY Parameters

and the MRU in the radio frame using the following equations:

$$f_D^{max} \ll \Delta f \ll \frac{1}{T_{CP}};\ T_{CP} \geq T_{Delay-Spread} \tag{4.9}$$

$$T = T_{CP} + T_{DATA};\ T_{DATA} = N_{CP} \cdot T_{CP} \tag{4.10}$$

$$T_{PS} \leq \frac{1}{K_T} \cdot T_C^{min};\ T_C^{min} = \min_k\ T_{C,k}(v,\theta) \tag{4.11}$$

$$F_{PS} \leq \frac{1}{K_F} \cdot B_C^{min};\ B_C^{min} = \min_k\ B_{C,k} \tag{4.12}$$

The PHY parameters relevant to the radio frame tabulated below can be obtained by setting f_D^{max}, T_{CP}, N_{CP}, T_C^{min}, B_C^{min}, K_T, K_F at 5 KHz, 10 ns, 15, 25 ms, 20 MHz, 2, 5 respectively, and using the above inequalities.

Parameters	Description	Typical Value
T	OFDM Symbol Time	0.16 μs
Δf	Sub-carrier Spacing	25 KHz
T_{PS}	Pilot Spacing (in time)	10 ms
F_{PS}	Pilot Spacing (in frequency)	3 MHz

4.5.2 Resource Block (RB) Allocation

• **Notation:** Let there be N MSs in the coverage area of the BS and only K MSs ($K \ll N$) among them request uplink access from the BS at time t. Let the MSs be represented by the set $\mathcal{N} := (1, 2, .., i.., N)$ and the K MSs by the set $\mathcal{K} \subset \mathcal{N}$.

Let T_k and B_k represent the requested time and bandwidth for access by the vehicle MS_k.

At the BS, let the total time and total bandwidth available at the start of resource allocation phase be T_{tot} and B_{tot} respectively. We set T_{tot} and B_{tot} similar to that of IEEE 802.11 ad standard.

$$T_{tot} = [100 - 1000]\ ms \tag{4.13}$$

$$B_{tot} = 2\ GHz \tag{4.14}$$

We denote the Resource Block (RB), available at the BS at time t, as an ordered pair $RB^t := (T_{tot}, B_{tot})$. At time t, MS$_k$; $k \in \mathcal{K}$, request for a Resource Sub-Block (RSB) and the BS schedules all/some of the K MSs for uplink access. The BS having to accommodate asymmetric RSB requirements, might not be able to schedule all MSs requesting access. Simply put, if MS_k's request gets accepted, the BS assigns MS_k an RSB spanning a bandwidth B_k and time-slot T_k i.e. $RSB_k^t := (T_k, B_k)$, else if the request is rejected by BS, MS_k contends to gain uplink access at a later time i.e. say at the start of the next RB, at time $t + T_{tot}$. The scheduling in the Time-Frequency

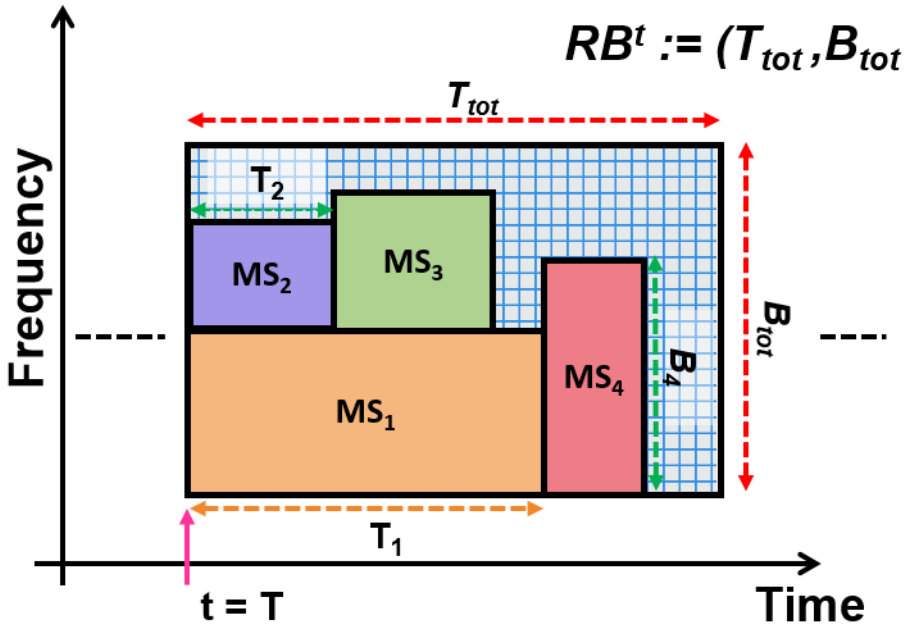


Figure 4.3: Resource Block Allocation in the Time-Frequency Grid

Grid (TFG) can be thought of packing many smaller rectangles, RSB_k^t, in a bigger rectangle of fixed size, RB^t; visually represented in Fig. 4.3. Note that since both RSB_k^t and RB^t are made up of several MRUs, the edges of all the rectangles can be deemed to take on only positive integer values.

