Universidade de Aveiro
2021

Tânia Soraia da Rocha Ferreira

Técnicas Multi-Camada para Melhoria da Fiabilidade do Backhaul mmWave

A Multi-Layer Approach to Enhance Reliability in mmWave Backhaul

Universidade de Aveiro
2021

Tânia Soraia da Rocha Ferreira

Técnicas Multi-Camada para Melhoria da Fiabilidade do Backhaul mmWave

A Multi-Layer Approach to Enhance Reliability in mmWave Backhaul

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Duarte Miguel Garcia Raposo, Investigador auxiliar do Instituto de Telecomunicações , e da Doutora Susana Isabel Barreto de Miranda Sargento, Professora catedrática do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho à minha mãe e à minha avó.

o júri / the jury

presidente / president

Professor Doutor Adão Paulo Soares da Silva

Professor Associado da Universidade de Aveiro

vogais / examiners committee

Doutor Rodolfo Alexandre Duarte Oliveira

Professor Auxiliar com Agregação da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Professora Doutora Susana Isabel Barreto de Miranda Sargento

Professora Catedrática do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

agradecimentos / acknowledgements

Em primeiro lugar queria agradecer à minha mãe, o maior exemplo de força e resiliência conheço. Obrigada pela tua dedicação e por fazeres da minha educação superior uma prioridade. Extendo também os meus agradecimentos à minha família, em especial à minha avó, por acreditarem em mim e pelo apoio incodicional.
Um obrigada especial ao meu namorado, João, por todas as palavras de apoio e, principamente, pela compreensão demonstrada ao longo deste trabalho.
Queria deixar também um agradecimento a todos os colegas de curso que marcaram o meu percurso académico durantes estes 5 anos, em especial ao Francisco Pinto, Miguel Carvalhosa, Hugo Leal, Gonçalo Cardoso e Jorge Silva pela ajuda e companhia que me deram durante as noites passadas a fio estudar ou a fazer projetos no DETI. Agradeço de igual forma às minhas amigas Maria Carvalhais, Andreia Figueiredo, Ângela Fernandes e Vânia Marques, pelo suporte que me deram nos meus momentos de desânimo.
Agradeço ainda aos meus colegas do NAP do Instituto de Telecomunicações, por me terem acolhido tão bem no grupo, por todo o conhecimento e pelo apoio demonstrado ao longo deste ano. Em especial, agradeço à Ana Almeida e ao Alexandre Figueiredo, que sempre se mostraram disponíveis para me esclarecer dúvidas. Agradeço também ao meu orientador Duarte Raposo e à professora Susana Sargento por me terem apostado em mim, proporcionando-me a oportunidade de participar num projeto desta escala. Obrigada pelo apoio e orientação prestada durante a realização desta dissertação.
Por fim, agradeço ao Instituto de Telecomunicações pelas condições de acolhimento.
Este trabalho foi parcialmente financiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER), através do Programa Operacional Regional do Centro (CENTRO 2020) do Portugal 2020 e por Fundos Públicos Nacionais através da FCT I.P. (OE) [Projeto SNOB-5G com o Número 045929 (CENTRO-01-0247-FEDER-045929)].

Palavras Chave

Backhaul 5G, Backhaul sem fios, ondas milimétricas, IEEE 802.11ad, fiabilidade da rede, classificação da qualidade da ligação, previsão da qualidade da ligação, network coding.

Resumo

O aumento exponencial do volume de tráfego observado na última década tem pressionado os opereradores de redes celulares para fornecerem serviços de tráfego *multi-gigabit* de uma forma económica e ecologicamente sustentável. Como tal, um número elevado de *small cells* tem sido instalado em cenários urbanos, para aumentar a capacidade da rede e suportar os requisitos desafiantes de largura de banda, capacidade e latência das actuais redes 5G. Contudo, a densificação extrema da rede tem um custo elevado quando utilizadas ligações de fibra. O *backhaul* sem fios oferece assim uma alternativa escalável e com uma boa relação custo-benefício às ligações de fibra, especialmente na banda de ondas milimétricas, onde larguras de banda grandes estão disponíveis. Ainda assim, a elevada susceptibilidade a obstruções nesta banda, só permite o funcionamento correcto em ambientes em que exista linha de vista. Para assegurar uma rede resiliente, devem existir mecanismos para mitigar o impacto negativo causado pela obstrução nas múltiplas camadas. Esta tese foca-se na implementação de mecanismos que melhorem a fiabilidade e resiliência de uma rede *backhaul* de ondas milimétricas, mais especificamente, os ganhos que a previsão antecipada da qualidade da ligação poderia ter no aumento da adaptabilidade da rede. Para esse efeito, foi instalada com sucesso uma testbed WiGig controlada por SDN num cenário exterior. Os dados recolhidos num ambiente real permitiram caracterizar o impacto de obstruções de curta e longa duração em múltiplas camadas da rede. Esta análise mostrou que uma obstrução contínua causa atrasos que podem chegar a quase um segundo e PER máximas de 40%, mesmo em cenários com ajuste automático da modulação. Por outro lado, uma obstrução de curta duração causou variações abruptas da taxa de dados, atrasos máximos que podem chegar a meio segundo, e perdas de pacotes máximas de 9%. Numa segunda etapa do trabalho, os dados reais foram também utilizados para desenvolver um mecanismo que classifica, com uma F1-score de 97%, a qualidade de uma ligação *mmWave* sob diferentes níveis de obstrução. Para além disso, foi demonstrado que um modelo de *deep learning* é capaz de utilizar as métricas de multiplas camadas de rede para detectar padrões temporais que podem ser utilizados para prever o valor das métricas mais relevantes para os próximos 3 segundos. O erro associado à previsão de cada métrica está contido dentro do intervalo de variação observado para uma ligação mmWave não obstruído. Por fim, foi explorado o potencial de técnicas lineares de network coding em aumentar a robustez em redes de backhaul baseadas em ondas milimétricas. Foi observado que a codificação de pacotes com RLNC de comprimento fixo aumenta a probabilidade de atingir taxas de entrega de 100%, com um aumento pouco significativo do overhead, em links mmWave altamente obstruídos. Adicionalmente, o comportamento da *testbed* utilizada foi replicado no ns-3 para desenvolver um modelo de simulação e facilitar a validação futura dos mecanismos propostos. Numa fase futura do projecto, estes mecanismos irão permitir conceber um *backhaul* de ondas milimétricas com uma elevada resistência e tolerância a falhas.

Keywords

5G backhaul, wireless backhaul, mmWave, IEEE 802.11ad, network reliability, link quality classification, link quality prediction, network coding

Abstract

With the exponential increase of traffic volume observed in the recent decade, cellular operators can no longer satisfy multi-gigabit traffic demands in an economical and ecologically sustainable way. As such, a dense rollout of small cells is being deployed in urban scenarios to increase coverage and support the challenging bandwidth, capacity, and latency requirements of current 5G networks. However, extreme network densification comes at a cost since the ultra-dense urban scenarios produce a massive amount of backhaul traffic in the core network, potentially becoming the network bottleneck. Typically, small cell base stations are connected to the core network through fiber cables, which offers high reliability at a high cost. Wireless backhauling offers network operators a scalable and cost-effective alternative to the wired backhaul, especially in the mmWave band, where large bandwidths are available. Still, this band suffers high susceptibility to the obstruction, which only allows correct operation under Line-of-Sight conditions (LOS). To ensure a truly resilient network, mechanisms should be in place to mitigate the negative impact caused by LOS obstruction at multiple layers. This thesis focuses on implementing mechanisms that improve the network reliability and resilience of a blockage-prone mmWave backhaul. More specifically, the gains that predicting in advance the wireless link quality could have on network adaptability. For that purpose, an SDN-controlled testbed using the standardized IEEE 802.11ad was successfully deployed in an outdoor scenario. The data collected in this network enabled characterizing the impact of long-term and short-term blockage on mmWave links. The conducted analysis shows that long-term obstruction causes maximum delays of about 1 second and PERs up to 40%, even with an automatic modulation adjustment. On the other hand, short-term blockage causes sudden data rate decreases of a few hundred Mbps, delays up to half a second, and packet losses up to 9%. The experimental metrics were also used as the basis for developing a new link quality classifier, which was shown to classify the wireless link quality with a test F1-score of 97%. Moreover, a novel deep learning forecasting model was shown to accurately detect network patterns to predict most of the relevant link's KPIs for the next 3 seconds. The prediction error associated with each metric is within the variation range observed for an unobstructed mmWave link. Furthermore, it has also been found that random linear network coding techniques can be effectively used to increase the robustness of highly variable mmWave networks. It was shown that rateless RLNC increases the likelihood of achieving 100% data delivery success rates, with low encoding overhead, in highly blocked links. Also, the behavior of the experimental testbed was replicated using ns-3 to develop an accurate simulation model that could be used for future validation of the proposed mechanisms. In a more advanced stage of the project, the proposed mechanisms will be used together to design a truly highly-resilient and fault-tolerant mmWave backhaul.

Contents

List of Figures

List of Tables

Abbreviations

5G Fifth Generation

A-BFT Association Beamforming Training

AC-RLNC Adaptive Causal Network Coding with Feedback

ADDTS Add Traffic Stream

AGC Automatic Gain Control

AI Artificial Intelligence

AIC Akaike Information Criterion

ANN Artificial Neural Network

AoA Angle of Arrival

AoD Angle of Departure

AP Access Point

ARIMA Autoregressive Integrated Moving Average

ATI Announcement Transmission Interval

AWV Antenna Weight Vector

BDDP Broadcast Domain Discovery Protocol

BHI Beacon Header Interval

BI Beacon Interval

BIC Bayesian Information Criterion

BRP Beam Refinement Phase

BSS Basic Service Set

BTI Beacon Transmission Interval

CART Classification and Regression Trees

CBAP Contention Based Access Period

CBP Contention Based Period

CE Channel estimation

CM Cluster Manager

COTS Commercial Off-the-Shelf

CRC Cyclic Redundancy Check

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CTS Clear To Send

CV Cross-validation

DCF Distributed Coordination Function

DIFS Distributed Inter Frame Space

DL Deep Learning

DMG Directional Multi-Gigabit

DoF Degree of Freedom

DTI Data Transmission Interval

EDCA Enhanced Distributed Channel Access

EDMG Enhanced Directional Multi-Gigabit

EMS Element Management System

FEC Forward Error Correction

FFNN Feed-Forward Neural Network

GP Grant Period

HCCA HCF Controlled Channel Access

HCF Hybrid Coordination Function

IBSS Independent Basic Service Set

KPI Key Performance Indicator

L-BFGS Limited-memory Broyden–Fletcher -Goldfarb–Shannon

L2TP Layer 2 Tunneling Protocol

LDPC Low-Density Parity-Check

LLDP Link Layer Discovery Protocol

LNC Linear Network Coding

LOS Line-of-sight

LSTM Long-Short-Term Memory

LTE Long Term Evolution

MAC Media Access Control

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MCS Modulation and Coding Scheme

MEC Multi-Access Edge Computing

MIMO Multiple-Input Multiple-Output

ML Machine Learning

MLP Multi-layer Perceptron

MSDU MAC Service Data Unit

MSE Mean Squared Error

MU-MIMO Multi-user Multiple-Input Multiple-Output

NAV Network Allocation Vector

NC Network Coding

NLOS Non-line-of-sight

NVF Network Functions Virtualization

OF OpenFlow

OFDM Orthogonal Frequency Division Multiplexing

ONOS Open Network Operating System

PBSS Personal Basic Service Set

PCF Point Coordination Function

PCP PBSS central point

PER Packet Error Rate

PHY Physical Layer

PL Packet Loss

PoP Point of Presence

QoE Quality of Experience

QoS Quality of Service

RAN Radio Access Network

RBF Radial Basis Function

RCPI Received Channel Power Indicator

RF Random Forest

RLNC Random Linear Network Coding

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

RSSI Received Signal Strength Indicator

RTS Request To Send

RTT Round-trip Time

RXSS Receive Sector Sweep

SARIMA Seasonal Autoregressive Integrated Moving Average

SBS Small cell Base Station

SC Single Carrier

SDN Software-Defined Networking

SDR Software-Defined Radio

SLS Sector Level Sweep

SNR Signal to Noise Ratio

SON Self-organising Network

SP Service Period

SPR Service Period Request

SR-ARQ Selective Repeat Automatic Repeat Request

SSW Sector Sweep

STF Short Training Field

SU-MIMO Single-user Multiple-Input Multiple-Output

SVM Support Vector Machine

TCP Transmission Control Protocol

TDD Time Division Duplex

TDMA Time Division Multiple Access

TXSS Transmit Sector Sweep

UDP User Datagram Protocol

VoIP Voice over Internet Protocol

WLAN Wireless Local Area Network

WPAN Wireless Personal Area Network

XOR Network Coding Exclusive-OR Network Coding

Chapter 1

Introduction

1.1 Context & Motivation

The increasing popularity in bandwidth-hungry applications has led to a massive increase in network traffic in the last few years. Today, with an annual growth of 25% of mobile users this number is expected to reach 80 billion by 2030 [1]. To meet the high traffic demands, 5G cellular systems are currently being deployed. The Fifth Generation (5G) of wireless communications is expected to support data rates 100 higher than the current 4G LTE, 3-5 times the spectral efficiency and 10-100 times the energy efficiency [2].

5G leverages the frequently unused mmWave spectrum (that ranges from 3-300 GHz) to achieve multi-gigabit data rates with low latency. Furthermore, it employs different spatial processing techniques like massive-Multiple-Input Multiple-Output (MIMO), that increase the spectral efficiency and channel capacity even further, as it is possible (due to the higher frequencies), to pack a large number of antennas into a smaller form factor. The leap seen in the transition from 4G to its 5G counterpart cannot be seen as a simple upgrade of specific technologies and techniques to increase capacity and achieve greater spectral efficiency, but rather as a new cellular network whose inner workings are completely different owning to the higher frequencies it uses, as well as the new services and the number of devices supported. In reality, achieving all the benefits that 5G proposes requires huge changes in how mobile networks work and their underlying infrastructure, especially in the backhaul/transport layer.

In order to provide increased coverage and capacity in densely populated urban areas, the use of small cells has emerged with 5G alongside typical macrocells and femtocells stations. Small cells are low-powered radio access nodes which can operate in both the unlicensed and licensed mmWave frequencies that typically cover distances up to 200 meters under NLOS conditions. Their main goal is to extend cellular networks to provide connectivity in densely populated regions with a rapidly increasing user base. Thus, in ultra-dense small cell deployments, the 5G backhaul must accommodate hundreds of gigabits of traffic from the core network. As such, two different alternatives can be used to provide backhaul connectivity to

5G small cells. Fibre connections are the most reliable option, and they provide the highest throughput. However, they are not always a viable option due to two main reasons: 1) in ultra-dense scenarios the CAPEX grows excessively as the number of small cells increases; and 2) fiber backhauls do not have a dynamic and re-configurable topology that can be changed according to the specific time-dependent data capacity demands.

Several wireless alternatives exist in the sub-6 GHz and mmWave bands to provide backhaul connectivity. Nonetheless, despite the current advances in enhancing spectrum efficiency, sub-6GHz wireless frequencies cannot achieve data rates exceeding 10 Gbps, which may not always be enough for the envisioned 5G networks and services. This has driven ongoing research for mmWave-based solutions that exploit the high bandwidth available in the mmWave spectrum to provide 5G connectivity to small cells, which fulfils the challenging bandwidth, capacity, and latency requirements. Wireless backhauling in this band offers network operators a scalable and cost-effective alternative to the wired backhaul.

Nevertheless, using such high frequencies inevitably leads to higher propagation losses. Using the free space model, simply changing the frequency from 3 to 30 GHz, results in an additional propagation loss of 20 dB for the same transmitter-receiver distance [3]. To cope with the higher losses, steerable MIMO antenna arrays with beamforming capabilities are usually employed to create highly directional beams where the power is focused to cover a specific direction. Still, this is not sufficient to compensate all the additional losses. Hence, the distances typically covered by these shorter wavelengths are lower than those covered by previous cellular systems. However, the most critical point of mmWave communications is its high susceptibility to blockage. Millimetre signals suffer less diffraction than microwave signals, which makes them much more vulnerable to blockage. Therefore, a sudden obstruction of the line of sight path is enough to cause a significant decrease in the maximum achievable throughput and lead to maximum delays above acceptable limits for critical applications. In the worst cases, link obstruction may even cause link failure.

To ensure a truly resilient network, there must be several mechanisms in place that allow to mitigate the negative impact caused by LOS obstruction at both network and MAC layers. Firstly, possible obstructions that may lead to link failures may be detected ahead of time, so that it is possible to trigger a traffic rerouting action at layer 3, using for instance a SDN controller to redirect it, before the negative impact of the obstruction is actually felt in the network performance. Moreover, ahead of time, link quality predictions may be used to switch the radio to one suffering from less interference in layer 2.

Lastly, network coding techniques can also be an interesting technology to reduce losses and to handle the high variations of delay caused by obstructions caused by obstruction. Such techniques can also make the use of the ahead-of-time predictions to increase or decrease the code rate.

This thesis focuses on researching mechanisms that improve the network reliability and resiliency of a blockage-prone 5G mmWave backhaul, more specifically the gains that predicting in advance the wireless link quality could have. This work aims to study the possibility of properly classifying a mmWave link quality according to expected blockage duration, and the possibility of predicting link quality metrics before a LOS obstruction occurs.

This work was developed in the context of the SNOB-5G project. SNOB-5G is an inniative led by Ubiwhere, in collaboration with Instituto de Telecomunicações, Universidade de Coimbra and the Research Laboratory of Electronics at MIT, for the development of a self-optimized, intelligent, and fault-tolerant wireless backhaul solution for 5G networks that may use 5G units installed in urban furniture (such as lamp posts) to meet the high bandwidth, capacity, and latency requirements in challenging urban scenarios [4]. In the scope of this project, we intend to leverage Artificial Intelligence (AI), Network Coding, Multipath routing, Software-Defined Networking (SDN), Network Functions Virtualization (NVF) and dynamic caching techniques based on Multi-Access Edge Computing (MEC) [5] to design a fully autonomous network system apt to work in a mesh topology and capable of accommodating innovate services. The resilience mechanisms proposed in this thesis will allow to reroute priority flows over alternative links and to further enhance the performance of network coding techniques in a SDN-controlled mmWave backhaul.

1.2 Objectives

The main goal of this work is the definition and implementation of new mechanisms to improve the network resilience of a 5G backhaul under different types of induced obstructions. These mechanisms can be used to further act in multiple layers (Transport, MAC and network) by allowing to reroute higher priority traffic with high QoS requirements to links that are expected to be more reliable, and by speeding up the network coding techniques designed to improve robustness in data transmission. This thesis has the following objectives:

- Characterize the real impact caused by long-term and short-term obstruction on real millimetre-wave backhaul links;
- Study the possibility of properly classifying mmWave link quality under different obstruction conditions using real datasets;
- Study the possibility to perform accurate and ahead of time predictions of the most relevant link KPIs using real datasets;
- Explore the type of robustness improvement provided by network coding techniques in links suffering from blockage.

1.3 Contributions

In order to develop mechanisms to improve resiliency and reliability in a real 5G backhaul network, it was necessary to acquire the KPIs of links for different types of obstruction. The backhaul deployed consisted of three nodes that were installed on the roof of the Instituto de Telecomunicações de Aveiro, a very isolated area without much interference, thus allowing for testing under a fully controllable environment. These nodes operate in the 60GHz band and use IEEE 802.11ad standard. The data acquired was extensively analyzed to characterize the impact of short-term and long-term obstruction in a time-variable mmWave link. The real dataset was then used to develop mechanisms for classifying and predicting the wireless link quality and its KPIs in a given time window. At the final step, it was assessed the specific improvements that network coding could provide to obstructed links and the impact that such mechanisms could have on SDN approaches.

The lack of configuration support in the equipment and the inability to access the packets circulating on the network make the task of simulating and testing the reliability mechanisms a time-consuming and difficult process. To facilitate the testing process, the physical testbed's behavior was replicated in the ns-3 simulator. However, the chosen module still had its limitations since it is not extended to operate with multiple radios nor supports the introduction of static and moving obstructions in a link. Thus, work extending the ns-3 module to our particular scenario was performed. The main contribution of this Thesis include:

1. Development of data acquisition network probes to collect both physical and transport layer metrics;

2. Creation of a dataset which reflects the behavior of two mmWave links in three different blockage states: non-blocked, long-term blocked, and short-term blocked;

3. Extensive characterization of the impact that different types and levels of obstruction have on multiple layer's KPIs (physical, network and transport metrics);

4. Machine Learning (ML) and Deep Learning (DL)-based pipeline to classify mmWave link quality according to the type of obstruction registered in a given time window: no obstruction, short-term blockage, and long-term blockage;

5. DL-based pipeline to forecast physical and transport KPIs that are expected to be observed for a given mmWave link in the next 3-second window;

6. Evaluation of network coding techniques in a real prone-blockage mmWave backhaul as a joint work between Instituto de Telecomunicações and MIT researchers;

7. Extension of the ns-3 WiGig module to simulate the real backhaul network, also as a joint work between IT researchers. This includes the development of a new module to simulate obstruction in a link, the extension of the original WiGig module to work

with four radios instead of only one, and the development of new antenna codebooks to simulate the properties of our node's radios.

As a result of contributions 1 to 5, a scientific paper is being developed, "Improving mmWave Backhaul Reliability: An ML-based Approach", that leverages the wireless link quality classifier and the KPIs forecasting methods to propose a multi-layer link quality forecasting framework that estimates the future quality of a mmWave link.

Contributions 5 and 6 will also generate two additional research papers in collaboration with MIT during different development stages of the SNOB-5G project. Currently, a paper is being developed, "All About Losses: Improve Millimeter-Wave Lossy Communication with Network Coding Techniques", describing the gains that the Adaptive Causal Network Coding with Feedback (AC-RLNC) technique provides in terms of reliability in a real mmWave network. In the future, another research paper will explore the possibility of using the forecasted link's KPIs to adjust the coding rate before sudden obstruction occurs.

With respect to the latter contribution, one scientific paper has been already written, alongside other researchers working in the SNOB-5G project, describing the processes required to adapt the ns-3 WiGig module to represent the deployed mmWave backhaul. The development process was validated in the real testbed through direct comparison of the measured link metrics and those that the simulator estimates in different scenarios.

1.4 Thesis outline

This document is organized in the following manner:

- **Chapter 1 - Introduction:** Contextualizes the thesis's motivation and the problem scope. Also, it describes the main objectives and contributions provided by this work;
- **Chapter 2 - State of the Art:** Presents background theoretical concepts required to understand the topics in discussion throughout the work (e.g., mmWave spectrum properties, the IEEE 802.11ad MAC and PHY layers, DL and ML theory). Moreover, it describes state of the art in improving the reliability of millimeter wave-based communications, in particular for the IEEE 802.11ad standard;
- **Chapter 3 - Experimental Deployment:** Details the backhaul infrastructure alongside a functional description of the 60GHz network nodes. Then, it describes the process of developing network and physical metrics data acquisition probes;
- **Chapter 4 - Multi-layer Data Analysis:** Provides a detailed analysis of the impact that each of the three scenarios has on the metrics at multiple layers;
- **Chapter 5 - Machine Learning Pipeline:** Describes the general pipeline used for processing the raw data acquired that is later fed to ML and DL models. After this

description, the chapter is organized into two subsections, one per machine learning problem. In the first subsection, the three link quality classifiers developed (Support Vector Machine, Random Forest and Multi-layer Perceptron) are described and compared in terms of performance for different aggregation window sizes. In the last subsection, the processes used to select and validate a Long-Short-Term Memory (LSTM) neural network that forecasts physical, network, and transport layer metrics for the next 3 seconds is detailed;

- **Chapter 6 - mmWave Modelling and Network Coding:** Presents additional contributions that have been made in the context of the SNOB-5G research project in order to test and validate resilience mechanisms that address the challenges of blockage-prone mmWave based backhauls. This chapter works on two different fronts. First, the process of adapting and extending the existing ns-3 WiGig module is presented. Then, the potential of network coding techniques to increase the reliability of the mmWave network suffering from obstruction is demonstrated;

- **Chapter 7 - Conclusions and future work:** Summarizes our findings and discusses future research directions.

Chapter 2

State of the Art

5G mobile networks are currently being deployed to meet challenging multi-gigabit data rates and low latency requirements for the next generation of services and applications. Recent advancements in MIMO and phased-arrays technologies in the mmWave spectrum make it possible to accommodate hundreds of gigabits of backhaul traffic generated from 5G ultra-dense small-cells, offering a scalable and cost-effective alternative to the typical fiber backhaul, but at the expense of higher susceptibility to obstruction. Thus, several proactive resilience and fault recovery mechanisms must be in place, in multiple layers, to ensure a truly resilient mmWave network.

This chapter provides an overview of the background concepts that support the work developed towards a reliable mmWave backhaul. This chapter is structured as follows. Section 2.1.1 discusses the propagation characteristics of mmWave communications in the 60 GHz band, and presents a description of the new PHY and MAC layers of the IEEE 802.11ad/ay standards. In the end, the architecture of a wireless 5G backhaul is described. Section 2.4 discusses how link's KPIs can be used alongside network coding techniques to improve the performance of an SDN-controlled mmWave network. Section 2.5 presents a comprehensive study of experimental works made at the 60 GHz band, in terms of hardware used, environments tested, their main results, and respective limitations. Moreover, it discusses current network resilience mechanisms proposed by the research community on different levels of proactiveness.