• **Packing Criteria:** To further qualify the objective in the scheduling step, we identify a pertinent set of requirements that the allocated RSBs need to satisfy:

1. Since the MS is expected to request for a RSB with T_i that is highly location dependent, all pending requests must expire after T_{tot} seconds. The MS has to contend for access again in the next RB. As a result, the scheduling cannot span multiple RBs thereby calling for an one-shot allocation of the RSBs. This means that the packing **does not span multiple bins**.

2. As each beam of the BS services only one MS in the allocated RSB, the RSBs are required to be **non-overlapping** in the RB.

3. Since the MS observes a channel that varies faster relative to the BS particularly due to required support for mobility, we emphasize that the RSB also has to be an ordered pair. This amounts to **fixed orientation and allowing no rotations** among the RSBs.

4. Since the mm-Wave channel access is expensive, the BS will allocate RSBs for the requests as and when they arrive. This means that the RSBs are **placed in the received order**. It is not removed from/repositioned within the RB post allocation.

5. For the system to be spectrally efficient, the BS will preferentially allocate adjoining RSBs that are similar to keep the spectrum switching overhead to a minimum. This calls for **guillotine packing** of the RSBs.

• **Packing Objective:** The BS has the following specific objective towards uplink access scheduling: Maximize the time-frequency resource utilization at the BS. The BS tries to minimize the amount of unallocated regions in the RB as shown in the Fig. 4.3. Put equivalently, the objective is to minimize the unused areas/whitespaces (trim loss) in the bigger rectangle. The above stated objective is identical to the one described in [89] as the online rectangular bin packing problem. The ordered tuple (T_{tot}, B_{tot}) represents the rectangular bin. The author provides an algorithm, with time complexity $\mathcal{O}\left(n^2\right)$ and space complexity $\mathcal{O}\left(n\right)$, that helps maximize the utilization, $\mathcal{U}$, of the time-frequency resource:

$$\mathcal{U} = \frac{\sum_{k \in \mathcal{K}} I_k \cdot T_k \cdot B_k}{T_{tot} \cdot B_{tot}} \tag{4.15}$$

where $I_k \in \{0, 1\},\ k \in \mathcal{K}$. Once an optimized packing is identified, the BS then schedules each MS_k by sending out a CTS_k on the corresponding beam. This opportunistic and dynamically adjusting design will significantly improve the performance.

4.6 Simulation Environment

Commercial 3D ray-tracing tools are now popular and are increasingly being used to accurately model channels for static indoor mmWave scenarios. Currently, such tools do not support vehicular simulations, particularly for link layer operations that involve directional, steerable beam patterns and therefore might prove unsuitable for our simulations. Moreover, such tools are quite expensive with yearly licenses costing in several thousands of dollars and also demand a steep learning curve before one can obtain useful results. For this reason, we specifically designed and built in MATLAB a simulator to address link layer operation characteristics of mmWave V2I network in an LOS urban setting with low-rise modern buildings.

The simulator window shown in Fig. 4.4 represents the 2D layout of a section of a city. The vehicular routes, the data needs for each vehicle is the input to the simulator. For all locations of the MS in its route, the azimuth and elevation angle pair which both the BS and the MS must point their beams at is determined and also displayed to help the user visualize the moving-vehicle scenario. We adopt the 3GPP microcellular model for the LOS probability specified as [90],

$$P_{LOS}^{3GPP}(x) = min\left(1, \frac{18}{x}\right)\left(1 - e^{-\frac{x}{36}}\right) + e^{-\frac{x}{36}} \tag{4.16}$$

$$P_{Outage}(x) = 1 - P_{LOS}^{3GPP}(x) \tag{4.17}$$

The vehicular routes are assumed to be usually along the long streets and at times run between the narrow spacing among the buildings. The environment allows for modeling for the variable speed of the vehicles, the distinct link state (LOS & Outage), directivity gain of narrow beam antennas, tracking of beam directions, quite relevant to mmWave V2I networks.

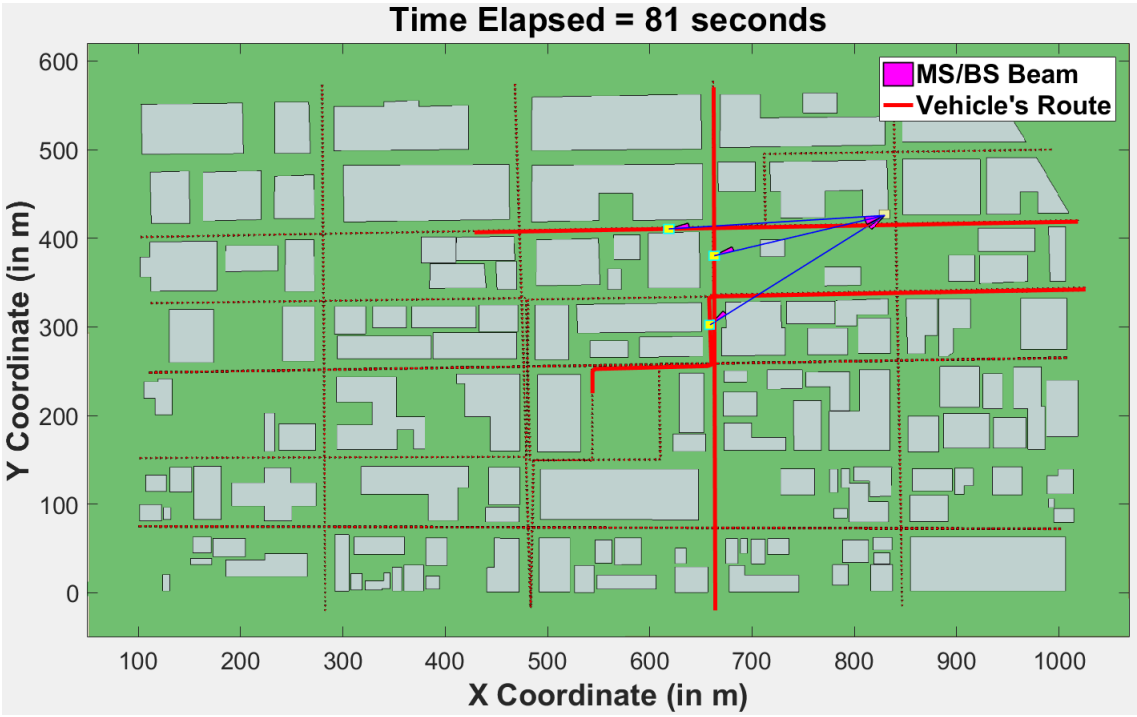


Figure 4.4: mmWave V2I Simulator: BS Serves Multiple Associated Vehicles

Chapter 5

Conclusion

In Chapter. 2, we describe a system built around the concept of state-action based design and slot-time synchronized operations that helps combine and realize the PHY and MAC layer that is IEEE 802.11b standard compliant. In addition, the system allows the user reconfigure the parameter values as needed. Using the MATLAB Coder to automatically generate MEX functions is beneficial in improving the speed consistency of our system blocks, which can vary its frequency resolution parameter. This work provides a testbed to experiment with new MAC protocols beyond that specified in the IEEE 802.11b standard. The state machine design enables modularity of code base and should allow for extensibility by the community. The three node system remains fair to the two bi-directional links for varying payload sizes in the DATA packet. Through our experiments we establish the role and efficacy of the implemented MAC layer towards mitigating packet collisions and enforcing fairness among DTxs in accessing a common channel.

There were a number of difficulties during the implementation that we had to overcome. Foremost, we had trouble realizing slot-synchronized operations, one of the most crucial issues in real-time testbeds. Second, it was difficult to pick the right energy threshold to deal with a variable noise floor due to environmental noise effects. Finally, our system required a thorough calibration step prior to running experiments. The minimum receive gain settings at the devices are always different. While performing the experiments, we took care to isolate the experimental setup from highly reflective metallic surfaces and external transmissions, as is typical in a lab environment.

These experimental results have provided us with performance benchmarks that will focus future work on further optimization and sophistication of the MATLAB-based link layer. This framework can be extended to perform evaluation studies on the co-existence of LTE and 802.11 Wi-Fi networks.

In Chapter. 3, we develop a handoff and medium access protocol that allows vehicles to dynamically switch between the mmWave and THz links for high bandwidth data transfer operations. We derive the capacity of the network that results from the protocol operation, and demonstrate how the switching action between these two access methods results in significant improvements over a single and constant choice. Furthermore, we propose an optimal procedure at the SDN controller for scheduling multiple vehicles for accessing a given small cell tower. Since the search for the optimal scheduling is a NP-hard problem, we design a computational-feasible greedy scheduling algorithm, exhibiting a polynomial-time complexity and excellent performance with respect to the optimal scheduling algorithm. Finally, we quantify the actual end to end data transfer rates possible for two tower locations within the Boston area. The analysis showed that a transfer of around 100 Terabit is possible with a single journey, by controlling the velocity of the mule.

In Chapter. 4, the multi-user access in mmWave V2I communication is motivated by the usage of concurrent independent beams realized using hybrid beamforming antenna arrays. A new user association phase with deterministic switched beam patterns is developed. A resource allocation scheme that maximizes the time-frequency utilization at the Base Station is presented. The scheme performs combinatorial optimization on a packing objective that involves the rectangular bin packing problem. We show that the time-frequency resource utilization improves with increasing density of the vehicles, and decreasing street intensity. A MATLAB-based simulator tool built for the link layer simulations is used to obtain the results.

This thesis has practical relevance and use to the engineering researchers. Code and software developed in this work and supporting published papers have been released for the research community. Our testbed can be used to experiment with and enables creation of new MAC protocols. Our work on link layer switching between mmWave and THz bands can potentially be adopted to model the achievable capacity and determine the optimal 3D placement of an aerial network of drones employing dynamic spectrum switching between distinct spectrum bands.