2.1 60 GHz networks

To introduce the topic, this section starts by describing the propagation characteristics of the millimeter waves (the so called V-Band), and then presents the standards and mechanisms available, as well as some of their limitations.

2.1.1 The V-Band

The current demand for extremely high data rates, alongside the spectrum scarcity at the sub-6 GHz bands, has led to an increasing interest in the mmWave frequency band (30 to 300GHz) from academia and industry. Compared with existing wireless technologies, such as WiFi and 4G,mmWave communications provide larger bandwidths, which directly translates to higher data transfer rates [6], and permits packing a large number of antenna elements into a small form factor due to the decrease in wavelength [7].

There has been significant work on the V-band, located between 57-64 GHz and 64-71 GHz, due to its ability to achieve extremely high data rates with a moderate level of licensing or no licensing at all. More specifically, the 7GHz band between 57-64 GHz has been used in the standardised IEEE 802.11ad, a new multi-gigabit wireless standard for WiFi at 60 GHz.

Nonetheless, the 60GHz band has several drawbacks. Firstly, due to the shorter wavelength, it suffers from high propagation losses, limiting the maximum coverage distance. This attenuation is conditioned by the propagation environment and accounts for losses associated with free-space propagation, rain and atmospheric and molecular absorption. For instance, the 60GHz band is subject to a free-space path loss at least 20dB higher than the attenuation in the 2.4/5 GHz bands. Besides, as shown in figure 2.1, the oxygen absorption reaches its peak (ranging from 15 to 30 dB/km) at the 60GHz frequency for frequencies below 100GHz [8]. At frequencies above 10GHz, the interaction of electromagnetic waves with the atmosphere and particles such as rain, snow, and hail also increases the attenuation. The particles' size is in the order of magnitude of the wavelengths used in the mmWave band and hence produces a scattering effect on the transmitted signal [9] [10]. The 60GHz signals also suffer from higher penetration loss when encountering obstacles like brick and the human body. For instance, the penetration loss at 60GHz is near 6 dB for a drywall, 3.6 dB in clear glass and 10.2 dB for mesh glass [11]. All these previously phenomenons contribute to the attenuation increase, making the task of maintaining a reliable connection at mmWave frequencies rather challenging.

Steerable MIMO antenna arrays with beamforming capabilities are commonly employed to counteract this strong signal attenuation. Due to its small wavelength, small electronically steerable antenna arrays can be realized, allowing it to concentrate the signal towards a specific direction with high gain, while keeping a very low gain in other directions. The use of these narrow beams decreases interference and enables spatial reuse [12]. That is, the mmWave base stations coverage areas can be overlapped with no significant system interference. While highly directional links address the path loss issue, it also introduces other challenges. Firstly, the transmitter and receiver must find an optimal beam combination to establish a directional link. For that purpose, there must be some sort of beam training mechanism in place. Secondly, relying on directional links makes the 60GHz communication much more vulnerable to obstruction. For instance, a human in the LOS path between the transmitter and the receiver can attenuate the signal by 20-30 dB which may result in temporary or long-term link outage [13]. Also, relying on NLOS transmission is not always an option as the received signal was shown in some experiments to receive less 10 dB and 20 dB using first-order and second-order reflected signals [14]. Thirdly, directional antennas reduce the number of multipath components which could be used to retrieve the original signal using equalization techniques [15]. Furthermore, recent work has shown that the mmWave channel is sparse, as independently of the number of beam directions available, the transmitter can only reach the receiver via a small set of dominating signal paths [16]. Therefore, systems in the 60GHz band should be designed with all these aspects in mind to ensure that the network can support reliable connections alongside high data rates.

Given the potential of mmWave communications, multiple international organizations have emerged for the standardization of protocols in that band, including ECMA [17], IEEE 802.15.3 Task Group 3c [18], IEEE 802.11ad standardization task group [19], the WirelessHD consortium [20] and the Wireless Gigabit Alliance (WiGig) [21].

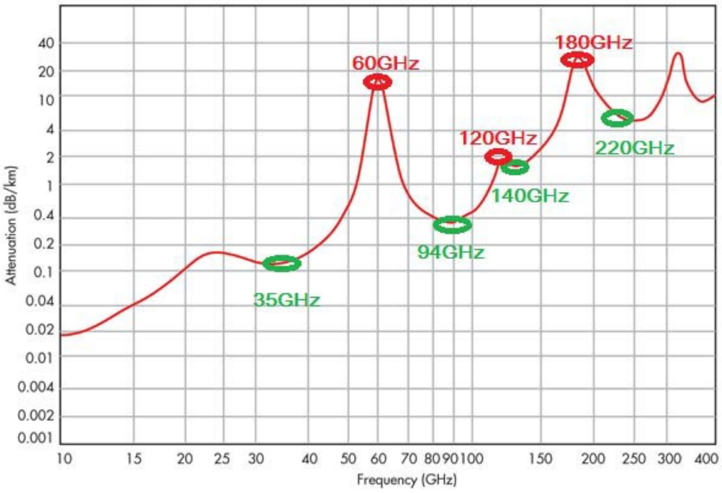


Figure 2.1. Atmospheric and molecular absorption at mmWave frequencies [22].

The standardization process began in 2007 with the WirelessHD, a global WPAN standard at 60 GHz developed for streaming high-definition multimedia content over a very short range. The standard, which had its first release in January 2008, was designed to support data rates up to 3.086 Gbps and was backed by a total of 26 tech companies, such as Broadcom, Intel, LG Electronics, Samsung, Sony. However, this technology was not widely adopted due to the fact that transmitters and receivers are expensive, have high power consumption, and can only withstand distances up to 10 meters [23]. Two years later, the task group TG3c introduced the IEEE 802.15.3c, a mmWave-based physical layer alternative to the existing 802.15.3 Wireless Wireless Personal Area Networks (WPANs). This standard added functionalities to the MAC layer that allows reducing the propagation disadvantages associated with mmWave communications, and provided three PHY operating modes for specific market segments with mandatory data rates exceeding 1Gb/s: Single Carrier mode (SC PHY), High speed interface mode (HSI PHY) and Audio/visual mode of the mmWave PHY (AV PHY). The Single Carrier mode is best suited for kiosk file downloading and office desktop usages. The HSI PHY was designed for bidirectional, NLOS, low latency communication in conference ad-hoc scenarios.

The AV PHY was designed to provide high throughput uncompressed video streaming [24].

In these last few years, the emergence of new applications with higher throughput and lower latency requirements, such as Augmented Reality (AR) and Virtual Reality (VR), led to the formation of the IEEE 802.11 Task Group ay in 2015 to further improve the performance of WiFi at 60GHz. Built on top of its predecessor IEEE 802.11ad, the 802.11ay defines new PHY and MAC layers that enable WiFi devices to achieve theoretical peak data rates up to 100 Gb/s by employing MIMO techniques that support up to 8 data streams, channel bonding, improved channel access, and enhanced beamforming training. Still, it is supposed to ensure backward compatibility and coexistence with 802.11ad DMG STAs operating in the same band [25]. Because the standard was only approved at the beginning of 2021 [26], there is still no IEEE 802.11ay compliant Commercial Off-the-Shelf (COTS) device available on the market.

The following section presents the MAC and PHY layers of IEEE 802.11ad, and describes the enhancements proposed for each of them in the next generation of 60 GHz networks.

2.1.2 WiGig Standard

The 802.11ad amendment defines a new physical and MAC layer in the 60GHz frequency band that provides support for multi-gigabit wireless applications with throughput up to 7 GHz. The IEEE 802.11ad features include fast session transfer for seamless data fallback and data increase between 60GHz and 2.4/5GHz PHYs, and MAC and PHY layer enhancements which include support for directional antennas, beamforming, and spatial reuse techniques. The main PHY layer enhancement presented by 802.11ad is the combination of two modulation schemes, the Orthogonal Frequency Division Multiplexing (OFDM) and Single Carrier (SC). While OFDM is designed to provide throughput up to 7 Gbps, it also requires high energy consumption. SC also achieves high data rates (up to 4.6 Gbps) but with lower power consumption and less complex transceivers. The standard novelty resides in the new Directional Multi-Gigabit (DMG) PHY layer proposed that, based on multiple-antennas beamforming techniques, is capable of compensating the increased attenuation in the 60GHz band and possibly the interference caused by obstacles during the transmission [27]. Due to this, a reliable communication link can only be established after the stations' transmitting and receiving patterns are pointed to each other. For that purpose, the IEEE 802.11ad MAC layer implements a novel beamforming training process employing different medium access schemes and association processes that are described in more detail later on. The directional nature of the communication also increases spatial reuse since multiple links can be established simultaneously with negligible interference, allowing for concurrent transmissions to occur, thus increasing the total throughput. To exploit this capability, the amendment specifies a spatial sharing and interference mitigation mechanism which performs its transmission scheduling based on the assessed amount of interference in overlapped links [28]. At the MAC level, IEEE 802.11ad continues to support the basic MAC access mechanisms of 802.11a/b/g DCF and PCF in addition to those of 802.11e HCF and HCCA, but also introduces an enhanced MAC layer to achieve data rates up to 1 Gbps [29], to cope with directional communication challenges as well as provide support for QoS and coexistence with other 60GHz systems [30]. The access during Contention Based Period (CBP) is based on the modified 802.11e EDCA that is fined-tuned for directional communication, which ensures high throughput and delay requirements Additionally, it also proposes a new aggregation scheme called Video Aggregation MSDU (VA-MSDU) to support video traffic during contention-based access periods [29].

PHY Layer

The new physical layer, the so called Directional Multi-Gigabit (DMG) PHY, defines six 2.16 GHz bandwidth channels around the 60 GHz ISM band (57-66GHz), as can be seen in Figure 2.2. An 802.11ad channel is roughly 50 times wider than the channels available in 802.11n and 14 times wider than the channels defined in 802.11ac [31]. However, some of these are not universally available except for channel 2 (60.48 GHz), which is defined as the default channel for the standard [32].

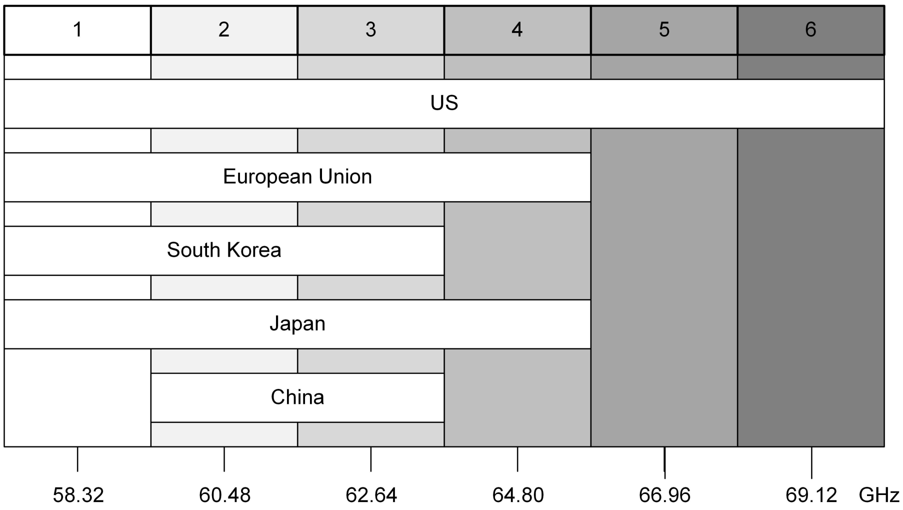


Figure 2.2. Channels in the 60 GHz band [32].

In the PHY layer, four procedures with different modulation and coding schemes are able to format 60 GHz carriers: a) control PHY (MCS 0); b) Single carrier (MCS 1-12) PHY; c) OFDM (MCS 13-24) PHY; and d) low-power SC (and MCS 25-31) PHY. The wide choice of methods makes it possible to fulfill different requirements depending on the usage scenario. Still, the implementation of OFDM and low-power SC is optional. The 11ad PHY also defines four Low-Density Parity-Check (LDPC) codes that are used with these MCS, each with a different rate ($1/2, 5/8, 3/4$ and $13/16$) but with a common rate of $672 bits$ [33]. The modulation schemes, code rates and PHY rates that may be associated with the different PHY modes are summarized in table 2.1.

All DMG PHYs use a packet structure similar to the one in Figure 2.3, only differing in

Table 2.1. IEEE 802.11ad's MCSs [33].

MCS	Modulation	Code rate	PHY rate
Control MCS	$\pi/2$-DBPSK	1/2 (spreading factor of 32)	27.5 Mbps
SC MCS	$\pi/2$-BPSK $\pi/2$-QPSK $\pi/2$-16-QAM	$1/2, 5/8, 3/4,$ $13/16$	385-4620 Mbps
OFDM MCS	Spread QPSK QPSK 16-QAM 64-QAM	$1/2, 5/8, 3/4,$ $13/16$	693-6756.75 Mbps

how individual fields for each MCS used. A Golay sequence forms each field in the packet frame. Each physical packet starts with a preamble that consists of a STF and a Channel estimation (CE) field which is used for packet detection, during AGC, for timing/frequency synchronization and to specify the PHY (Control, SC or OFDM) used in the data following the preamble.

To simplify implementation, both SC and OFDM MCSs share a common preamble. The control MCS uses a longer preamble than the one used in both SC and OFDM PHYs in order to operate at a lower SNR during the initial establishment of the directional links [33]. Following the preamble, there is a header that is different for every PHY that specifies the MCS used, the length of the data field, or a checksum. Following the header, the PHY payload transmits the actual data with the chosen MCS, which is protected by a Cyclic Redundancy Check (CRC). An optional packet field referred to as TRN can be appended to all PHY packets to fine-tune the AWVs of the sectors used in beamforming training, a unique trait of 802.11ad that is described in more detail in the MAC layer description. In that case, the training type (RX or TX) and length of the training field are also set in the packet's header field. This training field is transmitted using $\pi/2-$ BPSK modulation for robust communication. Note that the antenna configuration is only allowed to be changed during the transmission of these training fields, and thus, it must remain the same during the transmission of the preamble, head and data fields [32] [33].

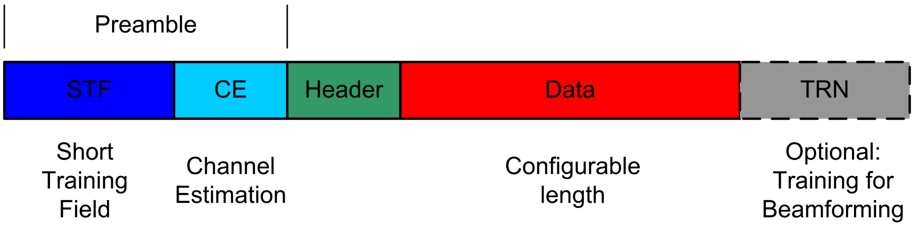


Figure 2.3. IEEE 802.11ad's DMG PHY packet format [32].

The next generation WiGig standard, IEEE 802.11ay, enhances the DMG PHY layer described earlier by providing support for new technologies such as channel bonding and Multiple-Input Multiple-Output (MIMO), which enable reaching faster data rates and covering larger distances than in IEEE 802.11ad. Channel bonding or channel aggregation refers to the ability to create wider channels by aggregating contiguous and/or non-contiguous 2.16 GHz channels to achieve bandwidths as high as 8.64 GHz. Multiple-Input Multiple-Output (MIMO) is also a new technique in IEEE 802.11ay and it enables the simultaneous transmission of multiple data streams, using either a SU-MIMO or MU-MIMO scheme. As the name implies, in Single-user Multiple-Input Multiple-Output (SU-MIMO), the multiple streams can only be sent to and received by a single STA (single user) at a time. In contrast, in Multi-user Multiple-Input Multiple-Output (MU-MIMO), data streams are distributed across multiple users that may be using the same time/frequency resources but that are spatially separable. Therefore, increasing the overall network throughput and spectrum/space utilization. The standard defines a maximum number of spatial streams per station of eight and a downlink MU-MIMO transmission up to eight stations. To support MIMO transmissions and channel bonding while also ensuring backward compatibility, IEEE 802.11ay introduces a new packet structure that contains new fields that support both DMG and EDMG STA modes [25].

MAC Layer

Legacy IEEE 802.11 networks implement a Basic Service Set (BSS) or an Independent Basic Service Set (IBSS) network architecture. In the BSS operating mode, one station, also known as Access Point (AP), acts as a gateway between the wireless and wired backbone. Hence, stations can only communicate with each other through the AP. In contrast, an Independent Basic Service Set (IBSS) or ad-hoc network supports peer-to-peer communication, and thus no AP is required. Any network node can communicate directly with another node within range. However, due to the challenges of directional communication, these two architectures do not suit most 60GHz usage scenarios. As such, the 11ad draft presented a new network architecture named Personal Basic Service Set (PBSS) in which, similar to IBSS, stations communicate in an ad-hoc manner, i.e., they can communicate directly with each other without having to rely on a common access point. Unlike in the IBSS, in the PBSS, one station is required to assume the role of PBSS central point (PCP). The PCP performs the jobs that are typically associated with an AP such as announcing the network and controlling the medium access. The PCP allocates service periods and contention-based periods to accommodate traffic requests of devices as well as provides basic timing to the PBSS [33].

In the lower frequencies IEEE 802.11 networks, the medium access is divided into a sequence of periodical beacon intervals that the AP initiates by transmitting omnidirectional beacon frames to announce the network and maintain its timing. The remaining time is used to transmit data between stations, usually following a contention-based scheme. The IEEE 802.11ad amendment extends this concept to directional 60 GHz networks by proposing a new temporal organization to cope with the challenges of mmWave communications. Figure

2.4 illustrates a typical Beacon Interval (BI) in 802.11ad. The BI is constituted by a Beacon Header Interval (BHI) and a Data Transmission Interval (DTI). The BHI can be further subdivided into three periods known as Beacon Transmission Interval (BTI), Association Beamforming Training (A-BFT) and Announcement Transmission Interval (ATI) [31]. These three sub-intervals are used to perform the steps required to associate each station with PCP, including forming a directional multi-gigabit link between the AP and the STA.

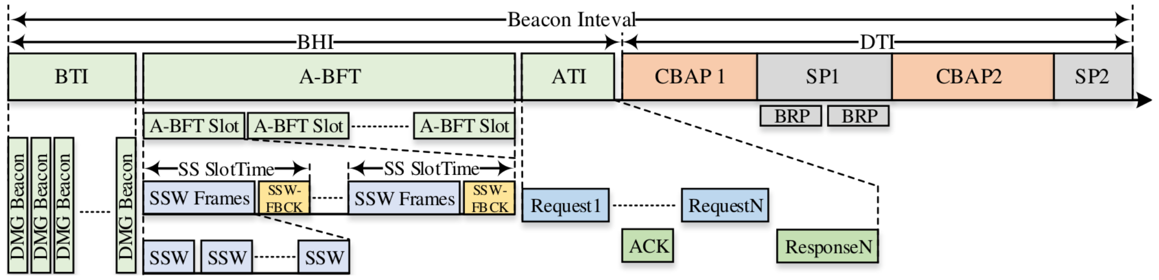


Figure 2.4. IEEE 802.11ad beacon interval structure [34].

During Beacon Transmission Interval (BTI), the PCP transmits a beacon frame in each of its transmit sectors to announce the network and to perform beamforming training of its transmitting antennas. These beacon frames mark the beginning of a BI and are transmitted using the control MCS to enhance robustness in error-prone channels. After that, the Association Beamforming Training (A-BFT) is used to discover the best transmit antenna sector for each STA to communicate with the AP. The ATI sub-interval is later used for exchanging management information between the PCP and the newly beam-trained stations. During this period, an STA can request the AP to allocate a Service Period (SP) in the following DTI for exchanging frames with another device. Since communication during ATI uses beam-trained stations, a higher MCS can be used, which makes the communication more efficient [31].

During Data Transmission Interval (DTI), as the name implies, stations can exchange data frames through a hybrid MAC scheme that comprises CBAP and scheduled SP. During Contention Based Access Period (CBAP) multiple stations contend for the channel using a modified IEEE 802.11 Enhanced Distributed Channel Access (EDCA) fine-tuned for directional communication. The EDCA adopts most of the design principles of 802.11 Distributed Coordination Function (DCF), including exponential backoff and RTS/CTS access methods from the 802.11 Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), but improves it by providing support for frame aggregation, block acknowledgments [31] [34], and support for quality of service using frame priority differentiation [35]. In contrast, a Service Period (SP) is a contention-free period that works similar to TDMA in cellular networks. When an SP allocation request is accepted by the PCP, the channel is reserved for communication between a pair of stations during a given amount of time. Due to the directional communication nature, simultaneous SPs can be scheduled for communication with different STAs due to the spatial reuse ability, as long as the concurrent transmissions do not interfere with each other.

The IEEE 802.11ad's hybrid medium access scheme uses traditional technologies such as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), Time Division Multiple Access (TDMA) and Polling for channel access in directional networks. These three technologies support contention-based channel access, scheduled channel time allocation, and dynamic channel time allocation, respectively. Each scheme addresses different aspects of mmWave communication, which makes it suitable for different IEEE 802.11ad use cases [31]. The operation of these medium access modes is described in the following paragraphs.

The CSMA/CA is a legacy distributed MAC mechanism for WLAN vastly used due to its simplicity and robustness. Under CSMA/CA, a station starts a transmission if the channel is idle for a minimum of a Distributed Inter Frame Space (DIFS) period, and other stations probe the channel for carrier signals. In the case the channel is sensed busy, the station starts a backoff counter uniformly distributed in the $0, ..., W_i$ range, where W_i is the contention window at the i-th attempt. The backoff counter is decremented whenever the channel is idle for at least a DIFS period and frozen when an ongoing transmission in the medium is detected or when the contention-based channel access is suspended. Only after this counter reaches zero, the station can initiate the transmission. The minimum of the contention window is doubled whenever a collision is detected to reduce the likelihood of future collisions. In its basic form, stations using CSMA/CA access the channel with equal probability. However, this mechanism does not allow priority-based service differentiation nor provides guaranteed resources for specific flows. IEEE 802.11ad ensures support for QoS during contention-based channel access through Enhanced Distributed Channel Access which creates virtual MAC queues for each access category (AC) [28]. This standard implements a combination of both physical and virtual carrier sensing. Under that scheme, there is an exchange of directional RTS/CTS messages in the initial contention-based channel access, protecting aligned beams from creating interference as they overhear the exchanged messages. Both RTS and CTS packets contain the proposed duration of the transmission; hence all nodes that overhear this exchange defer their transmissions for the proposed duration by updating their Network Allocation Vector (NAV) with the duration field. However, since these messages are being directionally transmitted, stations pointed towards other directions do not overhear the ongoing transmissions and thus do to defer when they should. This generates a deafness problem within the network. These deaf nodes will try to access the channel unsuccessfully, leading to an excessive increase in their contention windows. Because stations suffering excessive backoff have a lower probability of winning contention, recently active stations with smaller contention windows are favored, allowing them to transmit a high number of frames consecutively. This mainly affects short-term fairness, but in the long run, active stations can be considered to change sufficiently often to compensate for the frequent bursts of active stations [36]. Additionally, since wireless stations can only listen to the medium in a particular direction, there is an increased chance that a station detects the medium free while it is busy, which will likely cause collisions. This deafness problem could be overcome with the employment of omnidirectional receive patterns, but that would defeat the purpose of using directional communication in the first place [28].

To solve these issues, WiGig defined two new access schemes that allow stations to know in advance when a transmission between a pair of stations is expected. These schemes are known as pseudo-static channel time allocation and dynamic channel time allocation and are based in TDMA and Polling, respectively. Both allocate a dedicated SP within a BIs to the communication of two specific nodes but differ in the type of traffic streams they support.

For medium allocation, the standard extends the concept of streams of IEEE 802.11 HCF. As stated by the authors in [31], "a traffic stream is defined as a flow of MAC service data units that has to be delivered subject to certain QoS parameters characterized by a traffic specification." Under this mechanism, each requesting station must define its traffic demands in terms of allocation duration and traffic stream periodicity (isochronous or asynchronous characteristics). An isochronous stream will result in a recurring SP allocation with a constant rate, whereas asynchronous streams will result in non-recurring SP allocation [31]. The scheduled SPs are then propagated by the PCP at the beginning of the next BTI or ATI to all the associated STAs so that every node not involved in the communication can go into a power-saving mode. Also, because every station is aware of its peers due to the centralized approach, the communicating stations can steer their antenna beams towards each other, thus avoiding quasi-omni receiving and transmitting patterns. This makes static time allocations very useful for predictable traffic patterns as it avoids the need for scheduling a new allocation every time, thus limiting the signaling overhead to a minimum [37]. The main drawback is that the channel allocation traffic demands can only be modified during the beginning of the next BTI or ATI and not within that same BI to accommodate bursty traffic [28].