Bibliography

[1] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11, 1999.

[2] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What Will 5G Be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[3] "Market status statistics - Mobile Experts." [Online]. Available: http://scf.io/en/documents/050_-_Market_status_report_-_Mobile_Experts.php

[4] V. Petrov, A. Pyattaev, D. Moltchanov, and Y. Koucheryavy, "Terahertz band communications: Applications, research challenges, and standardization activities," in *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct 2016, pp. 183–190.

[5] T. S. Rappaport, G. R. M. Jr., M. K. Samimi, and S. Sun, "Wideband millimeter-wave propagation measurements and channel models for future wireless communication system design," *IEEE Trans. Communications*, vol. 63, no. 9, Sept. 2015.

[6] A. D. Angelica. Google's self-driving car gathers nearly 1 GB/sec. [Online]. Available: http://www.kurzweilai.net/googles-self-driving-car-gathers-nearly-1-gbsec.

[7] SAS. Are you ready for your smart car? [Online]. Available: http://www.sas.com/en_us/insights/articles/big-data/the-internet-of-things-and-connected-cars.html.

[8] Toyota. Toyota to display new map generation system at CES 2016. [Online]. Available: http://newsroom.toyota.co.jp/en/detail/10765074/

[9] J. Choi, N. G. Prelcic, R. C. Daniels, C. R. Bhat, and R. W. Heath., "Millimeter Wave Vehicular Communication to Support Massive Automotive Sensing," *accepted to IEEE Communications Magazine*, Sep. 2016. [Online]. Available: http://arxiv.org/abs/1602.06456

[10] A. Mahimkar, A. Chiu, R. Doverspike, M. D. Feuer, P. Magill, E. Mavrogiorgis, J. Pastor, S. L. Woodward, and J. Yates, "Bandwidth on demand for inter-data center communication," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, ser. HotNets-X. New York, NY, USA: ACM, 2011, pp. 24:1–24:6. [Online]. Available: http://doi.acm.org/10.1145/2070562.2070586

[11] Y. Chen, S. Jain, V. K. Adhikari, Z. L. Zhang, and K. Xu, "A first look at inter-data center traffic characteristics via yahoo! datasets," in *INFOCOM, 2011 Proceedings IEEE*, April 2011, pp. 1620–1628.

[12] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Dec. 2008. [Online]. Available: http://doi.acm.org/10.1145/1496091.1496103

[13] A. Kanno, "Fiber-wireless signal transport by terahertz waves," in *2014 International Conference on Advanced Technologies for Communications (ATC 2014)*, Oct 2014, pp. 766–769. [Online]. Available: http://dx.doi.org/10.1109/ATC.2014.7043490

[14] L. Vigneri, T. Spyropoulos, and C. Barakat, "Storage on wheels: Offloading popular contents through a vehicular cloud," in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2016, pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/WoWMoM.2016.7523506

[15] H. Shokri-Ghadikolaei, C. Fischione, G. Fodor, P. Popovski, and M. Zorzi, "Millimeter wave cellular networks: A mac layer perspective," *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3437–3458, Oct 2015.

[16] J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, "Millimeter-wave vehicular communication to support massive automotive sensing," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, December 2016.

[17] E. Uhlemann, "Connected-Vehicles Applications Are Emerging [Connected Vehicles]," *IEEE Vehicular Technology Magazine*, vol. 11, no. 1, pp. 25–96, March 2016.

[18] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, June 2011, pp. 163–168.

[19] J. B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, July 2011.

[20] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: a survey," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 148–157, May 2013.

[21] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "IEEE 802.11ad: directional 60 GHz communication for multi-Gigabit-per-second Wi-Fi [Invited Paper]," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 132–141, December 2014.

[22] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-Wave Cellular Wireless Networks: Potentials and Challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, March 2014.

[23] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter Wave Channel Modeling and Cellular Capacity Evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1164–1179, June 2014.

[24] N. González Prelcic, A. Ali, V. Va, and R. W. Heath, Jr, "Millimeter Wave communication with out-of-band information," *ArXiv e-prints*, Mar. 2017.

[25] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE Std. 802.11b-1999, 1999.

[26] MathWorks, Inc. (2016) USRP Support from Communications System Toolbox. [Online]. Available: http://www.mathworks.com/hardware-support/usrp.html

[27] R. Subramanian. (2016) 80211bSDR: IEEE 802.11b standard compliant link layer code for MATLAB-based SDR developed by GENESYS Lab at Northeastern University. [Online]. Available: https://github.com/80211bSDR

[28] ——. (2016) IEEE 802.11b link layer for MATLAB-based SDR. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/55784-ieee-802-11b-link-layer-for-matlab-based-sdr

[29] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[30] K. R. Chowdhury and T. Melodia, "Platforms and testbeds for experimental evaluation of cognitive ad hoc networks," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 96–104, 2010. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2010.5560593

[31] Ettus Research, Inc. (2015) USRP N200/N210 Networked Series. [Online]. Available: http://www.ettus.com

[32] B. Drozdenko, R. Subramanian, K. Chowdhury, and M. Leeser, *Cognitive Radio Oriented Wireless Networks: 10th International Conference, CROWNCOM 2015, Doha, Qatar, April 21-23, 2015, Revised Selected Papers*. Cham: Springer International Publishing, 2015, ch. Implementing a MATLAB-Based Self-configurable Software Defined Radio Transceiver, pp. 164–175. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24540-9_13

[33] GNU Radio Project. (2015) GNURadio: The free and open source radio ecosystem. [Online]. Available: http://www.gnuradio.org

[34] C. R. A. Gonzalez, C. B. Dietrich, S. Sayed, H. Volos, J. D. Gaeddert, P. M. Robert, J. H. Reed, F. E. Kragh *et al.*, "Open-source SCA-based core framework and rapid development tools enable software-defined radio education and research," *Communications Magazine, IEEE*, vol. 47, no. 10, pp. 48–55, 2009.

[35] M. Simon, P. Laaser, V. Filimon, H. Geltinger, D. Friedrich, Y. Raman, and R. Weigel, "An 802.11 a/b/g RF Transceiver in an SoC," in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, 2007.

[36] S. G. Kim and S. H. Cho, "Implementation of an embedded software modem platform," in *Advanced Technologies for Communications, 2008. ATC 2008. International Conference on*. IEEE, 2008, pp. 356–359.

[37] Y. Jiao, X. Wang, G. Xiao, and H. Chen, "Design, implementation and testing of an IEEE 802.11 b/g baseband chip," in *ASIC, 2007. ASICON'07. 7th International Conference on*. IEEE, 2007, pp. 934–937.

[38] WARP Project, Rice University. (2015) Wireless open-access research platform. [Online]. Available: http://warp.rice.edu/index.php

[39] M. Duarte, A. Sabharwal, V. Aggarwal, R. Jana, K. Ramakrishnan, C. W. Rice, and N. Shankaranarayanan, "Design and characterization of a full-duplex multiantenna system for WiFi networks," *Vehicular Technology, IEEE Transactions on*, vol. 63, no. 3, pp. 1160–1177, 2014.

[40] C. Hunter, L. Zhong, and A. Sabharwal, "Leveraging physical-layer cooperation for energy conservation," *IEEE T. Vehicular Technology*, vol. 63, no. 1, pp. 131–145, 2014. [Online]. Available: http://dx.doi.org/10.1109/TVT.2013.2271121

[41] H. V. Balan, M. Segura, S. Deora, A. Michaloliakos, R. Rogalin, K. Psounis, and G. Caire, "USC SDR, an easy-to-program, high data rate, real time software radio platform," in *Proceedings of the second workshop on Software radio implementation forum*. ACM, 2013, pp. 25–30.

[42] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker, "Sora: High performance software radio using general purpose multi-core processors," in *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2009, April 22-24, 2009, Boston, MA, USA*, J. Rexford and E. G. Sirer, Eds. USENIX Association, 2009, pp. 75–90. [Online]. Available: http://www.usenix.org/events/nsdi09/tech/full_papers/tan/tan.pdf

[43] J. van de Belt, P. D. Sutton, and L. Doyle, "Accelerating software radio: Iris on the Zynq SoC," in *21st IEEE/IFIP International Conference on VLSI and System-on-Chip, VLSI-SoC 2013, Istanbul, Turkey, October 7-9, 2013*, 2013, pp. 294–295. [Online]. Available: http://dx.doi.org/10.1109/VLSI-SoC.2013.6673295

[44] R. Marlow, C. Dobson, and P. Athanas, "An enhanced and embedded GNU radio flow," in *Field Programmable Logic and Applications (FPL), 2014 24th International Conference on*. IEEE, 2014, pp. 1–4.

[45] B. Özgül, J. Langer, J. Noguera, and K. Visses, "Software-programmable digital pre-distortion on the zynq soc," in *2013 IFIP/IEEE 21st International Conference on Very Large Scale Integration (VLSI-SoC)*, Oct 2013, pp. 288–289.

[46] C. Dobson, K. Rooks, and P. M. Athanas, "A power-efficient FPGA-based self-adaptive software defined radio," in *24th International Workshop on Power and Timing Modeling, Optimization*

and Simulation, (PATMOS), Palma de Mallorca, Spain, September 29 - Oct. 1, 2014. IEEE, 2014, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/PATMOS.2014.6951901

[47] T. Collins and A. Wyglinski, "Skynet: SDR-Based Physical Simulation Testbed," in *Vehicular Technology Conference (VTC Fall), 2015 IEEE 82nd*, Sept 2015, pp. 1–2.