For that purpose, IEEE 802.11ad also specifies a second mechanism, dynamic channel time allocation, that implements polling-based channel access during DTI. Built on scheduled SP and CBAP, the dynamic channel access is an extension of 802.11's PCF mode for directional communication that allows polled stations to request channel time to transmit multiple frames. During the polling period, the PCP polls the remote devices during the ATI period by sending multiple frames, to which stations can reply with a Service Period Request (SPR) message requesting channel time. In the SPR frame, each station declares the resource requirements necessary to satisfy its traffic demands and its QoS constraints. The PCP then allocates channel time within DTI according to the available resources to accommodate the SP requests and announces each granted allocation in a separate Grant Period (GP) by sending directional frames to the involved stations. Once again, since the PCP informs each involved station of the allocation in advance, the receiver knows the direction of the incoming signal, thus preventing the deafness problem. This centralized approach allows the adaptation of the scheduled time allocation dynamically (hence its name) within a BI to react to bursty downstream traffic, which makes it useful for unpredictable transmissions that need to be delivered under specific QoS conditions. The dynamic mechanism can also exploit unused channel time to extend an SP to complete an ongoing communication without additional scheduling or delay. Despite the benefits, the dynamic allocation scheme wastes resources polling stations which adds polling overhead. An example illustrating the dynamic channel allocation is shown in figure 2.5. In the schematic, it is possible to observe that the channel time allocation starts with a polling phase at the beginning of DTI by sending frames to the associated stations, answered with three SPR frames. After that, the PCP performs the dynamic allocations back to back, each following its respective grant period. In case of an allocation between the PCP and non-PCP station, only one grant frame is sent [28] [31].

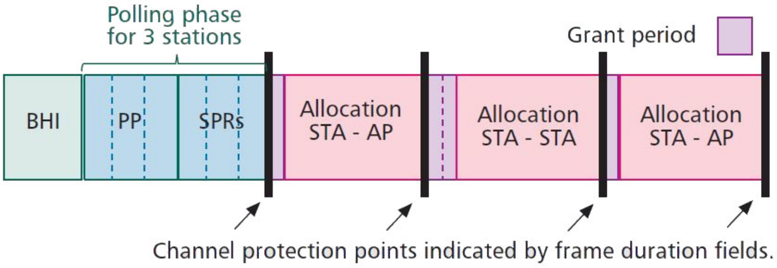


Figure 2.5. IEEE 802.11ad's dynamic channel allocation [31].

In IEEE 802.11ay networks, supporting new technologies such as channel bonding and MIMO, introduces new challenges that need to be addressed at the EDMG MAC layer. For instance, the channel access has to support allocation over multiple channel frequencies and MIMO operation through spatial sharing. The IEEE 802.11ay's medium access is also organized in beacon intervals similar to IEEE 802.11ad's, but now the A-BFT, ATI, DTI are extended to use multiple channels (a primary and secondary channels). Still, to ensure backward compatibility, network announcement and management frames are transmitted during BHI only in the primary channel. In this new standard, the PCP can either allocate an SP for the communication between two STAs over an aggregated channel, or the stations can follow a CBAP and start a transmission only after obtaining a transmission opportunity. The MIMO channel access uses physical and virtual carrier sensing mechanisms and the backoff procedure already employed in 802.11ad to exchange RTS/CTS messages after a network device obtains a Transmit Opportunity (TXOP). In MIMO channels, a TXOP is only obtained after all all the antennas intended to be used to transmit the data streams are idled for a given amount of time. The RTS messages sent indicate to the pairing STA the operating channel frequencies, the bandwidth of the frame, whether the subsequent transmission is made using SU-MIMO or MU-MIMO technologies, and the antenna configurations to be used [25]. This process allows EDMG STAs to transmit multiple streams using multiple transmit antennas to a pairing node with multiple RX antennas, which increases robustness by avoiding the outage when one of the streams is blocked [38].

Beamforming training

In order to cope with the increased attenuation at 60GHz, IEEE 802.11ad establishes directional links through a beamforming training process, which allows determining the appropriate transmit and receive antenna sectors for communication between a pair of stations. In 802.11ad, the AP-STA beamforming training occurs during BTI and A-BFT, while the

STA-STA beamforming training occurs only during DTI before the actual data frame exchange. Beamforming consists of two phases: Sector Level Sweep (SLS) and an optional Beam Refinement Phase (BRP), as shown in figure 2.6. In the first phase, an initial coarse-grained antenna sector is determined for either transmission (TXSS) or reception (RXSS) between two devices. During this process, the involved stations behave once as a transmitter and once as a receiver, as shown in Figure 2.7. For contextualization, the station that transmits first is called initiator while the other is known as responder [31].

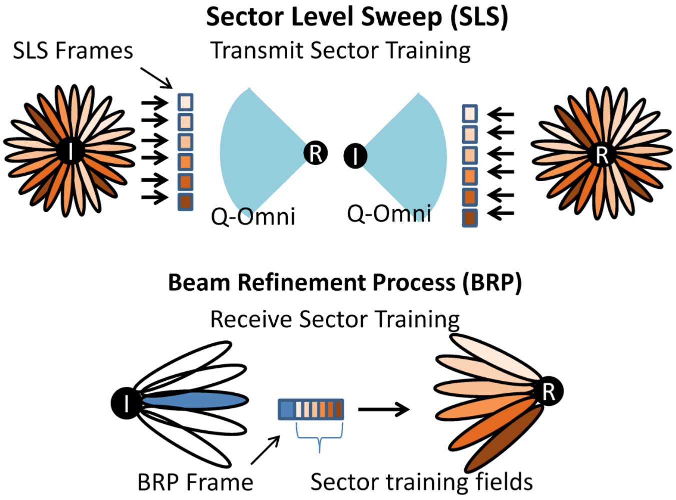


Figure 2.6. IEEE 802.11ad's beamforming training phases [39].

During Transmit Sector Sweep (TXSS) a station finds its best transmit antenna sector by sending SSW frames (or beacons for PCP's TXSS in BHI) frames from each of its sectors with the pairing station configured to receive in a quasi-omnidirectional mode. The sender sets an ID field in every SSW frame sent to identify the sectors, and the strongest one is reported to the pairing station alongside its SNR in either a feedback sector sweep feedback field inside the frames transmitted by the responder's TXSS (R-TXSS) and received by the initiator, or in a dedicated sector sweep frame at the end of R-TXSS. Conversely, during RXSS a station finds its best-receiving antenna sector by testing different receiving patterns when receiving SSW frames sent by the other station.

Overall, four sweep combinations can be used during beam training in Sector Level Sweep (SLS) depending on the type of training performed and who initiated the process. These include Transmit Sector Sweeps at both Initiator (I-TXSS) and Responder (R-TXSS) and Receive Sector Sweeps at both stations (I-RXSS and R-RXSS, respectively). As shown in figure 2.7, when the initiating station performs a Transmit Sector Sweep, the initiator's feedback is carried in every frame sent during the responder sector sweep, which ensures reception even under unknown optimum antenna configurations. In contrast, the responder's feedback is sent in the best transmit direction found using a single feedback SSW frame, to which the former replies with an acknowledgment frame. If the involved stations do not have sufficient antenna gain to establish a reliable link after transmitting sector training, usually a receive sector training is added to the SLS phase. Otherwise, only transmit sector training is performed at the SLS phase, postponing receive sector training to the following BRP phase. Usually, the last SSW ACK exchanged in SLS is also used for exchanging parameters for the next phase [31].

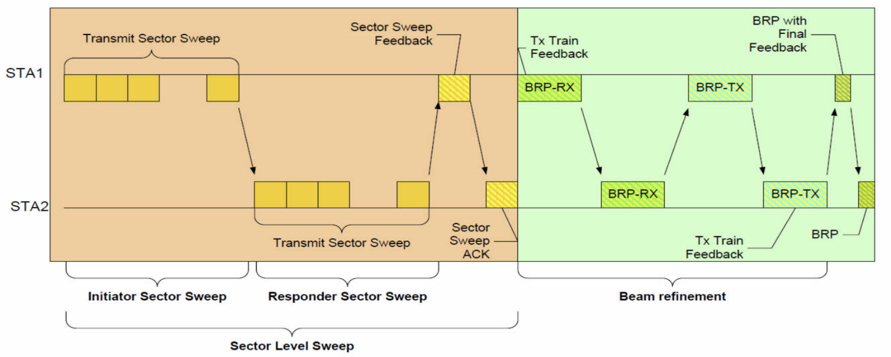


Figure 2.7. Example that illustrates the beamforming training protocol [33].

The following Beam Refinement Phase (BRP) fine-tunes the antenna settings found in the SLS phase to improve the quality of the directional communication even further. There are several mandatory and optional mechanisms to achieve this, but this thesis only focuses on the mandatory ones. Firstly, stations can perform receiving antenna training during the BRP phase if not performed in the initial SLS phase. For that purpose, STAs can request transmit or receive training by exchanging a BRP packet with either a transmit training (TRN-T) or receive training (TRN-R) field attached, as was shown in the general packet structure of figure 2.3. This packet transaction allows optimizing the transmit and receive AWVs independently from the predefined sector patterns tested in SLS. Instead of requiring a frame to be transmitted per sector as in the SLS, the BRP enables multiple antenna configurations to be tested using the same frame, substantially reducing training overhead. Each TRN-T and TRN-R field is transmitted or received with the antenna configuration that is being tested for its signal quality [31]. The requesting STA sets the frame's TX-TRN-REQ header field and appends the number of TRN-T fields necessary to train each configuration in the same BRP frame to perform transmitting antenna training. Alternatively, the station can wait for an ACK frame with a TX-TRN-OK field set from the pairing node before sending a BRP frame with the training fields attached. In the case of receiving antenna training, the requesting STA has to set the L-RX header field of the frame to specify the number of TRN-R fields that the pairing node has to transmit in the next frame. Figure 2.8 illustrates the BRP training stage.

The key concept to take from this interaction is that to find the optimal transmission configurations for a given STA, it must add TRN-T fields to the frames it would typically send to the pairing node. Conversely, to find the optimal receiving patterns for a given node, the pairing node must add TRN-R fields to the frames it would send to it, which will be received with different configurations. The peer node reports the transmitting setting with the highest SNR to the node under TX training, and the best-receiving settings are known right away by the STA since it is in receive mode. Additionally, the amendment defines an optional beam tracking mechanism that periodically adds training sequences (TRN fields) to the PHY frames sent to keep track of the SNR changes and trigger new beamforming training when needed [34].

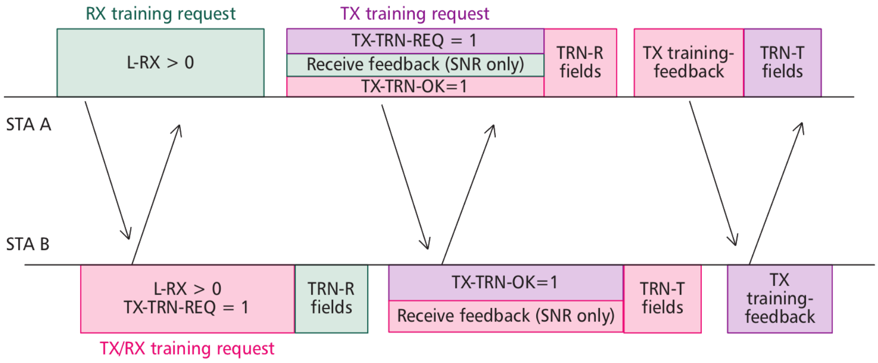


Figure 2.8. Example of a beam refinement protocol transaction [31].

The beamforming concepts described earlier are implemented in 802.11ad by means of different medium access schemes and association processes. Before stations are associated with the AP, they use an adapted version of beamforming training to communicate with the PCP that does not require any prior coordination and allows them to align theirs antenna beams while the best direction between the two is unknown.

This process is initiated in BTI with the PCP sending beacon frames in each sector while the non-PCP STAs listen in a quasi-omni direction. At the end of BTI, the PCP has performed transmitting antenna sector training (I-TXSS). Then, the multiple stations use the reserved contention-based access periods of A-BFT to perform their transmitting antenna beamforming with the PCP (R-TXSS). As previously mentioned, the A-BFT period is used to perform the first phase of the beamforming (SLS) training [31]. An A-BFT slot consists of a fixed-time allocation that accounts for the transmission of multiple SSW frames (one for each sector) by the non-AP station and one SSW feedback frame by the AP [31]. Instead of contending for the channel access through carrier sensing, during A-BFT stations randomly select a slot to perform its responder sector sweep (R-TXSS). Thus a collision is detected by a missing SSW feedback frame in their respective slot. The STA that fails to receive a SSW feedback backs off and retries later. Additionally, there are two mechanisms in place for stations in which the number of antenna sectors exceeds the number of reserved SSW frames per slot to ensure that every non-PCP station can finish its sector sweep: 1) the PCP can send a SSW feedback frame with the sub-optimal transmit sector the results from the incomplete sector sweep ; and 2) the station can contend for extra slots to finish its sector sweep during the same BI or the following; and 3) the beam training can move to a dedicated service period (SP) [31].

STAs that have successfully been associated with the PCP can proceed to perform STA-STA beamforming training before data frame exchange during DTI to form a directional multi-gigabit link. The beamforming training can occur during DTI following one of the three MAC mechanisms described earlier in 2.1.2: contention-based channel access, scheduled channel time allocation, or dynamic channel allocation. In the contention-based channel access, the initiator can directly begin an SLS after it gains control of the channel during a Contention Based Access Period (CBAP). On the other hand, STAs can reserve a dedicated SP by sending a Add Traffic Stream (ADDTS) request frame to the PCP. The PCP replies with an Add Traffic Stream (ADDTS) response frame based on the available resources using an admission control policy. If the allocation is accepted, the beam training parameters are included in the extended schedule element announced by the PCP during the BTI or ATI, which enables both nodes to begin the beamforming process once that SP is reached. STA-STA beamforming training can also be initialized during a dynamic channel allocation. In that case, a node requests an allocation to communicate with another device by sending a Service Period Request (SPR) frame with the intended training parameters [31]. The PCP receives this request during the polling phase and grants channel time by sending the grant frame to both stations. The Data Transmission Interval (DTI) also comprises the BRP phase to improve the signal quality of the AP-STA and STA-STA links formed during the SLS phase, but that exchange does not require specific channel time allocation.

To support MIMO operation in future next-generation WiGig networks, IEEE 802.11ay defines a new mandatory beamforming mechanism composed of two stages (SISO and MIMO phases) for determining the optimal transmitting and receiving antenna configurations to be used in simultaneous transmissions of multiple spatial streams [25]. Also, it defines an optional partial SLS method that enables fast link recovery in low latency applications, a group beamforming method to reduce the overhead of the training procedure by allowing multiple stations to be trained simultaneously, support for simultaneous receive and transmit beamforming training, and support for asymmetric beamforming training. Moreover, the standard specifies a variant of BRP frames, known as short SSW packets, to reduce further training overhead [40].

Since next-generation 60 GHz WiFi features mechanisms that support MIMO technologies and reduce the overhead associated with beamforming training operations, it is expected that the IEEE 802.11ay-based networks will be much more resilient to failures caused by obstruction. Still, it is not yet possible to define their ability to dynamically adapt to highly variable channels, since there are no COTS devices available. Until this point, the literature only focuses on introducing the new technologies supported by IEEE 802.11ay.

2.2 5G Backhaul

To meet the challenging bandwidth, capacity, and latency requirements of current 5G networks, dense rollouts of Small cell Base Stations (SBSs) are being deployed in urban environments, as shown in figure 2.9. The employment of these low-power, low-cost, and short-range SBSs provides an increased coverage and capacity in densely populated urban areas. However, the extreme densification also leads to several challenges that must be addressed, mainly in ultra-dense scenarios where a massive amount of backhaul traffic is generated. Because connecting this large amount of small cells to the core network with the traditional fiber backhaul is unfeasible owing to its high cost, network operators have been looking at wireless communications as a potential candidate for the 5G backhaul. Thanks to the availability of huge bandwidths, it is possible to achieve fiber-like performances at a fraction of the price using mmWave backhauls [41]. Note that, however, 5G networks are expected to adopt a heterogeneous network (HetNet), comprising multiple radio access technologies (RATs), where mmwave cells are overlayed traditional macrocells [42].

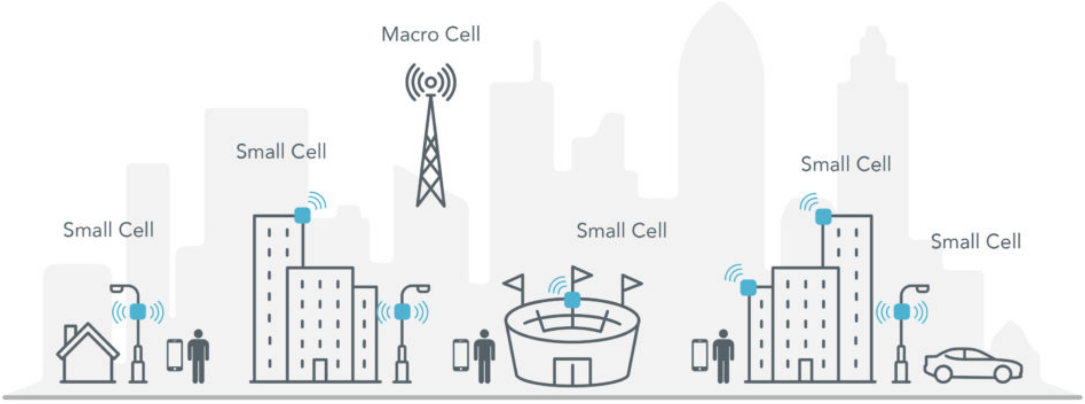


Figure 2.9. Network densification [43].

The IEEE 802.11ad standard presented earlier is currently being proposed by the research community [42] [44] [45] for small cell wireless backhauling due to the support for multi-gigabit data rates, which could be as high as 2.5 Gbps (in the SC PHY). Even with some commercial solutions being proposed by the industry for outdoor environments, such as [46] and [47] (the equipment that is used throughout this work).

Nonetheless, as previously described, this frequency band suffers from high propagation loss that can only be compensated using highly directional antennas, making mmWave links more susceptible to blockage and errors. Therefore, a sudden obstruction of the line of sight path is enough to cause a significant decrease in the maximum achievable throughput and lead to maximum delays above acceptable limits for critical applications.

Nonetheless, the high propagation loss in mmWave spectrum that can only be compensated using highly directional antennas makes mmWave links more susceptible to blockage and errors. Therefore, a sudden obstruction of the line of sight path is enough to cause a significant decrease in the maximum achievable throughput and lead to maximum delays above acceptable limits for critical applications, if proper resilience mechanisms are not implemented.

2.3 Data Processing and Learning

To facilitate the understanding of the aspects covered in the following chapters, the reader should be familiar with some time series, ML and DL-related background concepts. This section is organized as follows. First, definitions of concepts related to time series and machine learning are presented, adapted to the specific context of this work. Then, the theoretical foundations of different classification algorithms used in this thesis are presented. Next, the theoretical foundations of statistical-based forecasting methods and ML-based approaches are presented. In the end, the evaluation metrics that can be used to measure the performance of both forecasting and classification algorithms are described.

2.3.1 Machine Learning Definitions

The mmWave link quality metrics collected, such as Signal to Noise Ratio (SNR) and Received Channel Power Indicator (RCPI), are examples of a time series. A time series consists of data points collected over a defined interval of time whose values are time-dependent. Because data points are collected sequentially, there is usually a correlation between current and previous observations. This is what differentiates time series from cross-sectional data. Also, the statistical properties of a time series do not change over time. Therefore, time series with trends or seasonality components are not stationary. Usually, whenever a time series presents a non-stationary behavior, seasonal and non-seasonal differencing is applied to the series to make it stationary by computing, for instance, the second-order difference between consecutive observations [48].

The task of predicting multiple time steps into the future can be classified as a multi-step time series forecasting problem. This can be performed through recursive multi-step strategy or direct multi-step forecast. In this work, the forecasting methods developed only apply the direct strategy, specifically statistical-based and DL-based methods such as ARIMA and LSTM neural networks, described in the following sections. Because a model that solves this problem has to accept multiple time-dependent KPIs, which may be correlated, to forecast multiple time-steps of future KPIs, it can be classified as a multi-step multivariate input-output model.

Given that IEEE 802.11ad-based networks cannot support a reliable connection under complex environments, a proactive ML-based resilience framework is proposed in this thesis to eliminate the need for constant beamforming training. The approach addresses two distinct ML problems: 1) prediction of the wireless link quality according to the current KPIs, which is a supervised classification problem; and 2) forecasting relevant KPIs based on past observations, which is a supervised regression problem. The supervised learning label comes from the fact that both algorithms are trained with labeled datasets. During the training stage of a supervised Machine learning algorithm, the data fed to the model consists of a series of inputs and the actual outputs to be predicted. This enables the algorithm to determine the appropriate mapping function between inputs and outputs. More specifically, supervised learning problems are either classification or regression tasks according to the desired model output. In our regression problem, the model's output is a set of continuous metrics such as the expected SNR and PER. On the other hand, in a classification problem, the model assigns a label to the input data from a discrete set of categories. In this specific context, predicting a scenario tag that characterizes the wireless link quality is a multi-class classification problem as input data instances should be classified as belonging to one of three classes (normal, long-term blocked and short-term blocked link).

Typically, collecting samples for certain classes is more complex than for others. In this work, we have an imbalanced classification problem, as more samples were collected to characterize links in normal or long-term blocked conditions than for short-term blocked links due to the nature of the tests. Since most ML models are designed to work with balanced datasets, having severe class imbalances may result in poor predictive performance, especially in the minority class, if the appropriate evaluation metrics are not used.

The ML and DL models used have a set of model configurations known as hyperparameters that have to be set before the learning process begins. Hyperparameters are different from model parameters learned during the training process, such as weights and bias of neural networks, because they cannot be estimated from the input data. Examples of hyperparameters include the number of hidden layers in neural networks and the optimization strategy employed to minimize the cost function during the neural network's training. Hyperparameter tuning is used to find the set of hyperparameter combinations that maximize model performance. Common algorithms for performing such process include Grid Search, Random Search, and Bayesian Optimization. Since different datasets have different optimal settings with different impacts on the model performance, each hyperparameter combination must be evaluated on hold-out or cross-validation datasets. The challenge of the entire process relies on how to find a well-performing combination of hyperparameters while also keeping the search time at its lowest. There are two main tuning strategies to address this. The first is the grid search which exhaustively goes through each hyperparameter combination possible within a specified user hyperparameter range. This can be a very time-consuming task, especially with large hyperparameter subsets to search through. The second and last approach is the random search in which each hyperparameter subset is tested randomly rather than exhaustively, decreasing computation time. However, this does not ensure that the optimum local in that grid is found. Throughout this work, only the Random search approach was used to find the optimal set of hyperparameters for both classification and forecasting models. In this work, hyperparameter tuning was used alongside cross-validation to evaluate the generalization capability of the models developed based on results obtained for hold-out data. Thus allowing the comparison and selection of an appropriate model for the specific task. In its basic form, the k-fold CV, the training dataset is split into k smaller subsets. Then, the model is trained $k-1$ times with one of the $k-1$ training folds, and the remaining fold is used to evaluate whether the model is overfitting or underfitting the data. Thus, every data point gets to be in the validation dataset at least once. The overfitting and underfitting conditions are observed when models start to include the data noise in the mapping between input and output variables. It is usually a phenomenon that manifests itself in a testing score that is much higher or lower than the training score.

2.3.2 Machine Learning Classifiers

Currently, several types of classifiers can be used to map input data into a categorical value, which include (but are not limited to) decision trees, logistic regression, Support Vector Machine, Artificial Neural Network, and ensemble methods. In this thesis, Support Vector Machine (SVM), Random Forest (RF) and Multi-layer Perceptron (MLP) classifier were used.

Support Vector Machine (SVM) is a supervised Machine Learning algorithm that works by finding the best hyper-plane in the N-dimensional space that best separates the training data points belonging to each class. After training, this plane is used as a decision boundary for new classifications. On the other hand, Random Forest (RF) is a supervised ML algorithm that uses a collection of individual decision trees that work as an ensemble to predict the class of each data point. The predictions made by each tree are compared, and the most recurring label is used as the algorithm's final prediction.

Multi-layer Perceptron (MLP) is another supervised deep-learning algorithm that falls under the Feed-Forward Neural Network (FFNN) type. MLPs typically have three types of layers (input, hidden, and output), composed of nonlinear computational units, also known as neurons. Thanks to this nonlinear behavior, a sufficient large MLP can perform the mapping between almost any input vector and the corresponding output vector. All the neurons in a MLP layer are fully-connected to neurons of the adjacent layer, thus generating a connection with a given associated weight [49]. In MLPs data flows in the forward direction from the input to the output layer. That is, the output of combining the weighted inputs in the neurons using the activation functions is propagated onto the next layer. However, there is more to it. If MLP were solely based on forward propagation, it would be nearly impossible to find the optimal set of network weights, which would lead to generally bad performing models. In order to find the weights that minimize the error between the predicted and desired output, MLPs uses a backward propagation mechanism. Thanks to this mechanism, the neurons' weights are iteratively adjusted one layer at a time from back to the front of the network, i.e., the weights are updated starting from the output to the input layer according to the amount they contribute to the computed error until the error is bounded to a defined threshold [50]. In this work, instead of using an online learning technique where weights are updated for each training sample, a mini-batch learning approach was chosen where the errors are accumulated and updated periodically for a defined number of training samples, called a batch size. Every time one forward pass and one backpropagation pass have occurred for all training samples, it can be said that an epoch has passed.

2.3.3 Time-series Forecasting

Time series forecasting is a field that the scientific community has widely researched for stock prices forecasting, weather forecasting, and many other applications. Traditional statistical-based methods that work with univariate series such as the popular Autoregressive Integrated Moving Average (ARIMA) or the Seasonal Autoregressive Integrated Moving Average (SARIMA) models, exponential smoothing or GARCH have been some of the models proposed to solve these issues. These models mostly rely on seasonal and trend decomposition and often exploit some combination of the Moving Average, Integration, and Auto-Regressive techniques to find patterns in the data. However, the prediction performance falls behind whenever it is difficult to decompose the series onto seasonal, trend, and non-seasonal components to find patterns. Due to this, more complex modeling techniques are required. Machine Learning, in particular LSTM-based RNNs are being used to model complex non-linear behaviours between features. Besides extracting more complex relationships in multivariate inputs, its main benefit comes from its ability to identify temporal data dependencies. Therefore, an LSTM achieves superior performance over an ARIMA model at a cost of increased complexity. In this work, two different forecasting approaches were compared: a statistical-based ARIMA model and LSTM neural networks. This subsection describes the theoretical foundations of both approaches.

Autoregressive Integrated Moving Average

Autoregressive Integrated Moving Average (ARIMA) is a statistical analysis model that works on the assumption that past values have residual effects on current or future ones to forecast future outcomes. It is usually applied to stationary time series or series that can be made stationary through differencing. In ARIMA, the AR component stands for Auto-Regression and is responsible for modeling the output variable's dependence on its own lagged observations; the I stands for Integration and is the component responsible for differencing non-stationary data (i.e., the statistical properties such as mean and variance of the time series change over time); Finally, the MA component stands for Moving Average and enables modeling the dependency with past forecasting errors in a regression-like manner. The AR, I and MA parts are represented by parameters p, q and d, respectively, in both ARIMA and SARIMA models [51]. Whenever there is seasonality evidence, seasonal differencing is also applied to the time series in an attempt to make it stationary.

$$ARIMA(p, d, q) = AR(p) + I(d) + MA(q) \tag{2.1}$$

where parameters p, d and q represent:

- **p:** The highest order of autoregressive coefficients for the non-seasonal component of ARIMA. Also known as the lag order;

- **q:** The highest order of moving average coefficients for the non-seasonal component of ARIMA. That is, the size of the moving average window;

- **d:** The order of differencing that needs to be applied in order to make a given time series stationary. Also known as the degree of differentiating;

LSTM-based RNN

Usually, in time series, current events are dependent on what has previously happened. For instance, under this specific context, the link is expected to remain long-term blocked and thus face many abrupt variations in the signal quality if the link was also in that state in previous time windows. Due to their ability to identify time sequence patterns, Recurrent Neural Network (RNN) are a natural candidate to perform such task. The RNNs makes use of sequential observations to learn time-dependencies which could be used to forecast future trends accurately. This is where the memory aspect on RNN comes in hand. The network's hidden layers act as an internal storage unit for storing the patterns captured in earlier stages of reading sequential data [52].

To ensure that the temporal information persists on the network, RNN 's internal states (memory) have recurrent connections, which allows it to exhibit dynamic temporal behavior. During training, the algorithm is responsible for finding which inputs to keep and which inputs to discard to produce the desired output. By backpropagating the computed error through the network, it is possible to update the weights of every single neuron that participated in the output calculation, including the neurons far in the back that store previous information used by the neurons in the front. This leads to increasingly lower gradients the further the signal error travels back in time, thus getting to a point where their values actually vanish. This is why RNNs are described in the research community as suffering from a vanishing problem [53], which causes them also to have long-term memory problems [54]. Because of that, a special type of RNNs, the LSTM architecture, has been proposed. These enhanced RNNs are composed of Long-Short-Term Memory cells that are designed to improve gradient flow in deeper networks. What makes them so effective in time series forecasting is their ability to decide which information to keep in its hidden state and what to discard. For that purpose, a LSTM cell has several gates in place that protect and control the cell's internal state, such as the forget gate, the input gate, and the output gate [54]. These gates, which can be observed in the cell diagram provided in figure 2.10, are based on sigmoid layers and work by yielding a number in the range of 0 to 1, which depicts the amount of information that is saved, discarded, and passed to next layers. For a detailed description of how each gate in LSTM works, referrer to [54].

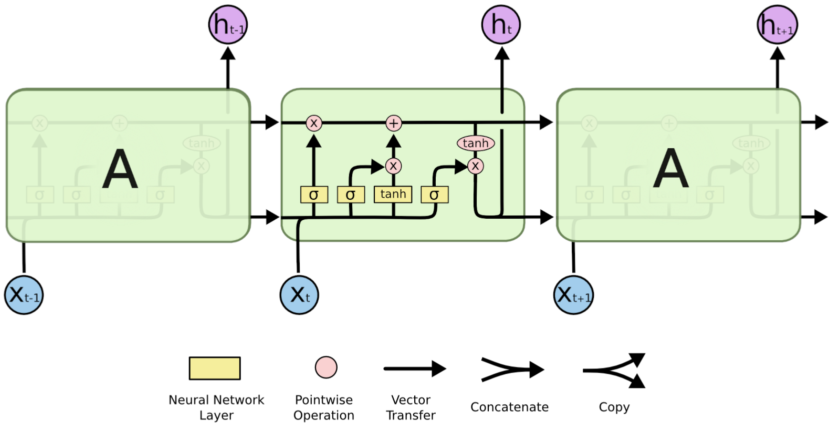


Figure 2.10. Diagram with several LSTM units connected, one per time step. Each unit is composed by four interacting layers [54].

2.3.4 Evaluation Metrics

Choosing the right metric is crucial for adequately evaluating the performance of the ML and DL-based models. Several metrics can be used to measure the performance of classification and regression algorithms, and the choice of one over the other only depends on the specifics of the problem. This section presents the metrics considered for evaluating the the regression and classification models described earlier.

Classification Metrics

There are different metrics for evaluating the performance in classification tasks such as accuracy such as accuracy, precision, recall and F1 score. Accuracy is the most popular metric for classification algorithms. It is calculated as the ratio between the number of correct predictions and the total number of samples to be classified, as shown by the following mathematical equation.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.2}$$

The TP, TN, FN and TN metrics stand for:

- TP: True Positives;
- TN: True Negatives;
- FN: False Negatives;
- TN: True Negatives.

As the name implies, the true positives and true negatives are data points that were assigned the correct label. On the other hand, FN and TN represent the number of data points the model incorrectly classifies.

However, accuracy is not suitable for evaluating the performance of models that use imbalanced datasets, such as the one collected in this thesis. The problem comes from the way it is computed. For instance, recall our specific problem, classifying the input's data wireless link quality according to one of three classes. Even if all samples (let's say 1000) collected for short-term blocked links are incorrectly classified as either a link in normal or long-term blocked condition, if almost all samples from the remaining classes (let's say 20000) are correctly classified, you still get an accuracy of about $accuracy = (20000 + 0)/(20000 + 1000) = 0.95$, which could provide a false sense of performance. Therefore, metrics that severely penalize misclassifications in the minority class should such as precision, recall and F1 score should be used. The appropriate evaluation metric depends on the specific context of the problem, but in general it is good practice to find a balance between both recall and precision using the F1 score.

Recall can be defined as the ratio of true positives to all actual positives, as shown by equation 2.3. Therefore, it is the metric to be applied whenever there is a high cost associated with false negatives. For multi-class problems, recall metric can be computed for each class as the ratio between the number of correctly classified samples and the total number of samples that in reality belonged to that class.

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

Precision can be defined as the ratio of true positives to the total positives observed, as shown by equation 2.4. It can be interpreted as the percentage of actual positives among all predicted positives and thus, may be selected whenever there is a high cost associated with false positives. For multi-class problems, a precision value can be calculated for each class as the ratio of the number of samples that were correctly assigned that class label to the total number of samples classified by the algorithm as belonging to that class (which includes the misclassified ones). This can be easily seen in figure 2.11 which represents the confusion matrix (TP, TN, FP and FN) and the metrics that can be calculated from there.

$$Precision = \frac{TP}{TP + FP} \tag{2.4}$$

As shown in equation 2.5, the F1 score metric performs a harmonic mean of precision and recall scores to compute a single value that reflects a trade-off between them. Since the F1 score takes into account how data is actually distributed, it is more suitable for evaluating models developed with unbalanced datasets. Thus, it was the metric used to select the best performing classification model returned by the hyperparameter tuning process.

$$Precision = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.5}$$

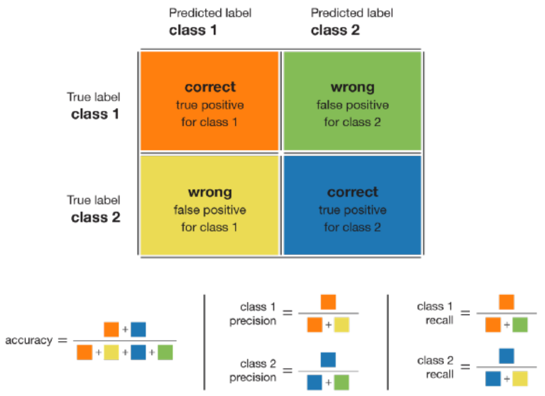


Figure 2.11. Confusion matrix commonly used while solving classification problems and metrics that can be extracted from it [55].

Regression Metrics

Accuracy and similar metrics described earlier cannot evaluate the prediction accuracy made by a regression model as the model's output is now a continuous value rather than a class label. Instead, error metrics specifically designed for these type of problems have to be used, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), RMSE and MAPE. Mean Absolute Error (MAE) (or mean absolute deviation) is the simplest error metric used for regression problems. As the name implies, its value can be computed by averaging the absolute error obtained in the prediction of each input sample. MAE is known for being more robust to outliers than MSE as the high errors they cause are not squared [56].

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y_i - \widehat{Y_i}\right| \tag{2.6}$$

Mean Squared Error (MSE) is the most popular evaluation metric for regression problems. MSE can be calculated by averaging the squared error between the true and predicted values for all N input samples, as shown by equation 2.7. It measures the variance of residuals. Because it is a differentiable function, it was used as the loss function to be minimized during training of the forecasting deep learning model. However, MSE is more sensitive to outliers as it penalizes high errors more than MAE due to its squared error operation [56].

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y_i}\right)^2 \tag{2.7}$$

Root Mean Squared Error (RMSE) is simply the squared root of MSE, as shown by the following equation. It measures the standard deviation of residuals [56].

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y_i}\right)^2} = \sqrt{MSE} \tag{2.8}$$

On the other hand, Mean Absolute Percentage Error (MAPE) is the percentage equivalent of MAE, as shown by the following equation. Since the real value is used to divide the absolute residual error, it should not be used in cases similar to ours where the real values can be zero. For instance, both Packet Error Rate and Packet Loss KPIs can reach zero under normal channel conditions.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|Y_i - \widehat{Y_i}\right|}{Y_i} \tag{2.9}$$

Statistical Metrics

There are two metrics used to score and select a statistical model among a set of models developed with the same data, the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). These criteria select the model which provides the best trade-off between the log-likelihood (the quality of the model's fit on the data) and model complexity. The AIC is a metric score based on frequentist probability and generally tends to allow for the selection of more complex models if that means improved performance. On the other hand, BIC is a score metric based on Bayesian probabilities that often penalizes more complex models [57].

2.4 Improving Backhaul Reliability

In wireless networks, the propagation channel can vary significantly over time, greatly affecting the radio's link quality. This is especially true for mmWave networks due to their high susceptibility to obstruction, thus, not ensuring a reliable connection, i.e., continuity of correct service [58].

IEEE 802.1ad standard defines three reactive mechanisms to deal with obstruction. Firstly, it can switch to an alternative beam pair by triggering a new beamforming process whenever a signal quality decrease below a certain threshold is detected. However, beamforming training operations have an associated high overhead, which translates directly to high delays. For instance, in [59] the beamforming training process triggered by misalignment was reported to be approximately 7.65 ms in an indoor IEEE 802.11ad deployment. This problem is further aggravated in the case of high-frequency short-term obstructions. Secondly, an STA can be used as a relay between a pair of STAs if the direct link is blocked. Lastly, because WiGig is backward compatible with legacy WiFi, the fast session transfer mechanism can switch the traffic to other IEEE 802.11 standards operating at lower frequencies. However, this approach prohibitively leads to a significant throughput degradation [60] Still, these three mechanisms are reactive, and thus actions are taken only after a significant link quality decrease is detected.

Because these approaches cannot provide the necessary robustness to support critical applications, proactive resilience and fault recovery mechanisms should be in place, if possible at multiple layers, to ensure a reliable mmWave performance. The concepts associated with the multiple reliability mechanisms proposed in this thesis are presented in the following section.

2.4.1 Network coding

In traditional networks, packets are simply cached and forwarded by devices from the source to destination, resulting in a separate transmission flow per message being delivered. By employing Network Coding (NC) techniques, intermediate nodes (such as switches and routers) can leverage algebraic functions to encode information from different packets. Depending on the encoding scheme, this packet combination may result in improved security (e.g., to packet sniffing), resilience to losses and link failures [61] [62] and support for Quality of Service (QoS) in both wired and wireless networks. Furthermore, NC was shown to maximize the network's overall throughput [63] [64] [65]. There are several coding techniques, but the most common include Exclusive-OR Network Coding (XOR Network Coding), Linear Network Coding (LNC) and Random Linear Network Coding (RLNC).

The main idea behind RLNC is presented in the diagram of figure 2.12 [66]. To encode using Random Linear Network Coding (RLNC), the data source (e.g., video) is divided into smaller chunks of equal length that are known as a "generation" of the original message. This lowers the computational cost of the encoding and decoding operation. Each generation is further subdivided into symbols of equal size (x bits), and the total number of symbols per generation is commonly referred to as the generation size. Each symbol contains the same number of bits, and its size is set according to the chosen Galois field (2^x). Then, the symbols are multiplied by scalar coding coefficients w_i that are drawn from be selected GF field. In the last step, symbols are linearly combined to produce the encoded symbol. The original symbols are then transmitted alongside the coded symbols to compensate for channel erasures. The coefficients used to generate multiple coded symbols compose the local encoding vector. The overall coding process from source to destination nodes forms the global encoding matrix [66]. This encoding operation can be viewed as creating a system of linear equations formed with random coefficients that the receiver must solve to retrieve the original data symbols. The number of necessary DoF needs to be known prior to adding the redundancy. Thus, it is necessary to implicitly or explicitly transmit the coding vectors from the sender to a receiver so that the operations are reversible.

A unique trait that Random Linear Network Coding (RLNC) techniques present over Linear Network Coding (LNC) is the ability to re-encode information without decoding it first. Recoding enables intermediate nodes between source and destination to use already coded symbols, alongside partially decoded symbols or uncoded symbols to generate new

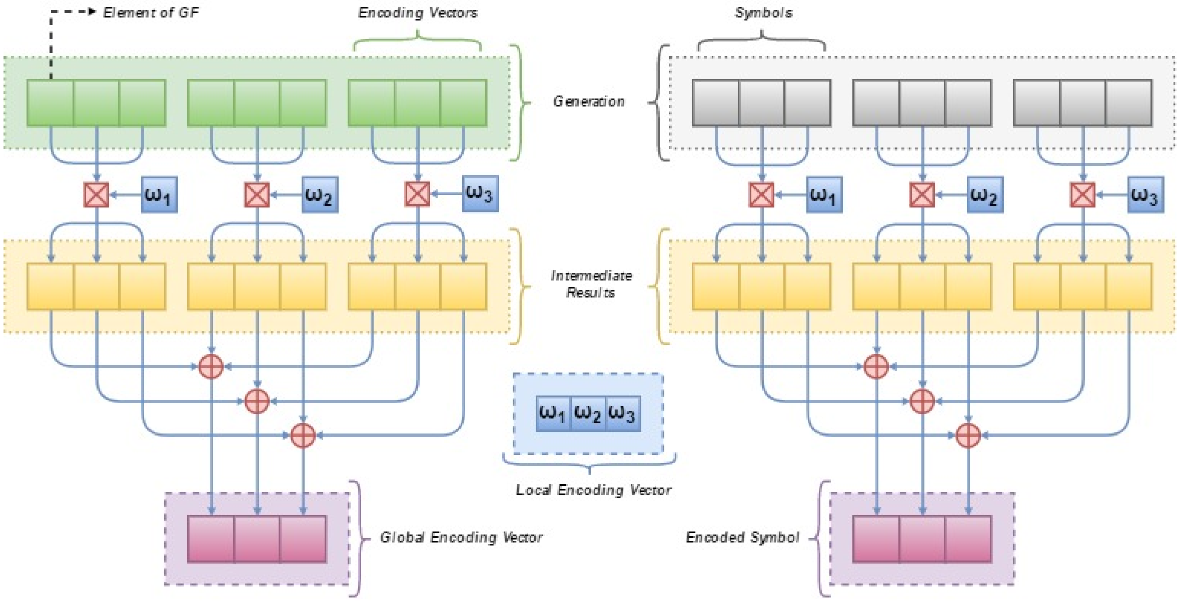


Figure 2.12. The encoding process of Random Linear Network Coding (RLNC) [66].

coded symbols. Recoded symbols and coefficient vectors have the same size as before. This operation usually is performed in an intra-session or inter-session according to whether it only combines packets belonging to the same session/flow or not [67]. This ability increases the likelihood of innovative transmission to the receiver as intermediate nodes can encode packets faster. The higher the probability of transmitting packets containing new information, the faster the decode process progress is. Moreover, in contrast with LNC techniques, in RLNC the local encoding vectors do not need to be computed apriori in all network nodes using the same algorithm (e.g., polynomial-time algorithms). Instead, these vectors can be assigned in a decentralized way [68].

Nonetheless, solely block-based coding methods such as RLNC can only achieve high data rates at the expense of high in-order delays, which makes them unsuitable for the high-speed, low-latency applications that the mmWave is envisioned for. Another drawback of this operation is that it does not support live stream traffic where the data might not be fully available from the start to create coded packets [67]. Note that by in order-delivery delay, we mean the time that passes between the first time an information packet is transmitted in a coded packet and the time the sender successfully decodes it [69].

Error control protocols like SR-ARQ, which use the feedback received from the sender (ACK and NACK) to perform TCP packet retransmissions, have been contemplated by the research community to reduce in-order delivery [69]. However, while SR-ARQ was shown to achieve near-optimal in-order packet delivery, it still struggles in channels with high RTT fluctuations where the feedback might be delayed [70]. Additionally, packet-level FEC mechanisms have been proposed to introduce redundancy alongside the feedback acknowledgments to further improve the performance of SR-ARQ in error-prone networks. These approaches are casual (i.e., use feedback acknowledgments), but they still cannot keep up with the changes in variable channels (i.e., non-adaptive). When employing non-adaptive approaches, it is not possible to adjust the number of DoF to fit the current erasure probability, possibly leading to more or less redundancy being added than what is needed. Hence, reducing the amount of useful throughput.

To address the aforementioned problems, MIT's researchers have proposed in 2019 a novel Adaptive Causal Network Coding with Feedback (AC-RLNC) for point-to-point communication channels which optimizes both the achievable throughput and in-order delay [69]. This technique exploits Forward Error Correction (FEC) at the packet level to ensure the required data rates, and the sender can track the erasure pattern through the feedback it receives from the other station to ensure low delays. The latter allows AC-RLNC to dynamically adjust the code rate to the specific channel conditions, thus being suitable for real-time transmissions under variable propagation channels such as mmWave ones. Their results showed that, for the point-to-point communication channel in the non-asymptotic regime, the proposed code might achieve more than 90% of the channel capacity. Furthermore, it was demonstrated, through numerical simulations and experimentally obtained commercial traces for IEEE 802.11n/ax/ac devices, that AC-RLNC could double the throughput gains and triple the gain in terms of in-order delay in a bursty channel when compared to SR-ARQ [69]. This is due to the sender's ability to track the rate of the channel through feedback received and the ability to adapt the apriori degree of freedom used during the encoding process according to the estimated missing degree of freedom. Retransmissions are also performed according to a criterion that is defined using the measured channel rate r, apriori DoF m_a and missing DoF m_d is superior to a given threshold th, as shown by equation 2.10. Thus, it is possible to determine the amount of posteriori DoF that should be sent to compensate variations. This approach was later generalized for multi-path multi-hop networks in [71].

$$r - d > th = r - \frac{m_d}{a_d} \tag{2.10}$$

The introduction of NC techniques at the network and transport layers loosens the requirements of the Forward Error Correction (FEC) mechanism implemented at the physical level. This is especially useful in mmWave networks where obstruction increases the delay, packet losses, and hence the number of normalized TCP retransmissions. As proven by author's of [72], for a ns-3 simulated mmWave cellular network, frequent switching between LOS an NLOS conditions results in numerous retransmissions by the TCP's congestion control mechanism which effectively decreases its performance and increases delay. This observation was further extended to IEEE 802.11ad-based networks in [73]. In that work, it was found that TCP does not account for the round-trip timeout changes that occur whenever any of WiGig resilience mechanisms (such as beamforming training and beam tracking) are triggered under normal and obstructed conditions. Figure 2.13 of [74] presents experimental results that show this phenomenon. It can be seen that the introduction of blockage at the 20-second mark causes link failure, which in turn leads to a decrease in the amount of supported throughput and triggers the beamforming training. During the beamforming training process, the milliliterwave link fails, and packets that were in the receive queue are left unprocessed, thus increasing the overall size of the receive queue. By the time the process ends, the packet's timeout has been reached, and the TCP treats the unprocessed packets as being lost. The detection of a timeout-based loss event makes the congestion control adjust its window, leading the throughput to be resumed to a value that is lower than its pre-loss one when the mmWave connection is reestablished. The interaction of NC and mmWave will be addressed in this thesis.

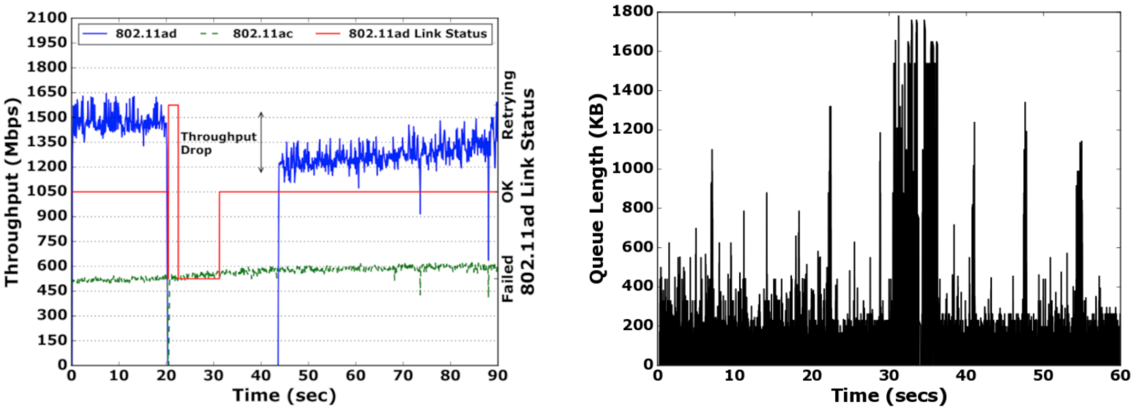


Figure 2.13. Throughput drop after mmWave link is obstructed (left) and the receiver queue size observed for during blockage (right) [74].

2.4.2 Software Defined Network

Software-Defined Networking (SDN) is an emerging new paradigm that decouples the management plane from the data plane (e.g., forwarding of packets) of network equipment such as switches and routers. Similar to legacy IP networks, the control plane of an SDN network performs routing, load balancing, enforces security policies, and supports end-to-end Quality of Service (QoS). With SDN, the network control becomes directly programmable, allowing administrators to monitor and manage network resources allocations in real-time to optimize performance [75]. A generic SDN architecture is presented in figure 2.14.

Control of an SDN network is orchestrated by logically centralized SDN controllers that maintain a global view of the network. These components contain the logic responsible for translating the application requirements down to the data plane, which allows the underlying infrastructure to be abstracted for applications and network services [76]. In this work, the SDN controller uses Open Network Operating System (ONOS) in the reactive forwarding mode, installing forwarding rules as new flows that are communicated by the SDN switches. This allows flexible adding and dropping of routing policies without any prior knowledge of the traffic [77], and thus the controller can readjust the routes to forward traffic in a specific communication path.

As shown in 2.14, the SDN controller has to communicate with the upper and down layers through a set of defined and standardized open interfaces. In the research community,

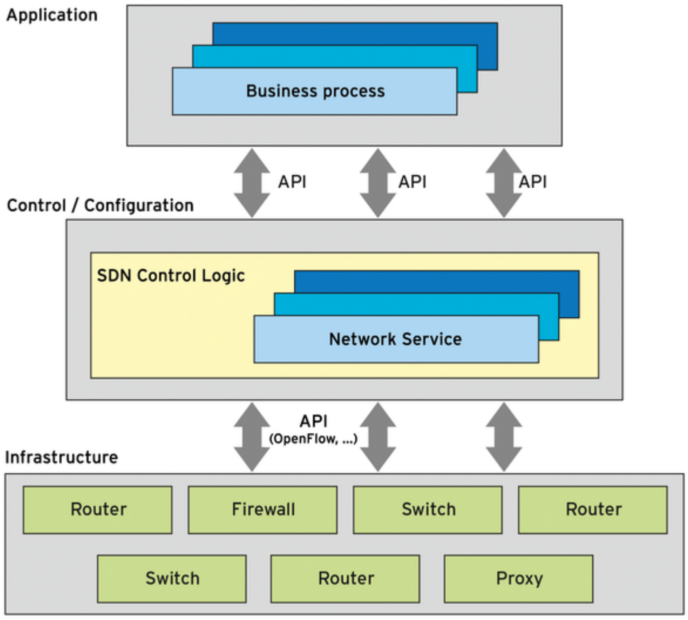


Figure 2.14. The SDN architecture [78]

the interface north of the control plane is denominated northbound API and is used by the controller to communicate with higher-level components. This interface allows the network applications to report to the controller their requirements and enables network operators to have direct control over the network. Popular northbound APIs include web interfaces and REST APIs. Conversely, the interface located south of the control plane is known as the southbound API, and its job is to establish communication between the SDN controller and the network devices (e.g., router and switches). This interface allows the SDN controller to manage and configure network switches' flow tables and dynamically adjust the underlying infrastructure to meet the real-time demands. Today the most popular southbound API connecting data and control plane is OpenFlow (OF) [79], which is also the interface used in our physical deployment. OpenFlow operates by adding entries in flow tables located in each network device, which define the actions to be taken for each flow, such as dropping packets and forwarding them to a specific port.