[48] MathWorks, Inc. (2016) comm.AGC System Object. [Online]. Available: http://www.mathworks.com/help/comm/ref/comm.agc-class.html

[49] ——. (2016) Communications System Toolbox. [Online]. Available: http://www.mathworks.com/help/comm/index.html

[50] J. M. Jornet and I. F. Akyildiz, "Channel modeling and capacity analysis for electromagnetic wireless nanonetworks in the terahertz band," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3211–3221, 2011.

[51] Q. Xia, Z. Hossain, M. Medley, and J. M. Jornet, "A link-layer synchronization and medium access control protocol for terahertz-band communication networks," in *Proc. of GLOBECOM*, 2015.

[52] C. Han, A. O. Bicen, and I. F. Akyildiz, "Multi-ray channel modeling and wideband characterization for wireless communications in the terahertz band," *IEEE Trans. Wireless Communications*, vol. 14, no. 5, May 2015.

[53] J. M. Jornet and I. F. Akyildiz, "Femtosecond-long pulse-based modulation for terahertz band communication in nanonetworks," *IEEE Transactions on Communications*, vol. 62, no. 5, pp. 1742–1754, 2014.

[54] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A Roadmap for Traffic Engineering in SDN-OpenFlow Networks," *Comput. Netw.*, vol. 71, pp. 1–30, Oct. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2014.06.002

[55] I. F. Akyildiz, S. Nie, S.-C. Lin, and M. Chandrasekaran, "5G roadmap: 10 key enabling technologies," *Computer Networks*, vol. 106, pp. 17 – 48, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128616301918

[56] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM*

Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1355734.1355746

[57] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009)*, pp. 1–51, July 2010.

[58] K. Tsukamoto, S. Matsuoka, O. Altintas, M. Tsuru, and Y. Oie, "Distributed channel coordination in cognitive wireless vehicle-to-vehicle communications," *Proc. of International Conference on Wireless Access in Vehicular Environments (WAVE2008)*, Dec 2008.

[59] N. González-Prelcic, R. Méndez-Rial, and R. H. Jr., "Radar aided mmWave beam alignment in V2I communications supporting antenna diversity," in *Information Theory and Applications Workshop (ITA)*, February 2016.

[60] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, 2014.

[61] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE Jour. Selected Areas in Communications*, vol. 32, no. 6, pp. 1164–1179, 2014.

[62] A. Goldsmith, *Wireless Communications*. Cambridge: Cambridge University Press, 2005.

[63] [Online]. Available: http://www.datacentermap.com

[64] [Online]. Available: https://www.cfa.harvard.edu/hitran

[65] J. V. Siles, E. Schlecht, R. Lin, C. Lee, and I. Mehdi, "High-efficiency planar Schottky diode based submillimeter-wave frequency multipliers optimized for high-power operation," in *2015 40th International Conference on Infrared, Millimeter, and Terahertz waves (IRMMW-THz)*, Aug 2015, pp. 1–1.

[66] K. M. K. H. Leong, X. Mei, W. Yoshida, P. H. Liu, Z. Zhou, M. Lange, L. S. Lee, J. G. Padilla, A. Zamora, B. S. Gorospe, K. Nguyen, and W. R. Deal, "A 0.85 THz Low Noise Amplifier

Using InP HEMT Transistors," *IEEE Microwave and Wireless Components Letters*, vol. 25, no. 6, pp. 397–399, June 2015.

[67] T. Rappaport, R. Heath, R. Daniels, and J. Murdock, *Millimeter Wave Wireless Communications*, ser. Communication engineering and emerging technologies. Prentice Hall, 2014. [Online]. Available: https://books.google.com/books?id=_Tt_BAAAQBAJ

[68] K. Haneda, L. Tian, Y. Zheng, H. Asplund, J. Li, Y. Wang, D. Steer, C. Li, T. Balercia, S. Lee, Y. Kim, A. Ghosh, T. A. Thomas, T. Nakamura, Y. Kakishima, T. Imai, H. C. Papadopoulos, T. S. Rappaport, G. R. M. Jr., M. K. Samimi, S. Sun, O. H. Koymen, S. Hur, J. Park, J. C. Zhang, E. Mellios, A. F. Molisch, S. S. Ghassamzadah, and A. Ghosh, "5G 3GPP-like Channel Models for Outdoor Urban Microcellular and Macrocellular Environments," *CoRR*, vol. abs/1602.07533, 2016. [Online]. Available: http://arxiv.org/abs/1602.07533

[69] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What Will 5G Be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[70] L. Simic, N. Perpinias, and M. Petrova, "60 ghz outdoor urban measurement study of the feasibility of multi-gbps mm-wave cellular networks," *CoRR*, vol. abs/1603.02584, 2016. [Online]. Available: http://arxiv.org/abs/1603.02584

[71] H. Mehrpouyan, M. R. Khanzadi, M. Matthaiou, A. M. Sayeed, R. Schober, and Y. Hua, "Improving bandwidth efficiency in e-band communication systems," *IEEE Communications Magazine*, vol. 52, no. 3, pp. 121–128, March 2014.