Despite the advantages, SDN is not fully deployed for several reasons. Firstly, some of the networking services for legacy networks lack support in SDN. Secondly, replacing every legacy switch in the current infrastructures with an OpenFlow-compatible device is infeasible owing to its high-cost [80]. To address these issues, hybrid SDN networks are being proposed where a limited number of SDN-enabled devices (which support both legacy and OF protocols) and dedicated SDN switches are used alongside legacy equipment [80] [81]. Even so, hybrid SDN-based approaches still have their limitations. OF messages are exchanged between the SDN switches and controller in a full SDN deployment to report the number of interfaces, associated capacities, and connection states of SDN devices. Furthermore, to discover the overall network topology and keep track of its changes, the controller periodically sends LLDP messages to neighbor SDN switches that are supposed to be received, processed, and later forwarded to other neighbor devices until they reach the controller once again. However, these messages cannot reach their final destination due to LLDP packets being dropped in legacy equipment, an issue that can only be solved by sending BDDP messages. BDDP functioning is similar to LLDP's with the distinction that packets are sent in broadcast, and thus can be forwarded by legacy equipment located between two OF switches' ports [80].

Still, similar to LLDP, BDDP can only be used by the SDN controller to gather information regarding OF devices as the controller is not aware of the individual status of each intermediate legacy link. This makes it difficult to detect specific failures in legacy links. Instead, the controller only detects link failures in the overall path formed between two OF switches. Besides, since OF messages do not have direct support for wireless hybrid-SDN networks [79], and thereby it is not even possible to know the capacity or state of wireless devices that are directly connected to OF switches. Support for this kind of operation is particularly challenging in highly variable mmWave networks that mainly rely on LOS conditions to operate. As such, the SDN controller is most likely to detect topology changes with a long delay, leading to temporary packet losses that compromise the QoS and QoE. A way to enhance the controller's ability to detect topology changes, and thus to increase the overall mmWave network reliability, is using estimations of future link qualities (made based on past observations) to reroute priority flows over alternative backhaul links before a severe link performance degradation happens. The estimations and predictions performed in this thesis will be used to self-optimize the SDN controller decisions.

2.5 Related Work in the Literature

This section presents innovation works in the literature on multiple fronts towards a resilient mmWave backhaul network. Aspects ranging from the characterization of the impact caused by long-term and short-term obstruction in an outdoor deployment, and the use of classification and prediction models plus network coding techniques to increase the reliability of mmWave networks are researched. For this reason, two main strands of action can be identified and are discussed below. In the first section, the findings and limitations of several experimental works on 60GHz networks are listed for indoor and outdoor scenarios and compared to the approach we will be fulfilling in this thesis. In the last section, a comprehensive study of existing techniques to increase the reliability of WiGig-based networks is presented alongside their shortcomings compared to the approaches that will be proposed in this thesis.

2.5.1 Work on Experimental 60 GHz Networks

Until this point, the experimental studies on the 60GHz band used wireless docking stations with radio chipsets that Wilocity typically manufactures. Moreover, they mainly focused on indoor measurements campaigns. In 2015, [82] investigated the feasibility of full-scale indoor 60

GHz networks by conducting several measurements using 802.11ad-compliant COTS devices in an indoor office environment. The authors conducted a multiple layer stack analysis using RSSI, PHY data rate, and TCP throughput to characterize the impact of the transmitter-receiver distance, antenna height, location, and orientation on the system's performance. Mainly, they found that RSSI is a weak indicator for PHY data and TCP throughput and that PHY data rate is not always a good indicator of higher-layer performance. Furthermore, they showed that static human blockage standing close to either the receiver or transmitter had a greater impact on the TCP throughput than a transient blockage. In that same year, authors of [83] presented the first in-depth beamforming, interference, and frame-level protocol analysis of COTS mmWave hardware in an indoor environment. Most interestingly, they found that interfering reflections from neighboring, unaligned devices may reduce the achievable TCP throughput by more than 20%. Later in 2018, Kacou et. al. [83] presented a 60GHz channel sounding experiment using a custom setup to compute the channel's RSM delay spread and associated path loss under LOS and NLOS conditions. Their practical measurements showed that their 802.11ad equipment has a 65-70 dB of propagation loss under normal operation. However, in the case of obstruction by a load bearing wall or by a wooden cabinet close to a plasterboard dividing wall, at least 25 dB of additional path loss was observed for the best non-direct path as it was not possible to establish communication. Additionally, they found that the TCP throughput decreases from values that vary between 1.6Gbps to 2.5Gbps in LOS conditions to values within the 0 to 1.6Gbps when faced with static obstruction from some common construction materials. Besides, it was observed that four people blocking a LOS path still enabled achieving a throughput greater than 1Gbps in indoor office environments for distances up to 10 meters.

In 2020, authors of [59] reported the ability to differentiate with 96% to 97% accuracy temporary from permanent blockage caused by humans in an indoor environment by analytically analyzing sudden drops in signal quality and TCP throughput. To do so, they deployed an experimental testbed that includes an INTEL WiGig sink chipset W13100 as the AP and the INTEL tri-band wireless-AC 17265 wireless cards as the wireless station (STA). Based on those observations, they proposed a handoff scheme to alternate between APs to mitigate the impact of obstruction. However, similar to the previous listed works, the blockage characterization is limited to a small set of metrics that include at best the TCP throughput, RSSI, and PHY data rate since other MAC and PHY metrics are not easily accessible in most commercial WiGig devices. To access these lower layer metrics, the authors of [84] proposed a scheme that extracts them from control files stored inside the "Wilocity Monitory" windows service. That work presented a preliminary study of the behavior lower layer (AGC attenuation, MCS, data rate, and CRC failure rate) metrics for a blocked and non-blocked indoor mobility scenarios that use a COTS 802.11ad device. More specifically, they describe the impact that transmit-receiver distance (up to 15 meters) has at multiple layers and define the influence of transient blockage at different speeds on the SNR and in the beam adaption process. Their results show that transient blockage may induce long-term suboptimal beam alignment on a 60GHz link, which causes a significant performance decrease. Later in 2019, [85] extended this work by providing a more in-depth analysis of static and transient blockage impact. His extensive analysis describes how different types of blockage affect the TCP throughput, the number of sent control packets, the MCS, and the beam pattern changes of an COTS device. Some relevant insights are provided in [85] . Specifically, the author showed that 1) static blockage often operates at higher MCSs than in the transient case as beam misalignment is more likely in mobility scenarios, 2) the rate at which beam tracking methods are triggered cannot keep up with pedestrian movement and 3) poor performance of the connection under long-term obstruction is due to limited beam-tracking capabilities and inefficient beam-sweep. [84] and later [85], are huge advances, as, until that point, no one had presented such a detailed analysis of blockage-prone WiGig networks at both a PHY and MAC level. Still, it has some limitations from a higher-layer perspective. Because it only focuses on TCP throughput, the performance of critical mmWave indicators such as delay and packet loss are excluded from his analysis. Furthermore, the measurements campaign was conducted indoors with equipment whose characteristics are completely different from those intended for outdoors. Note that most commercially available WiGig devices are designed for indoors to cover a maximum distance of 15 meters, even after beam steering.

Nevertheless, a few studies have reported to evaluate the feasibility of typically indoor' IEEE 802.11ad solutions on outdoor environments suffering from static and transient blockage. To the best of this author's knowledge, only [86] and most recently [87] address outdoor scenarios. Zhu et al. presented in [86] the results of measurements using COTS (Wilocity) 60GHz radios and a horn antenna emulating a 10x10 antenna array to evaluate the feasibility of 60GHz picocells in outdoor scenarios. The work explores the range, attenuation, and interference in urban scenarios and dispels some common myths about 60GHz communications. Their measurements show that employing large-enough phased arrays enables covering distances up to 79m while maintaining a 2+ Gbps data rate. Moreover, they showed that it is possible to circumvent blockage and establish robust outdoor connectivity by electronically steering the phased-array antennas to a reflecting-formed path, with a loss of only 5-7 dB. Also, they found that antennas on COTS radios can maintain high data rates, even under user motion, only needing to realign once every few seconds. This work described for the first time from an experimental point of view the potential of WiGig in outdoor environments, though it still has two major limitations. Firstly, the results of simulating a large array using a horn antenna can only be considered valid in an environment with no interference. Secondly, the cross-layer performance analysis once again only includes RSSI and TCP throughput. Most recently, Tran et al. [87] conducted an outdoor experiment campaign in a customized mmWave Meshed Backhaul that employs WiGig to demonstrate the potential of Mobile Edge Computing (MEC) technology in this type of networks. The tests performed on this testbed focus on evaluating the UDP throughput and end-to-end delay achieved in non-obstructed single-hop and multi-hop backhaul links with a fixed MCS of 9. It is also worth mentioning that so far, this is the only work that proposes a physical deployment for an SDN-controlled mmWave backhaul.

Table 2.2. Comparison of existing experimental measurement work for WiGig-based networks.

Work	Date	DUTs	Environment	Metrics	Scenarios Investigated	Limitations
[82]	2015	Dell Latitude E420 laptop with wil6210 (STA) Dell Wireless Dock D5000 (AP)	Indoor	TCP throughput	Normal Transient human blockage Static human blockage	Lacks extensive PHY and MAC layer cross-analysis
[83]	2018	Qualcomm 802.11ad mini PCIe cards	Indoor	TCP throughput	Normal Static obstruction from common construction materials Transient human blockage	
[59]	2020	INTEL tri-band wireless-AC 17265 wireless cards (STA) INTEL WiGig sink chipset W13100 (AP)	Indoor	TCP throughput RSSI	Normal Transient human blockage	
[84]	2016	D5000 wireless docking station (AP) Dell E7440 laptop (STA)	Indoor	AGC attenuation MCS, data rate, SNR, CRC failure rate and beam adaptation	Normal Transient blockage at different speeds	
[85]	2019	D5000 wireless docking station (AP) Dell E7440 laptop (STA)	Indoor	TCP throughput No. of sent control packets, MCS Beam pattern	Normal Static Blockage Transient blockage at different speeds	
[86]	2014	D5000 wireless docking station (AP) Dell 6430U laptop (STA)	Outdoor	RSS TCP throughput	Normal Static blockage from common construction materials Transient human blockage	Lacks extensive PHY and MAC layer cross-analysis Uses a 60 GHz horn antenna to simulate a 10x10 phased-array
[87]	2019	Custom setup based on Panasonic radio transceivers	Outdoor	Latency	Normal	Lacks extensive PHY and MAC layer cross-analysis Does not study the impact of transient and human blockage
This work	2021	CCS Metnet 60G nodes	Outdoor	RSSI, SNR PER, MCS Beam Index TX Beam Index RX L2 data rate RTT, PL TCP throughput TCP retransmissions	Normal Static blockage from metal object Transient human obstruction	Does not include UDP metrics

Table 2.2 provides a brief summary of the aspects discussed for the existing experimental work in the 60GHz band.

This thesis will present a more extensive PHY, MAC and transport layer measurement campaign for either indoor and outdoor WiGig networks. Furthermore, it will join the analysis of metrics from the transport (TCP throughput, number of TCP retransmissions, packet loss rate, RTT) and PHY/MAC layers (RCPI, SNR, Packet Error Rate, Beam Index TX, Beam Index RX and L2 data rate) to characterize static and transient blockage on 60 GHz links. More specifically, this work proposes a cross-layer analysis of the impact caused by long-term and short-term blockage on a real WiGig-based outdoor network intended for 5G backhauling.

2.5.2 Resilience Mechanisms

The resilience mechanisms implemented in IEEE 802.11ad only allow actions to be taken after a significant link quality decrease is detected, and after the underlying processes (such as beamforming training) have finished, which in the case of the beam alignment process has a high associated overhead [88] [89]. Because these methods cannot provide the necessary robustness to critical applications, several proactive and reactive resilience mechanisms are being reported to eliminate the need for constant beamforming training. The current research mainly focuses on beam recovery techniques that minimize the overhead associated with the beamforming training procedure. These approaches propose increasing mmWave reliability by either improving the beam scanning process [90], [91], [92], [93] and [94] or by predicting alternative beams in advance [95] and based on past records. For instance, in 2016, authors of [90] designed and evaluated MOCA, a protocol for Mobility Resilience and Overhead Constrained Adaptation for directional 60 GHz links. Under this protocol, several reactive techniques are used to restore directional links with low overhead. Firstly, the authors devised a Beam Sounding mechanism to estimate the beam quality for a few selected beams by exchanging control frames prior to the actual data transmission. Their implementation involves searching for alternate sectors in advance by modifying the beam refinement procedure to assess the SNR of multiple sector pairs rather than just the strongest sector pair. Their results show that the proposed scheme can reduce packet loss ratio and increase throughput in several conditions in a testbed using commercial 60 GHz transceivers. On that same note, in 2018, Shao et al. [91] proposed a new beamforming training scheme for IEEE 802.11ad-based networks that exploits the AoA sparsity of mmWave channels to enable group beam training of all user devices attempting to connect with an access point. The proposed approach, which was developed using a 60GHZ testbed and validated on a simulator, showed that compressed sensing reduces the cost of sector sweep by 50%. Later in 2021, the work presented on [95] proposed a fast beam restoration scheme, that uses past beamforming combinations observed during blockage as input to a Long-Short-Term Memory (LSTM). The LSTM can predict the best alternative beam directions for the next time frame, without depending on out-of-band information. The proposed scheme was observed to maintain communication sessions, reducing significantly the number of times the training procedure is triggered.

Another class of resilience mechanisms that has been described recently for mmWave networks is the methods that rely on out-of-band information (provided by sub-6GHz bands) to predict future obstruction, such as [96] and [97] . In 2019, Ali et al. [96] exploited the fact that the Received Signal Strength Indicator (RSSI) of sub-6GHz communication crosses a defined threshold much earlier than mmWave signals in obstructed links to propose a scheme to detect early signs of blockage. The simulations performed using a Fresnel-diffraction-based channel simulation tool showed that sub-6GHz signals can be used to predict the sudden changes in mmWave channels tens of ms ahead. Also, they demonstrated the possibility of predicting the strongest multipath component and its angle of arrival using the sub-6GHz observations. As for [97], the authors proposed a deep-learning neural network that used 3D ray-tracing software to develop a dataset that can be used to extract temporal information from sub-6GHz channels that enable predicting blockage and optimal beamforming combinations. Specifically, they show that a mapping function can directly predict the optimal mmWave beam (out of a codebook) from the sub-6GHz channel measurements under certain specific conditions. However, because these approaches use out-of-band information, they do not support standalone operational modes of mmwave bands.

To address this, some works [98] [99] are being developed to classify wireless link quality using in-band mmWave measurements. In [98], authors leverage RSSI variations to detect transient human blockage and link movement (between RX and TX) in an indoor environment. The proposed scheme classifies if the link is either short-term blocked or suffering from relative STAs movement by analyzing if the variance measured within a given time window falls below a defined threshold. According to their experiments, a 100% classification accuracy is possible to achieve after a 30ms time window. Besides the 30ms delay, this approach has two major limitations. Firstly, the dataset is collected with a customized solution that uses a steerable 34 dBi horn antenna rather than electronically steerable antenna arrays. In addition, because it is a mechanism solely based on RSSI measurements made in an indoor environment with short distances, it is bound to fail in outdoor channels subjected to rapid fluctuations.

Later in 2021, the work in [99] proposed a new learning-based prediction framework that leverages the measured receive and transmitted signal patterns to detect whether a mmWave link is blocked or not. Their observations showed that more irregular patterns are observed (with new side lobes and spikes) whenever a link suffers from obstruction due to signal diffraction and reflection in the obstacles. The dataset in this work was collected in an indoor testbed that uses two commercial TP-Link TALON AD7200 routers. To generate the dataset, they used the approach described in [100] which allows extracting the phase and magnitude of the signal patterns from coarse signal strength readings using a beamforming training process (with a custom codebook to probe different directions). The experimental results showed that the proposed framework can distinguish between blocked and non-blocked scenarios with an accuracy of about 90% for the three ML models (ensemble, K-nearest neighbors, and Support Vector Machine (SVM)). Nonetheless, because the approach relies on sequential probing of different directions, it is unsuitable for real-life applications. Also, the nature of the classification algorithm does not allow collecting data to identify the transient obstruction. Thus, the method is only able to recognize long-term blocked patterns.

In contrast, this thesis proposes a ML-based approach that leverages multiple layer metrics to classify if a mmWave link is in a non-blocked, long-term blocked or short-term blocked state. To best of this author's knowledge, this is the first approach that proposes using metrics beyond RSSI to identify the state of a WiGig network as until this point, access to other KPIs was limited by manufactures of IEEE 802.11ad COTS devices.

Even so, predicting the current link quality is not enough to ensure a reliable link quality. The only way to have a truly proactive mechanism is by predicting future obstructions before their impact on the network is experienced. This requires predicting in advance the expected value of main link KPIs such as SNR, PER, PL, throughput and RTT. Still, up to this point, only [101] has reported to forecast metrics in mmWave, specifically operating using the IEEE 802.11ad. In [101] the predictability of the throughput in a WLAN IEEE 802.11ad network was evaluated for real-time applications, in downloading data to an IEEE 802.11ad mobile phone under varying mobility patterns and rotations. For that purpose, the authors proposed using a small RNN that exploits temporal relationships between the MCS, RSSI, link status indicator, TCP throughput, and beamforming tx and RX sectors, and pose of the user's equipment (pitch and azimuth) to predict TCP throughput for different time scales. Their results show that TCP throughput prediction when the mobile phone is static can be performed with high accuracy for the 40 to $2000ms$ windows, with a 95th percentile error of 10.6% for the 40ms and 5.7% for $2s$. Under random mobility scenarios, the 95th percentile error increased to 38.1% and 19.2% for the $10ms$ and $2s$ timescales, respectively. For the 10ms timescale, that corresponds to an error above 10% for 40% of the time.

In this thesis, the forecasting is taken a step further by proposing a DL-based framework that can predict multiple future timesteps, even under heavy obstruction, of an extensive set of link KPIs which includes Signal to Noise Ratio (SNR), Round-trip Time (RTT), Packet Error Rate (PER), Packet Loss (PL), TCP throughput, RX and TX beam, among others.

Another alternative to increase the reliability of mmWave backhauls is to employ Network Coding (NC) techniques. This approach, however, has not yet been widely explored by the scientific community. To the best of this author's knowledge, network coding techniques were only proposed to improve the performance of mmWave systems in two instances [102] and [103]. Firstly, in 2015 Motorola Mobility Limited's researchers [102] introduced a multi-path approach where devices transmit and received RLNC encoded packets through several access points. Using a Monte Carlo system simulation, the authors demonstrated that for downlink, intra-session network coding provided significant improvements efficiency-wise for individual mmWave channels suffering from erasure due to link outage and non-line-of-sight conditions. For instance, they showed, that using network coding over a network composed of relay nodes deployed every 30 meters improved the data rate by 35%, thus achieving shorter transmission times when compared with the simply forwarding approach.

The second instance was reported in 2018 by Drago et al. [103], which defined the first reliable video-streaming architecture for mmWave networks employing multi-connectivity between Long Term Evolution (LTE) and mmWave Radio Access Networks (RANs) using network coding techniques. The authors use RLNC to simplify the management of the transmission on multiple links and provide additional robustness. The performance of the proposed architecture was evaluated using ns-3 mmWave module, real-video traces, and the network coding library Kodo [104].

To this day, the potential of network coding has only been explored for mmWave systems through simulation-focused approaches. Furthermore, network coding techniques have not yet been extended for IEEE 802.11ad-based wireless networks. Much less has an analysis been made of the gains it may have over a traditional network approach, either at a simulation or physical level. This thesis will present preliminary results that demonstrate that the introduction of fixed-size block RLNC techniques can increase the robustness of a deployed WiGig backhaul suffering from severe obstruction.

Table 2.3 provides a brief summary of the aspects discussed for the current work on the resilience of WiGig-based networks.

Table 2.3. Comparison of existing work on mechanisms for improving resilience of WiGig-based networks.

Work	**Date**	**Resilience Technique Type**	**Proposed**	**Validation Framework**	**Limitations**
[90]	2016	Improving beamforming scanning process	Beam sounding mechanism to estimate the link quality for selected beams Preemptive Fast Recovery mechanism from link outage Dynamic adjustment of beamwidth according to channel variations	Custom testbed using SDR with 60 GHz transceivers Custom MATLAB simulator that uses trace-based channel measurements of the custom setup	Testbed uses different horn antennas that do not have beamsteering capabilities Reactive mechanism
[91]	2018	Improving beamforming scanning process	Group beamforming training mechanism leveraging AoA sparsity and compressed sensing	SNR traces in the˜X60 SDR-based testbed	Reactive mechanism
[95]	2021	Improving beamforming scanning process	Fast beam restoration scheme that uses a DL model to predict an alternative beam	Simulated analytical channel and blockage model	Reactive mechanism Validated only through theoretical simulations
[96]	2019	Out-of-band information	Mechanism that uses early signs of blockage in the sub-6 GHz band to identify possible obstructions in the mmWave band	Fresnel-diffraction based channel simulation tool	Validated only through theoretical simulations
[97]	2020	Out-of-band information	Neural network that uses 3D ray-tracing software to predict the optimal mmWave beam and blockage status directly from the sub-6 GHz channel	Publicly-available dataset created using a 3D ray-tracing simulator	Does not support standalone operational modes of mmWave bands
[105]	2015	Wireless link quality classifier	Analytical mechanism that uses RSSI variations to detect transient blockage	Custom testbed that uses WARP FPGA, a 60GHz development board, and 34 dBi horn antenna	30 ms of induced delay to achieve 100% accuracy Does not use electronically steerable antennas Only suitable for indoors
[99]	2021	Wireless link quality classifier	DL-based model that uses the measured receive and transmitted patterns to detect obstruction	Tesbed using TP-Link TALON AD7200 routers	Not suitable for critical scenarios because it relies on sequentially probing different directions Only suitable for indoors Cannot identify the transient obstruction
[101]	2021	KPIs forecasting	DL-based model to predict TCP throughput in a mmWave indoor channel	Testbed consisting of an ASUS ROG Phone II and Netgear Nighthawk X10 WiFi router	Can only predict TCP throughput in the case of STA mobility and not under obstruction
[102]	2015	Network coding	RLNC employed to provide additional robustness to a mmWave backhaul	Monte Carlo System Simulation	Evaluated only through simulations
[103]	2018	Network coding	RLNC employed to provide additional robustness in a network employing multi-connectivity between lte and mmWave RAN	ns-3 mmWave module with Kodo	Not extended to WiGig-based networks
This work	2021	Wireless link quality classifier KPIs forecasting Network coding	ML-based approach that uses multi-layer KPIs to detect long-term and short-term obstruction DL-based forecaster to predict future multi-layer KPIs under variable channel conditions Mechanism that employs fixed-size block RLNC techniques to improve robustness in error-prone networks	Experimental outdoor testbed using CCS Metnet 60G nodes	Does not use UDP metrics

2.6 Summary

This chapter introduced the background concepts that supported the work developed towards a resilient mmWave backhaul. We first discuss the advantages and disadvantages of radiation propagation in the 60GHz band and describe techniques used at the physical and MAC level to overcome those limitations in WiGig standards operating in the same band. More specifically, it was found that, although mmWave propagation allows achieving multi-gigabit data rates with low latency, the high frequencies used results in higher propagation losses, which can only be overcome by establishing highly directional links. For this reason, the WiGig-based WLANs deployed in our infrastructure, such as IEEE 802.11ad/ay, introduce new physical and MAC layers that include support for beamforming training of phased-array antennas that allow the establishment of highly directional links between two devices.

Afterwards, a WiGig-based 5G wireless backhaul architecture was described, and potential problems it may experience were investigated. Specifically, it was argued that the directional nature of mmWave communication leads to increased sensitivity to obstructions that can compromise the reliability of the network. Moreover, it was shown that the proprietary IEEE 802.11ad mechanisms are not sufficient to maintain a reliable connection as they have a high overhead associated. Furthermore, these mechanisms are reactive, and re-training is only triggered when performance degradation is already detected, which ends up compromising QoS and QoE. Later, the concepts associated with the multi-layer resilience mechanisms based on SDN-controlled networks, network coding, and machine learning for wireless link quality classification and metric forecasting were presented.

This chapter ended by presenting a comprehensive study of existing experimental work on 60GHz networks where metrics measurements are made in various scenarios. It was found that this thesis presents the first extensive cross-analysis of multi-layer metrics in an IEEE 802.11ad network subjected to long-term and short-term blockage. Moreover, we present a study of the mechanisms proposed in the literature to increase the reliability of WiGig-based networks. Our analysis has shown that the methods proposed so far present one or more set of limitations: they are reactive, allowing actions to be taken only after quality degradation is detected; they are usually focused on improving particular processes, usually related to the beamforming process; moreover, those that do not focus on beamforming training are validated in simulation environments; those that are validated in a real environment are in indoor environments. In contrast, the work presented in this thesis, and described in the following chapters, proposes proactive mechanisms for SDN-controlled WiGig-based networks that leverage the estimated wireless link quality and relevant future KPIs to act at multiple layers; and network coding techniques to improve the robustness in error-prone channels.