[72] G. L. Stüber, *Principles of Mobile Communication (2Nd Ed.)*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[73] A. Dadgarpour, B. Zarghooni, B. S. Virdee, and T. A. Denidni, "One- and two-dimensional beam-switching antenna for millimeter-wave mimo applications," *IEEE Transactions on Antennas and Propagation*, vol. 64, no. 2, pp. 564–573, Feb 2016.

[74] V. Va, X. Zhang, and R. W. Heath, "Beam switching for millimeter wave communication to support high speed trains," in *Vehicular Technology Conference (VTC Fall), 2015 IEEE 82nd*, Sept 2015, pp. 1–5.

[75] A. Alkhateeb, O. E. Ayach, G. Leus, and R. W. Heath, "Hybrid precoding for millimeter wave cellular systems with partial channel knowledge," in *Information Theory and Applications Workshop (ITA), 2013*, Feb 2013, pp. 1–5.

[76] R. W. Heath, N. Gonzlez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave mimo systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, April 2016.

[77] A. Alkhateeb, G. Leus, and R. W. Heath, "Limited Feedback Hybrid Precoding for Multi-User Millimeter Wave Systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6481–6494, Nov 2015.

[78] J. G. Andrews, T. Bai, M. N. Kulkarni, A. Alkhateeb, A. K. Gupta, and R. W. H. Jr., "Modeling and Analyzing Millimeter Wave Cellular Systems," *CoRR*, vol. abs/1605.04283, 2016. [Online]. Available: http://arxiv.org/abs/1605.04283

[79] C. N. Barati, S. A. Hosseini, S. Rangan, P. Liu, T. Korakis, S. S. Panwar, and T. S. Rappaport, "Directional Cell Discovery in Millimeter Wave Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 12, pp. 6664–6678, Dec 2015.

[80] H. Shokri-Ghadikolaei, C. Fischione, P. Popovski, and M. Zorzi, "Design aspects of short-range millimeter-wave networks: A MAC layer perspective," *IEEE Network*, vol. 30, no. 3, pp. 88–96, May 2016.

[81] V. Va, J. Choi, and R. W. H. Jr., "The Impact of Beamwidth on Temporal Channel Variation in Vehicular Channels and its Implications," *CoRR*, vol. abs/1511.02937, 2015. [Online]. Available: http://arxiv.org/abs/1511.02937

[82] A. Hammoudeh and D. Scammell, "Measurements and characterisation of RMS delay spread and coherence bandwidth in indoor radio channel at millimetre waves," in *High Frequency Postgraduate Student Colloquium, 2002.7th IEEE*, 2002, pp. 7 pp.–.

[83] M. G. Sanchez, A. Hammoudeh, E. Grindrod, and A. Siamarou, "Coherence bandwidth measurements and analysis for millimetre-wave mobile communications," in *Antennas and Propagation, 1999. IEE National Conference on.*, April 1999, pp. 89–92.

[84] H. H. Hmimy and S. C. Gupta, "Statistical model of delay spread and coherence bandwidth for wide-band pcs at millimeter-waves in an urban mobile radio environment," in *Communications,*

1996. ICC '96, Conference Record, Converging Technologies for Tomorrow's Applications. 1996 IEEE International Conference on, vol. 2, Jun 1996, pp. 1232–1235 vol.2.

[85] M. K. Samimi and T. S. Rappaport, "Local multipath model parameters for generating 5G millimeter-wave 3G-like channel impulse response," in *2016 10th European Conference on Antennas and Propagation (EuCAP)*, April 2016, pp. 1–5.

[86] G. R. MacCartney, M. K. Samimi, and T. S. Rappaport, "Exploiting directionality for millimeter-wave wireless system improvement," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 2416–2422.

[87] J. H. Kim, Y. K. Yoon, Y. J. Chong, and H. J. Hong, "Millimeter-wave delay spread measurement and simulation at los urban low-rise environments," in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct 2015, pp. 1194–1196.

[88] H. J. Thomas, R. S. Cole, and G. L. Siqueira, "An experimental study of the propagation of 55 ghz millimeter waves in an urban mobile radio environment," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 1, pp. 140–146, Feb 1994.

[89] J. Jylänki, "A thousand ways to pack the bin – a practical approach to two-dimensional rectangle bin packing," 2010. [Online]. Available: https://github.com/juj/RectangleBinPack

[90] E. U. T. R. Access, "Further advancements for E-UTRA physical layer aspects," *3GPP Technical Specification TR*, vol. 36, 2010.

Appendix A

Proof of Proposition 1

To prove the proposition, first, we note that, due to the relative movement, the distance between the mule and the SD-BS is a function $f(\cdot)$ of the time, whose expression depends on the mule mobility patterns, i.e., $d(t) = f(t)$.

Since the closed-form expression for the channel capacity derived in (3.3) is a function of the distance, for each time $t \in [t_{\text{in}}, t_{\text{out}}]$, the relative distance $d(t)$ has to be computed in order to evaluate the corresponding capacity.