Chapter 3

Real Deployment

This chapter details aspects related to the mmWave deployment, and is organized in two parts that represent the first development cycle of the SNOB5G project. The first part describes the functionalities and limitations of the equipment used, as well as the architecture used to deploy the infrastructure in an outdoor environment. The second part depicts the methodology employed to collect multi-layer metrics on different mmWave links under three different blockage scenarios. The following sections describe a general overview of the deployed network, as well as a functional description of the network operation and the available service features.

3.1 Backhaul Infrastructure

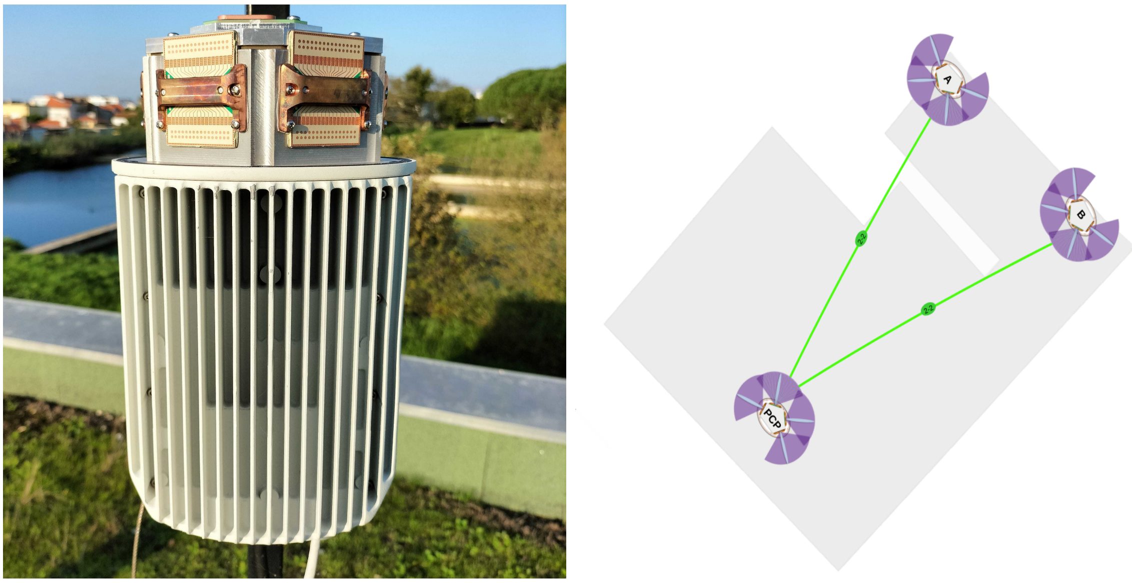
This section describes the backhaul infrastructure, its functionalities and architecture.

3.1.1 Network Overview

The backhaul mmWave network used in this work adopts a more general architecture than the distributed approach described in [41]. In this architecture, multiple Small cell Base Stations (SBSs) can be connected to the core network via fiber connections at Point of Presence (PoP) locations.

The mmWave network used throughout this work is composed of three CCS Metnet 60GHz nodes [47], similar to the ones presented in Figure 3.1a. The system was developed for deployment on outdoor environments, such as street furniture and lampposts. However, in this case, they were installed on the rooftop of Aveiro's IT branch on a support placed at the height of about 1.5 meters. This ensures that the work is developed under a fully closed and controlled outdoor environment. The deployed testbed is presented in figure 3.1b.

Each node has four radios covering a 90º range that overlap to cover a 300º horizontal



(a) CCS Metnet node and its four radios.

(b) Testbed used throughout this work.

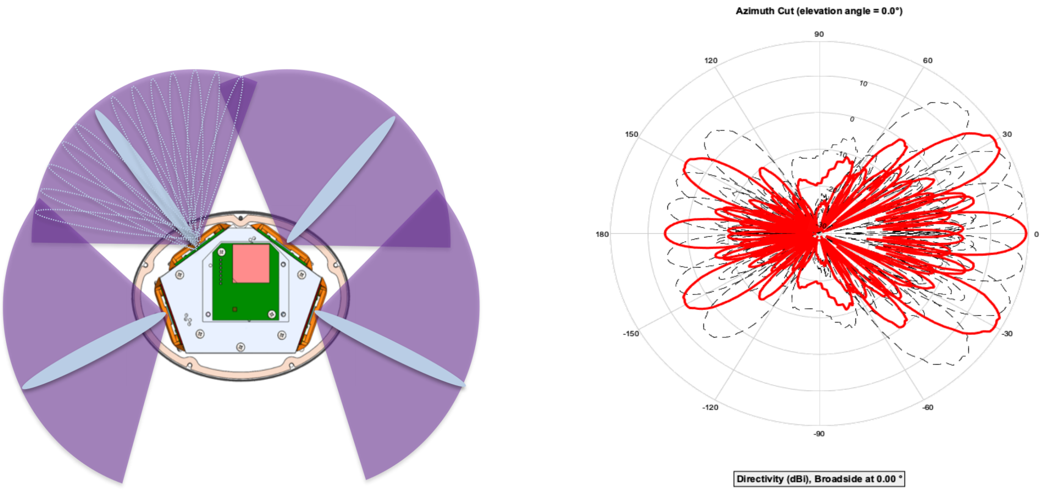
Figure 3.1. Nodes and topology of the deployed outdoor testbed.

field of view, as shown in figure 3.2a. The current v1 mesh nodes leverage the standardized 802.11ad (WiGig) technology, described in Chapter 2 in Section 2.1.2, to form a wireless 5G meshed backhaul. According to the vendor, this network is able to form Self-organising Network (SON) and self-healing links between neighbor nodes that dynamically adapt to the specific channel conditions. However, it was observed that this feature is still very limited.

The system operates between the 57GHz and 66 GHz unlicensed band which supports channels 1 to 4, each with a bandwidth of 2.16 GHz. The employment of a 19 dBi beamforming steerable antenna in each radio module enables the establishment of directional links between STAs, which ultimately allows to cope with the 60GHz channel propagation loss, reducing interference due to spatial reuse. The 300º wide range covered by each node is further divided into a set of 64 discrete sectors (with 5º horizontal beamwidth), that can be used to concentrate the transmitted signal in the receiver's direction. Each sector is implemented using precomputed antenna weight vectors for the four modules' phased arrays. As shown in figure 3.2, the use of phased-arrays antennas with beamforming capabilities allows the sender node to concentrate the signal in the best direction it has found towards the receiver (see the training process in Chapter 2, Subsection 2.1.2).

While the standardized IEEE 802.11ad defines a point to multi-point network architecture that requires LOS - which typically covers distances up to 300-500m with phased-arrays antennas -, the used system takes this one step further, by engineering a multi-point to multi-point topology. This equipment can cover distances up to 500m, depending on the location, availability and available capacity. Additionally, the system implements a proprietary interference avoidance technology that allows coexistence with other 60GHz devices [47].

The system's SON allows nodes in close vincity to operate with negligible co-channel


Azimuth Cut (elevation angle = 0.0°)
90
120
60
150
30
180
0
-150
-30
-120
-60
-90
10
0
-10
Directivity (dBi), Broadside at 0.00 °

(a) CCS node's multisector beamforming phased arrays [106].

(b) Radiation pattern used by a radio's node to establish 3 different directional beams.

Figure 3.2. Example of establishing directional links with the phased-array beamforming process.

and adjacent channel interference, thanks to its interference monitoring mechanism, dynamic scheduling and automatic routing. Moreover, it enables automatic configuration of radios, removing the need for manual configuration and complex frequency planning, enabling mobile operators and broadband service providers to rapidly deploy gigabit connectivity at a relatively low cost. This solution can also be described as scalable, since the introduction of new nodes is automatically detected, causing the network to automatically reorganize itself. However, the reorganization capability is still limited, since it is part of a feature that is still under development. Additional information regarding the system's characteristics and features is provided in table 3.1.

In this equipment, the employment of a Time Division Multiple Access (TDMA) and dynamic Time Division Duplex (TDD) alongside the high spectral reuse at mmWave frequencies allows the usage of both space and time switching in the four 802.11ad channels. Optimizing the spectrum utilization while also minimizing the interference eliminates the need for complex frequency planning. Using TDD, the node is able to support asymmetric traffic and time-varying uplink and downlink demands, which are commonly observed in real scenarios.

3.1.2 Functional Description

The network comprises several 60GHz mesh nodes that can be either completely wireless or have a wired connection to the core network. A node acts as a wired node if it is connected to a wired Gigabit Ethernet connections, or one of its 1G/10G optical ports. If the node is not connected to the PoP, it acts as a remote node, and access to the core network is given through one of the wired nodes. The CCS system also features an EMS, based on HTTPS and SNMP protocol, which provides a graphical overview of the state of the nodes and the

Table 3.1. Metnet mesh version 1 node specifications [47].

Frequency Band	60GHz mmWave unlicensed 60GHz mmWave unlicensed 57GHz to 66GHz band
Topologies	Point to MultiPoint MultiPoint to MultiPoint (Mesh)
Capacity	10 Gbps per node
Radio Access	Metnet SON utilizing TDMA Dynamic TDD
Antenna	Beamforming Phased array 19 dBi gain
Beam angle	Horizontal 300° electronic steerable 5° beamwidth, Vertical 20° fixed
Modulation and encoding	10 levels of adaptive encoding MCS 0–9 up to 2.5 Gbps
Range	Up to 500m* Dependant on location, availability, and capacity
Interfaces	Up to 4 Ethernet interfaces 2 x Fixed RJ45 100/1000 Base-T 2 x optional 10G or 1G SFP optical
Ethernet Service features	Native Ethernet 802.1Q (VLAN Tagging) 802.1p QoS (Class of service) Differentiated Services Code Point QoS (DSCP) 802.1ad QinQ

mmWave links. Within EMS, the network is organized in clusters. In the current version, every wired node takes part in a process to elect a cluster manager. The EMS Cluster Manager (CM) uses the radio scans collected from nodes to optimize the network topology, as it can perform both link and frequency planning. Hence, when a new node enters the network, the CM is able to configure its radios and links using its whole-network view. If needed, the CM can also re-plan the network topology, which may include changing frequency channels, or the radios for a new set of transmitting/receiving STAs. Still, our tests showed that dynamically changing links previously formed to use other neighbor nodes is not yet possible.

When a wired node is initialized, one L2TP tunnel is established between the EMS and the wired node if the EMS (which is deployed on a docker container inside a virtual machine) is accessible over the external IP network. The EMS then uses this tunnel to assign it an IP address within its private management network. In contrast, remote nodes do not have a direct connection to EMS. Rather, they depend on the radio link discovery of IEEE 802.11ad to establish the first connection to the EMS using a wired node as their gateway.

At the start-up, each remote node enters a discovery state during which it performs radio link discovery, to find any possible node neighbor, in particular PCPs. The radio link discovery relies on the beamforming process standardized in 802.11ad to form directional links between STAs. In this state, the STA assumes the role of the responder in the beamforming process with the PCP. If the station has found more than one wired connection, it will choose to be associated with the PCP with the best link quality and later transits to a Station mode. If no viable candidate is found, the remote node remains in discovery mode, until it finds a connection to a wired node. After the link discovery and association is complete, an L2TP tunnel is also established between the remote node and the EMS - through the wired node -, allowing the EMS to assign a private IP within the private management network. Once it joins the management plane (mplane), the node starts to report the discovered information to the cluster manager, allowing it to be configured by the CM. As expected, the mplane traffic reaches the wired nodes via the mmWave links established and is forwarded to the EMS using the L2TP tunnel.

Note that there is a deamon process implemented on each node responsible for setting up and maintaining the management network, and managing the tunneling in the wired nodes. By default, the L2TP tunnels between wired nodes and EMS are not encrypted. However, there is also the possibility of using IPSEC encryption if necessary.

As previously mentioned, in the firmware version deployed in the testbed (5.1.5 version), the SON still has its limitations, since it is not possible to dynamically adjust the network topology according to the link quality variations. The only type of topology reconfiguration supported happens at the IEEE 802.11ad level, upon link failure or whenever a new node is added to the network. When the transmitted signal is completely blocked, the standard ensures that the link is reestablished, by either finding an alternative beam to communicate, or by finding alternative routes through neighbor nodes.

Nonetheless, recent progress has been made in this direction. Version 5.8.0, released on June's first of 2021, includes a continuous beam measurement mechanism that works continuously in the background, to collect data from network pairs of radio antennas. This feature equips the cluster manager with a comprehensive set of data relating to all possible links, that it can use in the future to build a near-optimal topology, which ultimately could be used to adjust poorly performing links dynamically. Thus, it is expected that the vendor's next software releases will leverage this beam measurement information to improve the SON algorithm. Note, however, that this adjustment is likely to be a reactive mechanism, where decisions can only be made after a significant reduction in the link performance is detected.

Lastly, in terms of Ethernet service features, the Metnet node supports 802.11Q VLAN, allowing the Ethernet ports to be assigned as a tagged member of multiple VLANs, or an untagged member of a single VLAN. Each Metnet node may select from eight priority queues numbered from 0 (low priority) to 7 (high priority). The traffic can be mapped to the different priority queues by either the IP Differentiated Services Code Point (DSCP) value or VLAN 802.11p tag. With both 802.1p QoS and IP DSCP, traffic is prioritized according to the class it belongs to rather than what device and port it is being sent to, enabling scalable networking. Additionally, each node supports 802.1ad (QinQ). In QinQ, also known as IEEE 802.1ad or stacked VLANs, the traffic is double tagged by encapsulating the VLAN tagged traffic with another 802.1Q tag, the service provider tag (S-tag). QinQ enables a service provider (SP) to segregate different customers' traffic in its infrastructure while still giving the customer a full range of VLANs for its internal use. The buffer Size is also adjustable. The support for QoS could be exploited in the future by the SDN controller to ensure that priority traffic requirements are met.

3.1.3 Backhaul Architecture

As previously mentioned, the system uses the EMS to perform the tasks of node configuration, software upgrades, node statistics processing, displaying, and event handling. As shown in Figure 3.3, it has two main components:

- **EMS client/frontend:** the part of the system that the user interacts with. It includes the representation of the network, link performance statistics, node health status, network management and VLAN configuration. The information presented in the interface is generated by the server-side of the EMS;
- **EMS server/backend:** the backend includes an API that generates the data displayed in the frontend, a database to save the data, and the logic that implements the EMS functionalities. The backend is built using Python for the server and the Golang framework. The database used belongs to the document database category, more precisely to the NoSQL database family, as it uses a JSON-like format to store structured and unstructured documents. MongoDB scales both vertically or horizontally to accommodate large data loads [107].

In the context of this work, the EMS server is an API that runs on a docker container, hosted on a dedicated 64-bit Debian machine, and the EMS client is a standard web application that can run on any PC with access to the EMS server. Since the application is designed to run inside a docker container, installing and configuring docker inside a dedicated machine/server (known as the EMS host) was needed. The host specifications matched the minimum required specs to host a system, that manages a mmWave mesh network with up to 20 nodes, which is more than enough for testing purposes. After a successful EMS deployment, the nodes' firmware was flashed using the vendor's built-in web configurator. A commissioning laptop was connected to each mmWave node, via an Ethernet cable connected to the node's Ethernet port, for the mesh nodes provisioning. The product configurator web interface enabled the installation of the latest approved node image at the time, the 5.1.5 version. Using this

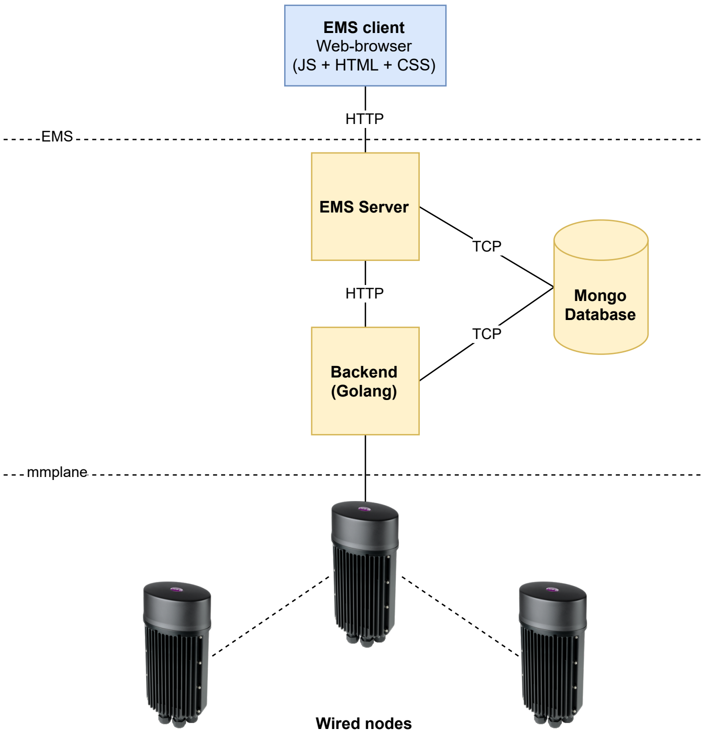


Figure 3.3. EMS architecture.

interface, it was possible to statically configure the EMS's server IP so that wired nodes could access it.

After both EMS system and nodes were correctly configured, each node was connected to one of two SDN switches, which are also connected to a switch inside the laboratory data center, as shown in figure 3.4. The SDN switches are controlled by a SDN controller - Open Network Operating System (ONOS) controller -, responsible for the management and control of the network. Thus, the logic implemented in this controller enabled the selection of a specific traffic route, by detecting a manual transition to the inactive state of the Ethernet port associated with the alternate path. For instance, to force traffic to flow between the PCP and the wired node B, the port 0 of the Software-Defined Networking (SDN) switch 2 (i.e., the port of the alternate path) was put in the inactive state, which caused the Software-Defined Networking (SDN) controller to detect this change and force traffic to flow in the desired route.

In order to inject the traffic within the mmWave mesh, two hosts were connected to both start and end nodes, as can be seen by the introduction of hosts 1 and 2 in the network diagram in figure 3.4. The Software-Defined Networking (SDN) switches then forward this traffic according to the flow tables (installed by the Software-Defined Networking (SDN) controller), in the desired communication path.

As previously mentioned, the wired and remote nodes form a private management network used for internal communication between the EMS, the cluster manager entity, and the network

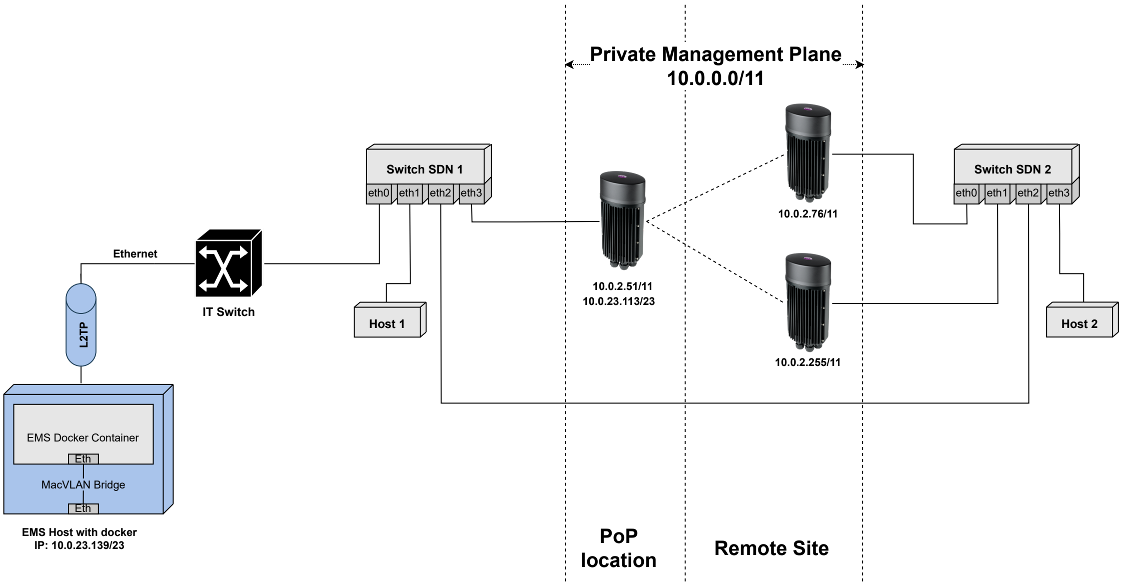


Figure 3.4. Deployed Testbed Infrastructure.

nodes. Typically, this network is extended from wired nodes at PoP sites to EMS via L2TP tunnelling. This management network carries different types of traffic, including heartbeat messages (between CM and nodes), HTTP requests (e.g., to retrieve link statistics information from nodes), SSH traffic, and ARP messages. Because there is no direct connection to the EMS, the traffic from remote nodes to wired nodes is routed using the mmWave links. However, once it reaches the wired node, it uses the L2TP channel to send it to the EMS. Because the wired node has an Ethernet connection to the IT's network, it also has an IP inside the laboratory network that could be used to access both wired and remote node's command line via SSH. Leveraging this, it was possible to develop bash scripts that automate the data collection process in the different scenarios, which allow testing over a long time window all possible communication paths (i.e., the different radio combinations) between a specific pair of stations. The process associated with this data collection is described in the next section.

3.2 Network Monitoring

This section describes the KPIs available for the testbed, and the process for data collection.

3.2.1 Multiple-layer Probing

The EMS GUI displays a set of real-time KPIs (e.g., RCPI, SNR, MCS, PER), that are collected for each mesh link, using the information reported by each node to the CM. The CCS Metnet web page can access these metrics stored inside the local Mongo database through the API. Following that logic, a python script was developed to access these values outside the web page context. The python script uses HTTP requests to access the backend API, so that an object containing the link metrics can be returned from the local EMS database. These metrics were later uploaded to a collection inside a newly deployed Mongo database hosted outside the EMS container. This database collection is composed of entries that display the instantaneous KPIs values for a specific path, which include:

- RSSI (dBm): the Received Signal Strength Indicator. Since the vendor's measured this at the antenna connector, it does not account receive antenna gain;
- SNR (dB): the Signal to Noise Ratio. This is a link quality metric that directly influences the IEEE 802.11ad standard, since it is used to select the best transmit and receive beams/sectors during beamforming training;
- PER (%): the Packet Error Rate, i.e., the ratio (in percentage) between the number of packets with errors after FEC and the total number of transmitted packets. An increase of the PER is expected in the case of temporary LOS obstruction;
- MCS TX: the Modulation and Coding Scheme (0-9) used during transmission in one direction (i.e., uplink or downlink). For more details regarding the type of modulation performed and respective code rate, refer to Chapter 4 in Section 4.2.4.
- MCS RX: the Modulation and Coding Scheme (0-9) used during transmission in the other direction (i.e., uplink or downlink).
- Beam Index TX (Max), Beam Index TX (Min) and Beam Index TX: the identifier of the maximum, minimum and average beam used for transmission, respectively, in a one second window;
- Beam Index RX (Max), Beam Index RX (Min) and Beam Index RX: the identifier of the maximum, minimum and average beam used for reception, respectively, in a one second window;
- Mbps TX (Max) and Mbps TX: the peak and average link transfer rate measured in Mbps.

Additionally, each table entry has specific tags that identify the type of link formed and the time of measurement. These are:

- Channel: the frequency channel used (1 to 4). For more information regarding each channel carrier frequency, please refer to chapter 2, section 2.1.2.
- Channel Type: determines if sub-channelization is used. By default and in this work context, this value is equal to zero indicating that sub-channelization is not used;
- Path ID: the communication link identifier. The uplink and downlink between each node pair has a different tag associated;

Table 3.2. Metrics obtained with the *iperf3*'s TCP and UDP test, and with ping command.

UDP metrics	**TCP metrics**	**Ping metrics**
Received Data Rate (Mbps)	Received Data Rate (Mbps)	No. of Transmit Packets
Jitter (ms)	Transmit Data Rate (Mbps)	No. of Received Packets
	No. of Restransmissions	No. of Packets lost
	Lost packets (%)	Packet Loss Rate (%)
		RTT Minimum (ms)
		RTT Average (ms)
		RTT Maximum (ms)
		RTT Mean Deviation (ms)
		Packet Duplicate Count
		Packet Duplicate Rate (%)

- Timestamp: the measurement time. This value is stored with a second precision.

The developed mmWave data monitoring script was set up as a service that runs in the Host 2' background through Linux's systemd. Hence, new mmWave link statistics were continuously updated in the external Mongo database. Any machine with access to the database can access the physical metrics stored referring to a given time window, and export them in a CSV or JSON format for further processing.

To fully characterize the mmWave network behavior across multiple layers in different scenarios, network metrics were collected simultaneously to the physical ones. For that purpose, the *iperf3* tool was used to develop a TCP and UDP-based network probe that can provide additional information regarding link quality at the transport layer. *iperf3* is a tool for active measurements of the maximum achievable bandwidth on IP networks. For each test, it reports the bandwidth, loss, and other related parameters [108]. The network probe developed consists of two main components: an application server and the respective client. For the client part, a python script was developed, which instantiates two parallel *iperf3* clients whose jobs are to collect UDP and TCP metrics. To avoid link saturation caused by TCP originated traffic, the TCP client was configured to only use a maximum of half of the Ethernet connection capacity (500 Mbps). Moreover, the *pingparser* library was used to parse the output of the system's ping, so that additional network metrics, such as RTT, can be obtained. The ping metrics were calculated by averaging the metrics measured for 10 packet transmissions. Table 3.2 shows the network metrics that can be obtained with each method. Moreover, the client script is also responsible for loading the three types of metrics at a given timestep, as a single entry in a new collection inside the deployed Mongo database. As for the server, it was implemented by simply running two *iperf3* servers in the same ports defined by the client script.

The network-level probe described was used to obtain network data relating to the two testbed's links in a sequential manner. Contrary to the physical data metrics, the network data of each link is not possible to be obtained at the same time. Instead, individual test runs of the network probe must be performed for each one of them, by deactivating the ethernet port connected to the SDN switch that is associated with the alternate path. Depending on the configured mode, the *iperf3* traffic flows more in one direction than the other, meaning the uplink or the downlink. By default, the client uploads to the server. However, the client can be configured to operate in reverse mode, making it download from the server instead.

After the deployment of these probes, it was possible to access all the metrics in an organized manner for a first analysis. While a capable probe was developed to load and access UDP metrics, it was found that they stopped being loaded into the database, during intermediate testing, due to a sporadic error. To avoid repeating these time-consuming experiments, it was decided to leave out UDP-related metrics in the algorithms developed in the following chapters.

3.2.2 Probing Verification

As previously discussed, mmWave-based standards, such as IEEE 802.11ad, are highly susceptible to blockage due to the directional nature of the communication. As such, sporadic blocking may cause a temporary decrease of supported MCS, which is further aggravated in continuous blockage scenarios where the decrease in received power is observed for a long duration of time. In more extreme cases, obstruction was also observed to cause total link failure, thus compromising the QoE and QoS even further.

In order to be able to correctly characterize the consequences of different types of blockage in our deployed backhaul's performance, the physical and network metrics described earlier shall be compared to distinguish the behavior under the three distinct blockage scenarios. These scenarios are defined according to the type and duration of the blockage event crossing the LOS path, and can be described as:

- **Normal scenario:** scenario where each path for each MCS mode tested operates under a non-obstructed environment for the entire test duration (15 minutes). In this scenario, the only factors limiting link quality are the distance between STAs and the communication channel itself. Each physical and network metric achieves the best possible value under normal operation. For instance, metrics that should be maximized (e.g., SNR, RSSI and TX/RX MCS), are set with the highest possible values, and metrics that should be minimized (e.g., PER, Packet loss) remain very low.
- **Short-term blockage scenario:** scenario where a link operates under temporary LOS blockage for a short time period. During the data acquisition in this scenario, all paths were short-term blocked for the entire experiment (5 minutes). In this scenario, the instants at which the rapid signal quality drops occur are unpredictable. Hence, high fluctuations relative to the average value are expected. Moreover, the greater the frequency of obstruction, the more rapid variations are to be expected;

- **Long-term blockage scenario:** scenario where each path for each MCS mode tested operates under a continuous LOS blockage for the entire experiment (15 minutes). Under continuous LOS blockage, the statistical behavior of each metric is expected to be similar to the one presented by metrics collected under the normal scenario, except for the average value, which should be lower for metrics that should be maximized, and higher for metrics that should be minimized.

To collect the data for each scenario, the mmWave monitoring service and the network data probe were used to measure and load the KPIs to a database. However, the firmware version used did not provide a GUI functionality that allowed selecting specific radios that establish the communication in each link node. Rather, the selection has to be manually done by changing a JSON file stored inside each remote node, which can be a tedious process, especially considering the number of different paths to be tested. To address this, a bash script was developed to automate the selection process. Besides configuring the appropriate parameters for testing a specific path (i.e., the radios used on each end), this script can test different paths sequentially and register the time at which a specific path measurement began and ended.

Thus, we tested each path and each MCS for the three scenarios for extensive periods. Besides that, the script eliminated the waiting time observed between configuring the node to use a particular radio and the link being effectively tested with that radio. Note that the way the equipment is configured, if it is not possible to establish a link between a pair of STAs using a given combination of radios, it has to wait 30 minutes for this configuration to be cleared, and only the connection is reestablished.

The script accepts three inputs which are used to perform the required node configurations: 1) the ID of the paths[1] to test; 2) a list of the MCS modes (either fixed [1-9] or dynamic) to test each path; and 3) the duration of each experiment. For instance, to test the path between PCP and B formed with radios 1 and 2, the script will perform the following steps sequentially:

- Get access to node B's command line using the wired node as a gateway;
- Change the JSON file inside node B to test a specific radio combination at node B (radio 2) and node PCP (radio 1). Note that a reboot is necessary at node B for those changes to be applied effectively;
- Continuous monitoring if path PCP/B/2/1 has been effectively established or not after the reboot. If that path is not formed within the first 5 minutes, the chances are that LOS communication is not possible using the required radio combination specified. In

[1] The term "path" refers to the physical link formed by a specific transmit and receive radio at each end. For example, a link between node PCP and node B could ideally have eight paths using the four radios on either side if the LOS condition was met.

that case, the script is instructed to wait the required 30 minutes until the JSON file is cleared.

- On the other hand, if the path has been established, the MCS mode is configured at each node end. The MCS mode can be configured to either be fixed at each node or set to automatic. Note that the received signal level must be at least equal to the one specified by the vendor to form a specific path between two stations with a given fixed MCS.
- The test initial timestamp is registered inside a text file, as well as the end timestamp estimate that was calculated using the test duration input.

The aforementioned steps were performed for all paths to test, thus obtaining a fully automated test configurator tool, which iterates over all MCS modes and registers the dates in which each experiment has started and ended. A generic description of how this testing tool works is given in algorithm 1.

After the development and validation of the test automation tool, data collection for each scenario was initiated. To collect the metrics in the continuous blockage scenario, a metallic object, such as the one shown in figure 3.5 was added between the stations that form each link. This obstacle was placed in a location where a significant reduction of the instantaneous RSSI and SNR was observed when compared to the average value registered in the normal scenario. The physical and network metrics for each experiment, in the long-term blockage and normal scenarios, were collected continuously during 15 minutes using the tool described in the algorithm 1. The short-term blockage scenario data was obtained for each path, by having one and two people at a time, moving freely during 5 minutes, in the location corresponding to the estimated LOS of the link being tested. These movements were sometimes made in a completely random manner, and in other times they were performed in an alternating way following a horizontal, vertical and diagonal line to the line where the link between the two nodes to be tested was estimated to exist. Because obtaining data in this scenario requires people to be blocking the link, it was unrealistic to collect data for all possible paths between a pair of stations with all MCS modes for 15 minutes. Instead, the test was only done for a smaller set of paths with automatic MCS for 5 minutes using the same tool.

Also, given that the tests for the three scenarios were carried out over the course of a month (a non-continuous process), the datasets obtained already reflect the variations that happen in the channel due to atmospheric conditions, such as wind and fog.

At the end of each experiment, the recorded timestamps that mark the beginning were used to retrieve the transport, network, and physical metrics from the database, which were used later to analyze the consequences of long-term and short-term blockage. This study is presented in chapter 4. Furthermore, this data was fed to a wireless link quality classification algorithm and a forecasting method after a processing stage, which are described in chapter 5.

Note that, there were many situations where the introduction of long-term and short-term blockage led to the temporary link failure, making it impossible to acquire data in extremely blocked scenarios. In those cases, the MCS would switch back to automatic mode even when previously set. Because of that, the data entries collected after a link failure were removed from the fixed MCS ones, and thus were not considered neither for the analysis nor for the machine learning algorithm, depicted in chapter 5, to avoid additional complexity.

Algorithm 1: Automating the test of different paths and MCS modes

```
Data: The pair of STAs to test ($Wired, Remote$)
      A list of paths to test $P = [p_0, p_1, p_2, p_3, p_4, ..., p_{n-1}]$
      A list of MCS modes to test $M = [1, 3, 5, 7, 9$ and $Automatic]$
      The experiment duration $D$ in minutes
Result: Text file with the start and end timestamps for each experiment
for $p_i$ in $P$ do
    ssh Remote
        Change JSON file to test radio combination of $p_i$
        Reboot
    end
    $link_{ON} = False$
    while $\neq link_{ON}$ do
        Read from database active links
        if $p_i$ is established then
            break;
        end
    end
    for mcs in M do
        ssh Remote
            Change MCS TX to mcs
        end
        ssh Wired
            Change MCS TX to mcs
        end
    end
    Register $t_{start}$ as the current time
    $t_{end} = t_{start} + D$
    Write the $t_{start}$ and $t_{end}$ in the output file
end
```

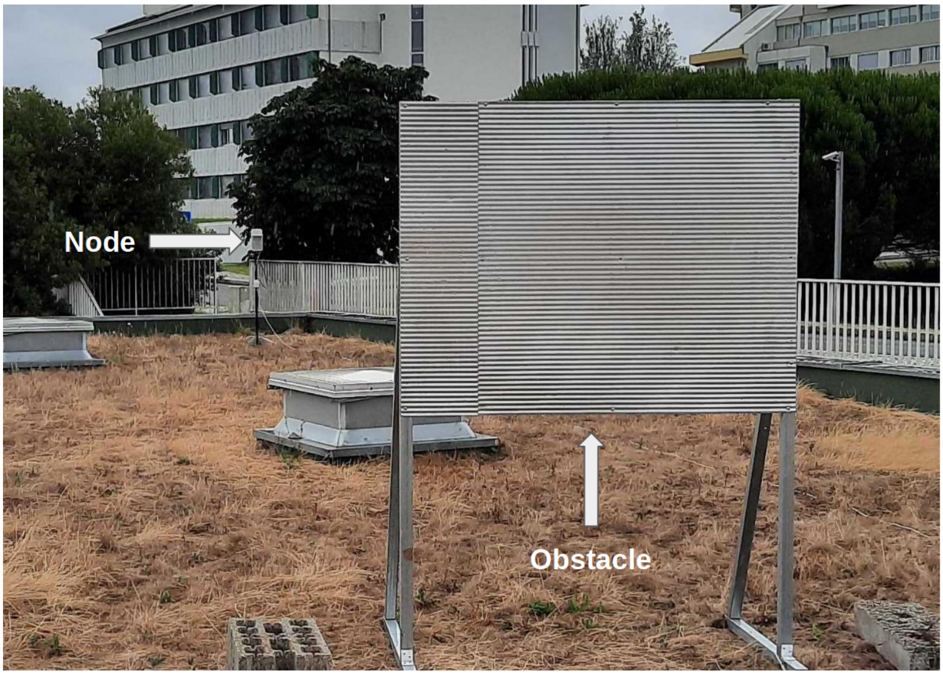


Figure 3.5. Metallic obstacle used for the long-term blockage scenario.

3.3 Summary

This chapter described the functionalities and limitations of the commercial solution used, as well as the processes required for deploying a physical testbed in an outdoor environment. More specifically, it presented the architecture used to successfully deploy a mmWave backhaul that employs the standardized IEEE 802.11ad. Afterwards, it described the process followed to acquire transport, network, and physical layer metrics. For this purpose, a mmWave monitoring service was developed to continuously collect physical metrics and upload them to a database that could be easily accessed. Furthermore, a multi-layer probe was developed, which measures and loads higher layer metrics in a database that can be easily accessed.

Finally, it discussed the process of testing mmWave links under normal, long-term, and short-term blockage conditions using a custom automation script. The multi-layer dataset built - with the tools and methodologies proposed in this chapter -, allowed us to characterize the impact of long-term and short-term blockage in a mmWave backhaul. The next chapter will discuss the collected metrics, regarding each scenario, initiating an exploratory data analysis.

Chapter 4

Multi-layer Data Analysis

With the metrics collected, we proceed to analyze the impact of long-term and short-term obstruction on the quality of millimeter-wave links at multiple layers. This chapter presents the results obtained from this analysis. The chapter is organized into two sections, one for each type of metric measurement. The first section presents the behavior of the physical and MAC-level KPIs under short-term and long-term blockage, while the second section details the impact of blockage in the transport and network-layer metrics.

4.1 MAC and PHY Layers

Several conclusions can be drawn by jointly analyzing the metrics collected at the physical layer for different paths. The next section provides insights relating to the most relevant physical and MAC-level KPIs: Signal to Noise Ratio (SNR), Packet Error Rate (PER) and Modulation and Coding Scheme (MCS).

4.1.1 Signal to Noise Ratio

This section discusses the results obtained with respect to the SNR. Some considerations are depicted as follows:

1. A link suffering from either long-term or short-term blockage has a lower average SNR when compared with the non-blocked scenario;

2. The mean SNR of the uplink and downlink of a given path may not be similar, even in a non-blocked scenario. In fact, this is the case for almost all analyzed paths, since they use different beams for transmitting and receiving to avoid interference. Thus, each direction has its communication channel, which is likely to be submitted to different conditions;

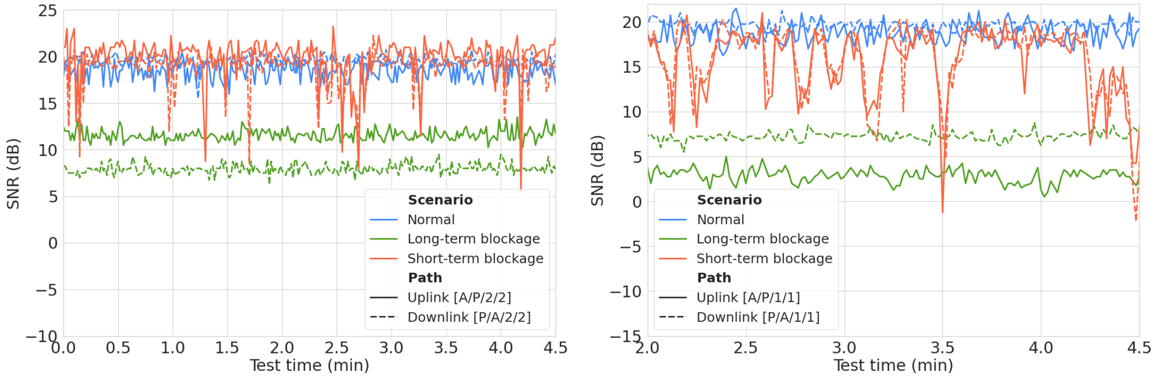
3. The SNR of two path directions suffering from long-term blockage may not experience the same average drop, even if both uplink and downlink have similar values under regular operation. Depending on the location and rotation at which the obstacle is placed, one communication channel direction may be affected more than the other;

4. The average SNR drop is not the same for all long-term blocked links, since different paths (formed by different radios) cover different regions of the space. Thus, placing an obstacle in the LOS connection formed by two radios facing each other is more likely to affect the signal quality than in paths formed by radios on different sides;

5. Under short-term blockage, the uplink and downlink formed between a pair of STAs followed a similar behavior, apart from the maximum value reached that is close to what was registered in LOS conditions. In both directions, the LOS blockage occurred at the same instants, although with different intensities;

6. In some cases, the SNR drops were more pronounced when the link was short-term blocked than when it was long-term blocked. This makes sense as there is a higher chance of achieving complete beam obstruction.

Figure 4.1a and 4.1b obtained for path paths P/A/2/2 and P/A/1/1, respectively, present examples in which all the conditions described above are met under the three scenarios. Firstly, the uplink and downlink's average SNR are different in all three scenarios. Moreover, both directions present a similar behavior for the same scenario, since the instants at which transitions are observed are the same. Lastly, in a short-term blocked link, the SNR alternates between a maximum value close to the non-obstructed scenario's average and a minimum value that may be lower than the mean registered in the long-term blocked scenario. The minimum and maximum SNR are obtained in instants where the LOS path is crossed or not, respectively. This behavior was also verified for the SNR of the remaining 2 paths that could be established between node P and A, the graphs of which are shown in figures A.1a and A.1b of the appendices.



(a) Path P/A/2/2.

(b) Path P/A/1/1.

Figure 4.1. SNR of two mmWave paths under normal, long-term blocked and short-term blocked scenarios.

Because data is acquired with a 1-second periodicity, determining the exact value of the duration of the SNR drop caused by short-term human obstruction is not possible. Nevertheless, since each metric collected represents the average measured for that time window, we get an upper limit of 1 second. Therefore, it is observed that each line-of-sight obstruction is reflected, in most cases, in the generation of a single point.

4.1.2 Packet Error Rate

This section discusses the results obtained with respect to the Packet Error Rate. Some considerations are depicted as follows:

1. The mean PER is higher for a long-term blocked link than for a link in normal operation when the MCS is fixed, since there is consistently lower SNR caused by a lower RSSI;

2. Even if a specific link has a SNR that is equal or superior to the established threshold for a given MCS, supporting it with negligible PER is only possible if the RSSI is also superior to the minimum defined in the standard;

3. Non-obstructed links operating with fixed MCS mode present negligible PER (mostly under 1%);

4. Non-obstructed links operating in the automatic mode present a larger variability than in the fixed mode (usually up to 10%) due to slight MCS adjustments;

5. Generally, a PER peak is detected, regardless of the scenario, at instants where a significant signal quality decrease occurred, such as human obstruction or natural factors (e.g., strong wind);

6. Setting a higher MCS in a long-term blocked link results in increased PER as the channel's margin decreases;

7. The PER of links operating in the automatic MCS mode present more significant fluctuations when compared to the values recorded for the same scenario with the fixed MCS. These fluctuations are due to the node's ability to vary the modulation, and the redundancy introduced accordingly to the measured SNR and PER;

8. As an extension of the previous statement, channels that face short-term obstruction present a larger PER variability when compared with either long-term blocked and non-obstructed channels. The PER remains smaller under long-term blockage scenario because there are not as many rapid channel fluctuations;

9. Still related to the previous topic, under the short-term blockage, the PER is more likely to achieve its maximum value at times the LOS path is unexpectedly blocked, and its minimum value at instants where there is LOS connection;

10. Even in non-obstructed and long-term blocked scenarios, where most SNR fluctuations are still close to the mean, the PER may sporadically reach peak values, comparable to those obtained with short-term blockage, due to natural causes;

11. Generally, a PER increase is observed whenever a significant SNR decrease is detected due to natural causes or induced obstruction. However, the rapid SNR variations do not always necessarily lead to higher error rates.

Figure 4.2 shows how the PER of the uplink A/P/2/1 is affected by the MCS increase under long-term blockage. This specific link had an average SNR of 7.5 dB, which in theory is enough to support modulations up to MCS 7 (table 4.1) as its threshold is set to 5.3 dB. However, this path only received on average -75 dBm (before antenna gain), which does not comply with the -68 dBm of MCS 7 nor the -69 dBm requirement of MCS 5. Thus, the uplink cannot maintain a reliable connection with either one of these schemes. Figure 4.3 confirms that, since it shows that long-term blockage has caused PERs to be contained within the $30 - 100$ range for the MCS 7, while in the non-obstructed environment it is null.

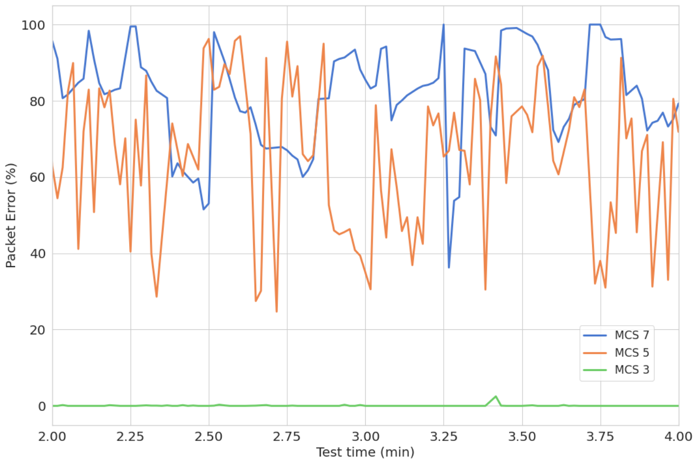


Figure 4.2. The influence of the MCS increase in the PER for link A/P/2/1 under long-term blockage in automatic MCS mode.

Table 4.1. Single carrier PHY mode supported by the CCS Metnet node.

MCS	Modulation	FEC rate	RSL min (dB)	SNR (dB)	Layer 2 Line Rate (Mbps)
0	DSSS	12	-84.52	-11	22.4
1	BPSK	1/2	-73.72	-0.2	308
2	BPSK	1/2	-72.52	1	616
3	BPSK	5/8	-71.32	2.2	770.4
4	BPSK	3/4	-69.92	3.6	924
5	BPSK	13/16	-69.02	4.5	1000.8
6	QPSK	1/2	-69.72	3.8	1232
7	QPSK	5/8	-68.22	5.3	1540
8	QPSK	3/4	-66.72	6.8	1848
9	QPSK	13/16	-65.72	7.9	2002.4

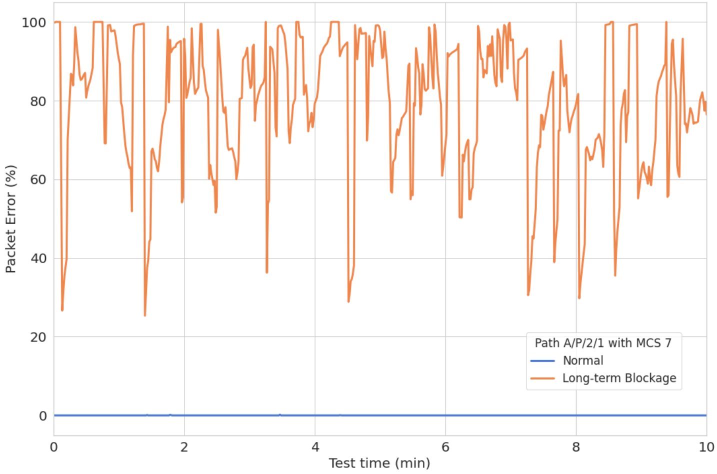


Figure 4.3. Comparison between the PER obtained for uplink A/P/2/1 under normal and long-term blockage scenarios with MCS 7 fixed.

In links that operate with an automatic MCS mode, the PER fluctuations observed are much more controlled than those obtained with the fixed MCSs tests. Auto mode allows the modulation scheme used by a link to be adjusted as soon as there is a significant drop in signal quality and/or when high PERs are experienced. This capability causes instants that follow PER peaks to exhibit lower PERs soon after the maximum, as can be seen in figure 4.4 for link A/P/2/1. This image shows evidence that a link suffering from long-term blockage is more likely to present larger PER variability than when it is short-term blocked or operating in LOS conditions. Still, this relationship is not always true. The existence of higher PERs depends on different aspects of the positioning of the transient obstacle, such as its rotation, its distance from the transmitter or receiver, among others. Therefore, it is not possible to estimate precisely how large the impact of the two types of obstructions will be without knowing the specific details of the blocker's position.

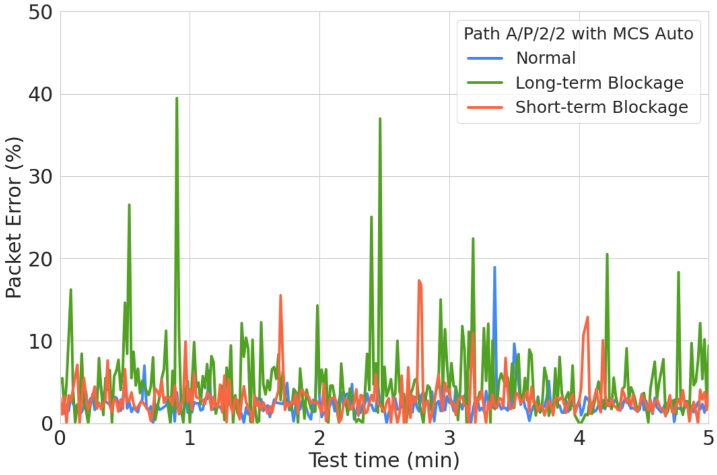


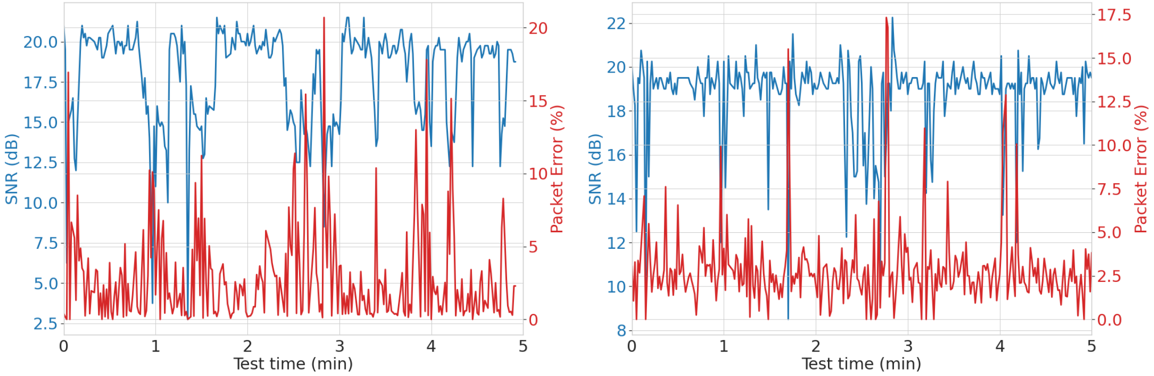
Figure 4.4. Comparison between the PER obtained for uplink A/P/2/2 under normal, long-term and short-term blockage scenarios with automatic MCS.

As shown by the PER statistics of table 4.2, long-term and short-term blocked scenarios can cause PERs that can be as high as 64% and 88%, respectively. The results of this table are further detailed in table A.1 of the appendices. Furthermore, the experiments showed that a received power below the -70 dBm mark leads to PERs contained withing the 90 to 100% range in relatively low fixed MCS. Note that, with -70 dBm, only a MCS equal or below 3 can be supported (table 4.1) with negligible PER.