By accounting for this, the proof easily follows by observing that, according to the proposed protocol, $i)$ the transmitter and the receiver can exchange data only if their relative distance $d(t)$ at a certain time $t \in [t_{\text{in}}, t_{\text{out}}]$ is smaller than $d_{\text{th}}^{\text{mm}}$; $ii)$ when at a certain time $t \in [t_{\text{in}}, t_{\text{out}}]$ the distance $d(t)$ is $d_{\text{th}}^{\text{THz}} < d(t) \leq d_{\text{th}}^{\text{mm}}$, the communication in mmWave band is not affected by a reduction of the available capacity for transmitting control packets, since the LTE interface is used for this purpose (i.e., to return ACKs); $iii)$ when at a certain time $t \in [t_{\text{in}}, t_{\text{out}}]$ the distance $d(t)$ is $0 < d(t) \leq d_{\text{th}}^{\text{THz}}$, the communication in THz band does not suffer a reduction of the available capacity for transmitting control packets, since the mmWave interface is used for ACKs. Hence, all the times $t \in [t_{\text{in}}, t_{\text{out}}]$ such that the corresponding distances $\{d(t)\}$ are smaller than $d_{\text{th}}^{\text{mm}}$, i.e., for which the transmitter and the receiver are in contact, may be ideally dedicated for data transfer.

Appendix B

Proof of Corollary 1

The proof easily follows by accounting for the result in Proposition 1 as well as the hypothesis of uniform strict movement. Specifically, we first observe that if the mule is traveling along the path between the point A and E, as depicted in Fig. 3.2, according to a uniform strict movement, at each time t belonging to $[t_{\text{in}}, t_0]$ the corresponding distance $d(t)$ belong to $\mathcal{R}_{mm} \bigcup \mathcal{R}_{THz}$. In addition, since the velocity is uniform, all the distances covered during such a movement contribute equally to the computation of the overall capacity $\mathcal{C}^{\mathcal{O}}$ that, as a consequence, can be computed as:

$$\mathcal{C}^{\mathcal{O}} = \frac{1}{d_{\text{th}}^{\text{mm}} - d_{\text{min}}} \int_{d_{\text{min}}}^{d_{\text{th}}^{\text{mm}}} \mathcal{C}(\eta) d\eta \tag{B.1}$$

Then we observe that, to compute the net transferred bits, it is sufficient to multiply such an overall capacity with the average time T_c in which the transmitter and the receiver are in contact, i.e., the time spent to travel the path from A to E. In fact, as observed in the proof of Proposition 1, since the ACKs are sent through the second-best available option, there is no reduction in the available capacity for transmitting control packets. Hence, the entire T_c may be ideally dedicated to data transfer.

However, such a contact time has to be reduced by ϵ_s^{mm} to account for the time spent at the start of the mmWave communication to synchronize the transmitter and the receiver at a finer granular level. In fact, as detailed in Section 4.1, although the transmitter and the receiver know the positions of each others, around point A in Fig. 3.2, a granular synchronization is needed. Similarly, around point B in Fig. 3.2, an additional time ϵ_s^{THz} is devoted for synchronizing the transmitter and the receiver at a finer granular level to start the THz communications.

As detailed in Section 4.2, the interaction time of the vehicle with the data center is divided into distinct uplink (UL) followed by downlink (DL) phases. This implies that the contact time has

to be reduced by an additional quantity ϵ_{tr} to account for the switching time transceivers spend to change their operational mode. Hence, by accounting for the above analysis we get that the time available for data transmission is given by:

$$T = T_c - \epsilon_s^{\text{mm}} - \epsilon_s^{\text{THz}} - \epsilon_{tr} \tag{B.2}$$

By multiplying (B.2) with (B.1), the proof easily follows by further observing that, under the hypothesis of uniform strict movement, T_c is given by $\frac{2d_{\text{th}}^{\text{mm}} \cos\alpha}{v}$, where v is the uniform average velocity and α is the angle formed by: (i) the distance $d_{\text{th}}^{\text{mm}}$ between the mule and the tower at time t_{in}, and (ii) the direction of the movement.

Appendix C

Proof of Proposition 2

By accounting for constraint (3.19), at most one vehicle can be scheduled by the data center in each time slot. Hence, the average number of bits N_v exchanged with vehicle v during the entire time horizon $\mathcal{K} = \{k_i, \ldots, k_e\}$ is obtained as sum of bits n_v^k exchangeable in time slot k, for each time slot k assigned to vehicle v, i.e., $N_v = \sum_{k \in \mathcal{K}} \phi_v^k n_v^k$.

When two consecutive time slots are assigned to different vehicles, a scheduling overhead cost has to be paid. To account for such an event, we define the indicator function χ_v^k as in (3.23), and by exploiting the equation (3.3), it results:

$$n_v^k = \int_{(k-1)T + \chi_v^k T_v^O}^{kT} C_v(d_v(t))dt. \tag{C.1}$$

The proof easily follows by noting that the total number N of exchanged bits in the considered time horizon $\mathcal{K}$ is given by the sum of the average numbers of bits $\{N_v\}$ exchanged with the vehicles in $\cup \mathcal{V}_k$