Table 4.2. Statistics of the measured Packet Error Rate (PER) in the worst-performing link between P and A for the different MCS modes and the three scenarios.

PER (%)					
MCS Mode	**Scenario**	**Mean**	**Std dev**	**Min**	**Max**
MCS 1	Normal	0	0	0	0
	Long-term blockage	1,3	5,1	0	46,7
MCS 3	Normal	0	0	0	0
	Long-term blockage	10,2	24,7	0	100
MCS 5	Normal	0	0	0	0
	Long-term blockage	49,1	43,2	0	100
MCS 7	Normal	0,3	0,0060	0	1,8
	Long-term blockage	66,8	33,4	0	100
Auto	Normal	1,6	1,4	0	19,0
	Long-term blockage	6,4	9,3	0	64,7
	Short-term blockage	2,4	4,8	0	88,2

Finally, the relationship between signal quality drops and the error rate was explored further. Figure 4.5a demonstrates an occasion where the PER varied at a rate roughly equal to that of the SNR. In this figure, it is possible to observe that the error measured for the uplink A/P/2/1 during short-term blockage is lower at instants where the signal quality is higher, and is higher when the signal quality drops. If that were true, we would expect the error rate always to be higher whenever the SNR decreased. However, as shown by the experimental results of uplink A/P/2/2 in figure 4.5b, SNR drops do not always translate to PER maximums. Instead, it is necessary to describe its behavior as the result of a complex interaction of multiple aspects, such as the code rate of the FEC mechanism at the PHY layer used to code the packets. In other words, achieving or not a maximum PER value depends more directly on whether the amount of redundancy introduced by the MCS at the time when the obstruction occurs is sufficient to compensate for the additional error introduced by natural or blockage-induced channel variations, rather than the obstruction value itself.



(a)

(b)

Figure 4.5. PER and SNR observed for two mmWave links operating in automatic MCS mode, which are submitted to short-term blockage.

4.1.3 Modulation and Coding Scheme

This section discusses the results obtained when changing the modulation and coding scheme. Some considerations are depicted as follows:

1. Increasing the MCS on a link enables it to transfer data at higher bit rates, if the received signal strength and SNR meet the outlined threshold values;
2. In the automatic mode, the link MCS is dynamically adjusted according to both SNR and PER. Therefore, a decrease in the modulation supported may be triggered by either detecting a significant SNR loss, a high PER or both;
3. As soon as the line of sight is no longer obstructed, the MCS is reset to its pre-loss value.

The CCS mesh nodes used adjust the MCS used for a communication link according to both the SNR and PER measured at a given instant. Figure 4.6 shows the MCS adjustment made by the system to establish and maintain a link between node A and P (using radio 2 at both ends) that is submitted to short-term blockage. In this figure, it is clear that time windows with fewer MCS changes are the ones that present small SNR fluctuations around the mean value of 20 dB and a controlled small PER. In these windows, the MCS is set up to the maximum possible value (9) for that connection, only changing occasionally to a smaller MCS (8) due to sporadic small decreases in the signal quality.

Because dynamic MCS adjustment is performed on a instantaneous SNR and PER basis, it was difficult to pinpoint the MCS decrease to a single metric. Instead, a careful analysis had to be performed considering both metrics to understand the instantaneous MCS changes properly. Under this work context, the MCS changes were either due to one of three different conditions. Firstly, the MCS decrease can be triggered due to a significant SNR drop alongside a high PER, as shown in instant t_6 of figure 4.6, where the MCS decreases to 8. Secondly, the MCS can be changed solely due to the presence of high error rates, as shown in instants

t_2 and t_4. A maximum PER of 15% and 11% was registered at t_2 and t_4, respectively, while the SNR value remained practically unaltered. Thirdly, the MCS change can be triggered in some cases due to a signal quality decrease, which is the case of instant t_3. In t_3, the SNR has decreased from 20 dB to 9 dB with a 5% error rate.

Still, it is only possible to delineate a general behavior of the dynamic adjustment algorithm. As shown in instant t_1 and t_5, not every decrease in signal quality and/or increase of the PER results in a MCS adjustment.

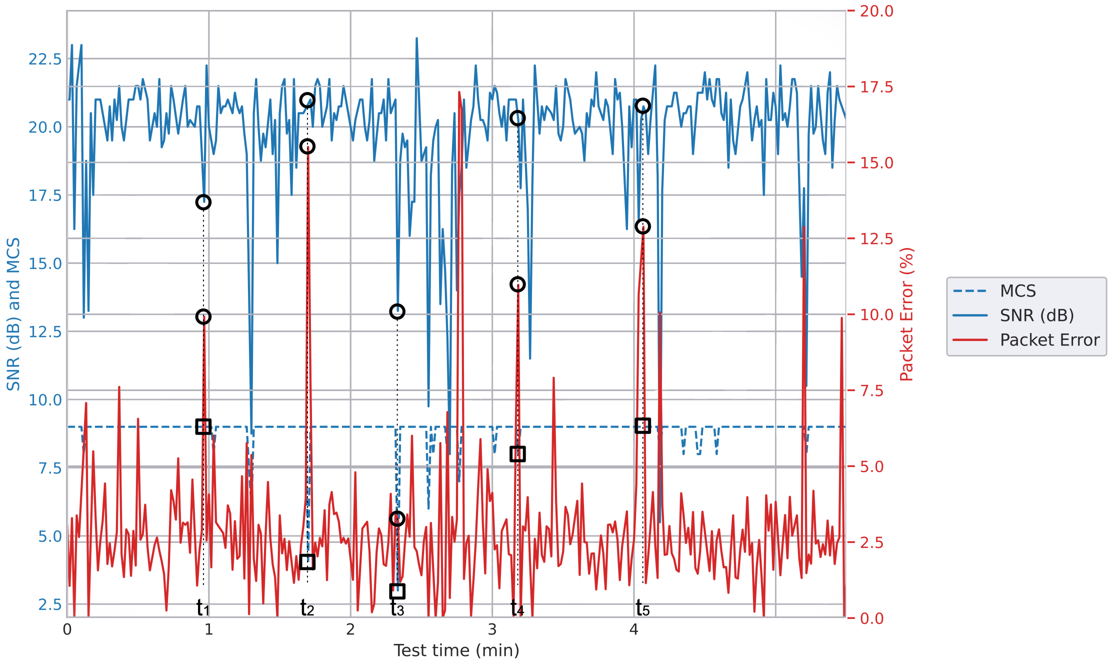


Figure 4.6. The influence of the SNR and PER in the dynamic MCS adjustment mechanism for link A/P/2/2 under short-term blockage.

4.2 Network and Transport Layers

Once the impact of the main KPIs at the physical layer is understood, these findings are used to detect general patterns perceptible to the human eye at the transport layer: more specifically, the behavior of packet delay and TCP performance in highly variable millimeter-wave networks. The following section provides insights relating to the most relevant TCP and network-level KPIs: Round-trip Time (RTT), packet loss, TCP throughput, and TCP retransmissions.

4.2.1 RTT

To obtain the RTT metric, the *pingparser* library pings the client host for a given amount of time. In each time window, the 10 ICMP packets transmitted through the network were used to compute an average, maximum and minimum round-trip-time for each possible communication path and every scenario. The following discussion presents the findings from our analysis for the average and maximum RTT.

1. A non-blocked link has an average value that decreases as the MCS index increases (for the fixed mode). Even so, some small fluctuations around the mean are observed due to the varying nature of the millimeter-wave channel;
2. In a non-blocked link, the average RTT value measured with MCS 1 is 10 times higher than any other MCS tested, since the throughput traversing the network is at least 2.2 times lower than that observed in other tested modulations;
3. Links operating with higher index MCSs (i.e., excluding MCS 1) in both modes, under a non-obstructed environment, have a maximum average RTT that does not surpass the 20 millisecond mark. Also, the calculated standard deviation is always lower than 3 milliseconds;
4. The average RTT of a long-term blocked link resembles the behavior observed for links operating in the non-blocked state when the PER is low;
5. Long-term blockage and short-term blockage is observed to cause larger variability in the mean and maximum RTT value when compared with the normal scenario;
6. A heavily blocked link can have a maximum RTT which can be in the order of one or a few seconds;
7. Statically and short-term blocking a mmWave link using the automatic MCS mode causes a significant increase in both the average and maximum RTT when compared with the values registered in LOS conditions;
8. Links suffering from obstruction and high PERs present higher average RTTs for MCS with the lowest code rates as they are less likely to compensate the error introduced by the channel;
9. A less robust MCS will lead to an increase in the number of retransmissions;
10. The average RTT of all paths formed between a pair of stations shows similar average values for the same MCS.

Figure 4.8 shows the RTT measured for the link A/P/2/2 in the non-blocked and long-term blocked scenarios. Through the analysis of subfigure 4.8a, it is possible to observe that, under normal conditions, the mean RTT is higher for lower MCS. This is expected since, as shown by table 4.1, lower MCSs have lower associated data rates. On the other hand, it can be seen that, under long-term blockage, the link experiences higher delays in less robust MCS (i.e., MCS with lower code rates). For instance, in subfigure 4.8b, a maximum RTT of $150ms$ is experienced with MCS 5 (which uses a 13/16 code rate), while at MCS7 (which uses 5/8 code rate) that value drops to 85ms. MCS 1 follows the same behavior observed in its non-blocked state, since the PER is close to zero. That is, even under an obstruction, the received signal strength is enough to maintain a reliable link for such robust modulations schemes. This figure also demonstrates that the introduction of long-term obstruction causes high fluctuations of the average RTT, even if the modulation scheme is fixed. This behaviour is also observed for the case where the MCS is automatically adjusted for both long-term and short-term blocked links, as can be seen by the graph in figure 4.7. This variability makes it difficult to estimate with certainty the extent of the obstruction's impact.

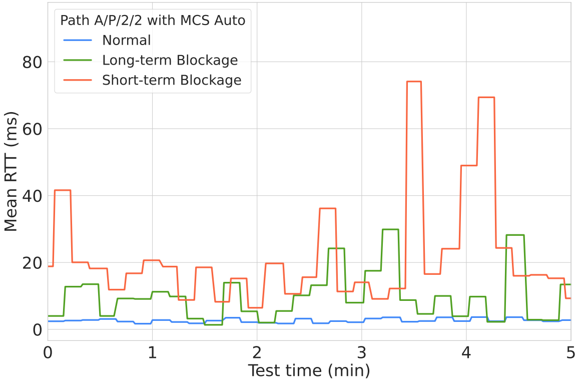
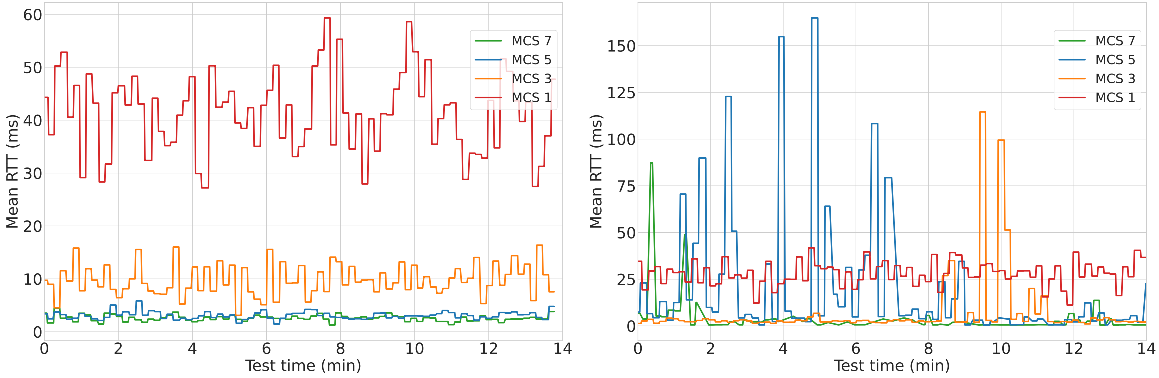


Figure 4.7. RTT in link P/A/2/2 under normal, long-term and short-term blocked scenarios for the automatic MCSs.



(a)

(b)

Figure 4.8. Average RTT observed for the uplink A/P/2/2 under normal operation (a) and long-term blockage (b).

Tables 4.3 and 4.4 detail the statistics of the average and maximum RTTs, respectively, registered for the worst-performing link under non-obstructed, long-term blocked and short-term blocked scenarios. The results in these tables show that the worst-performing links under long-term and short-term blockage have average RTTs around 1.3s, 0.8s, and 0.5s and a maximum RTT of 4.7s and 1.8s for different fixed MCSs. Note that, the final results presented on these tables were obtained by interpreting and aggregating data from the tables presented in the appendices. Specifically, tables A.2, A.3 and A.4 for the mean RTT; and tables A.5,

Table 4.3. Statistics of the average RTT metric in the automatic MCS mode for the worst-performing link.

	RTT Mean (ms)				
MCS Mode	**Scenario**	**Mean**	**Std dev**	**Min**	**Max**
MCS1	Normal	41,67	7,93	18,66	58,75
	Long-term blockage	42,85	134,53	14,51	1324,96
MCS3	Normal	9,98	2,87	3,45	17,15
	Long-term blockage	167,88	230,26	0,72	819,34
MCS5	Normal	3,09	0,72	1,47	5,83
	Long-term blockage	208,18	138,27	5,66	554,03
MCS7	Normal	2,47	0,59	1,5	4,5
	Long-term blockage	3,8	8,96	0,49	87,32
Auto	Normal	2,65	1,34	2,65	12,26
	Long-term blockage	5,51	15,34	0,46	133,99
	Short-term blockage	25,3	6,46	21,18	93,15

Table 4.4. Statistics of the maximum RTT metric in the automatic MCS mode for the worst-performing link.

	RTT Max (ms)				
MCS Mode	**Scenario**	**Mean**	**Std dev**	**Min**	**Max**
MCS 1	Normal	79,33	4,57	58,44	88,7
	Long-term blockage	112,5	478,62	36,29	4677,09
MCS 3	Normal	24,25	7,09	6,49	40,03
	Long-term blockage	402,37	491,59	1,15	1846,55
MCS 5	Normal	7,69	3,05	3,61	21,82
	Long-term blockage	684,92	323,73	13,49	1360,27
MCS 7	Normal	6,55	2,84	2,34	20,11
	Long-term blockage	16,53	47,25	0,61	401,3
Auto	Normal	8,66	12,84	2,33	110,22
	Long-term	32,93	105,73	0,59	897,83
	Short-term	35,5	66,94	2,8	529,78

A.6 and A.7 for the max RTT. Consultation of these tables' content is advised to anyone who wishes to understand better the impact of obstruction in each link of the tested MCS combinations.

Lastly, as shown in table A.2 of the appendices, the mean and max RTT remained practically constant across all paths formed between node A and P when operating under a non-obstructed environment.

4.2.2 Packet Loss

In our specific context, two types of retransmission mechanisms can be identified that act at two different layers. Firstly, retransmissions can be made at the MAC layer level of IEEE 802.11ad. Secondly, TCP-based retransmissions can be also performed at the transport layer. The following discussion presents our findings from the analysis of the packet losses at the transport layer.

1. A non-obstructed mmWave link using both automatic and fixed MCS usually present a

low average packet loss (less than 4%), with only some sporadic losses occurring due to timeouts;

2. TCP is not adapted to work with the new PHY and MAC layer mechanisms of IEEE 802.11ad;

3. The higher the MCS, the more likely the link is to suffer from packet losses. In these conditions, a higher number of TCP retransmissions will be triggered to overcome the channel erasures;

4. Short-term blocking a link may lead to packet losses up to 9% even under dynamic MCS adjustment;

As shown in table 4.5, a non-obstructed link usually presents a very low packet loss for all supported MCSs. Still, some sporadic losses were observed in normal scenarios (e.g., the 11.4% losses measured in the normal scenario with MCS 1). These losses are likely caused by the timeouts when the channel is busy performing the 802.11ad MAC mechanisms (e.g., beamforming training). As was reported by authors of [73], TCP does not change its timeout limit to account for these new operations, and thus it is forced to discard any packets that were not processed before the channel was occupied. Additionally, table 4.5 shows that long-term blocking a mmWave link can cause average packets losses of about 98% when a high MCS is fixed. Furthermore, it is demonstrated that short-term blocking a link can cause maximum packet losses up to 9% in the automatic MCS, contrasting with the 3% packet losses of non-obstructed and long-term blocked links. Figures 4.9 and 4.10 depict these aspects. It can be seen that, on a highly obstructed link measured in the long-term blocking scenario, the higher the MCS used, the higher the losses. Conversely, when the MCS can be dynamically adjusted according to the channel conditions, the losses incurred are lower than those registered in with high fixed MCSs. Moreover, it shows that long-term and short-term blocking a mmWave link leads to higher average packet losses than in non-obstructed environments.

Tables A.8, A.9 and A.10 in the appendices present the intermediate results obtained for each link with the fixed and dynamic MCS modes under normal, long-term and short-term blockage conditions, respectively.

Table 4.5. TCP packet loss statistics of the worst-performing link under the different scenarios for each MCS mode tested.

		Packet Loss (%)			
MCS Mode	**Scenario**	**Mean**	**Std dev**	**Min**	**Max**
MCS 1	Normal	4.96	2.19	1.45	11.54
	Long-term blockage	1.77	0.30	01.06	2.93
MCS 3	Normal	0.93	0.46	0.012	3.96
	Long-term blockage	12.45	7.49	0.01	21.77
MCS 5	Normal	0.27	0.43	0.01	2.41
	Long-term blockage	49.20	36.46	2.86	100
MCS 7	Normal	0.32	0.52	0.006	2.11
	Long-term blockage	97.53	09.09	37.5	100
MCS Auto	Normal	0.45	0.51	0.008	2.26
	Long-term blockage	0.19	0.35	0.008	2.59
	Short-term blockage	01.05	0.94	0.01	8.68

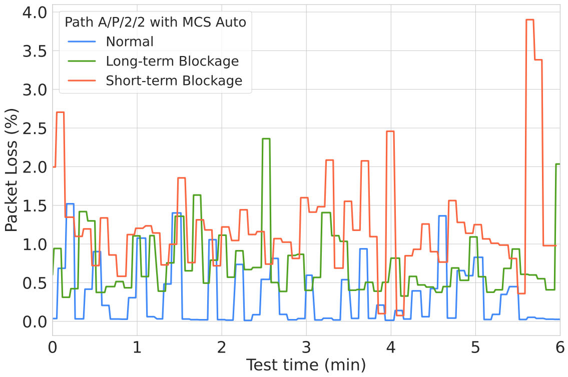


Figure 4.9. TCP packet losses for path P/A/2/2 under normal, long-term blocked and short-term blocked scenarios with the automatic MCS mode.

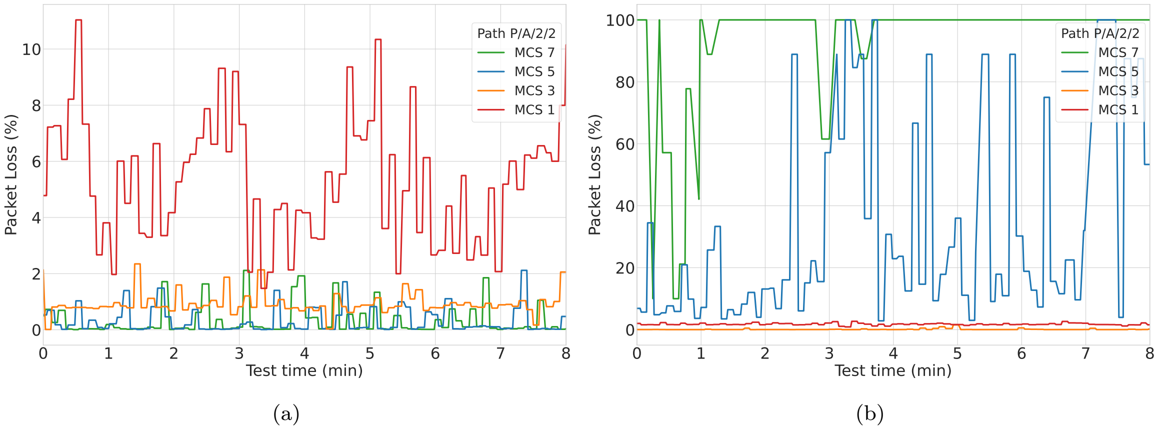


Figure 4.10. TCP packet losses for path P/A/2/2 in the normal scenario (a), and in the long-term blocked scenario (b) for the fixed MCS mode.

4.2.3 TCP Throughput

The following discussion presents the findings from our analysis of the TCP throughput.

1. Increasing the MCS in a non-obstructed link leads to an increase in the maximum achievable throughput;
2. Long-term blocking a mmWave link causes a high PER in higher index MCSs, which in turn will decrease the average achievable data;
3. In the worst-case scenario, a heavily blocked link can have transmitted rates as low as 1 Mbps when using fixed MCSs. This means that the network's delivery capacity is so low that TCP reduced its congestion window from 500 to 1 Mbps. Thus, only allowing traffic generated by the retransmissions to circulate through the network in an attempt to deliver the traffic to its destination successfully;
4. A short-term blocked link performing dynamic MCS adjustments has larger data rate fluctuations, as a sudden SNR decrease can make the modulation go from MCS 9 to MCS 1 within a second.

Table 4.6 shows the results obtained for the worst-performing link for each MCS mode. Several conclusions that validate the previous statements can be taken. Firstly, increasing the modulation scheme index increases the achievable data rate, as we went from transmitting at a 67 Mbps rate in MCS 1 to 496 Mbps with MCS 9. Secondly, the long-term blockage caused the average TCP throughput to drop significantly in higher index MCSs. For instance, long-term blockage reduced the throughput from 496 Mbps in the non-blocked scenario to almost 0 Mbps in the long-term blocked one. This observation can be further confirmed by the results of figure 4.12 for link P/A/2/2, since both a fixed MCS of 5 and 7 were not able to transmit more than a few Mbps when long-term blocked. Still, there was an irregularity observed for all paths tested. More specifically, the data rate increased with the introduction of the obstacle in the MCS of 1. Thirdly, the data rate of a link suffering from a short-term blockage varies very quickly in the automatic MCS mode, as shown in figure 4.11. This behaviour is a result of the sudden changes of the MCSs that are made in an attempt to control the PER.

As a final remark, tables A.11, A.12 and A.13 of the appendices should be consulted by anyone who wants to gain a better understanding of the results.

Table 4.6. TCP throughput statistics of the worst-performing link for each MCS mode tested.

		TCP throughput (Mbps)			
MCS Mode	**Scenario**	**Mean**	**Std dev**	**Min**	**Max**
MCS 1	Normal	66,42	19,8	17,29	101,55
	Long-term blockage	238,52	5,12	225,42	250,05
MCS 3	Normal	467,79	18,84	423,51	499,08
	Long-term blockage	64,55	130,45	4,2	444,74
MCS 5	Normal	495,75	11,1	387,82	499,9
	Long-term blockage	1,06	2,14	0	14,21
MCS 7	Normal	496,2	8,29	441,01	499,87
	Long-term blockage	0	0,01	0	0,1
MCS Auto	Normal	495,23	7,27	452,59	499,87
	Long-term blockage	328,76	59,99	213,14	488
	Short-term blockage	273,11	82,15	75,05	365,36

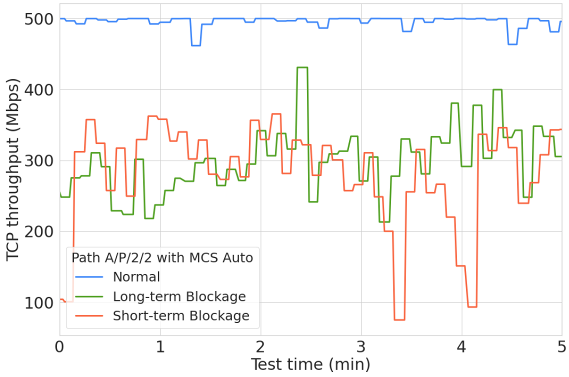
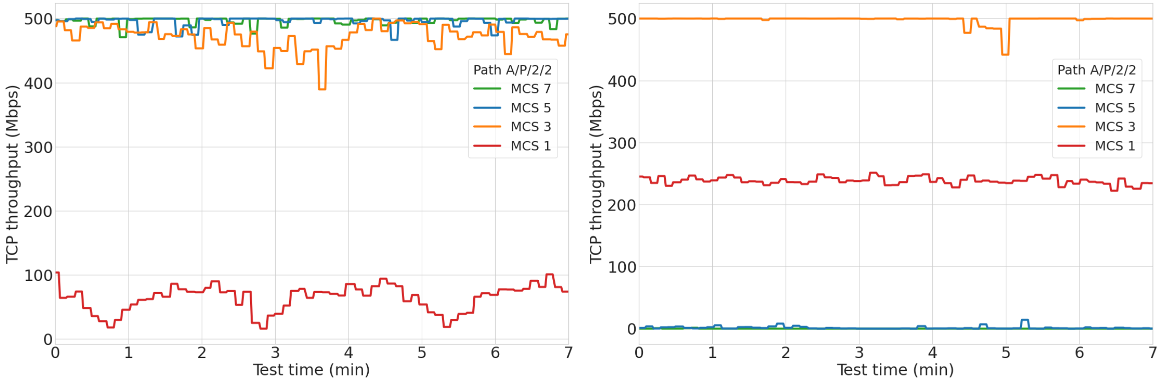


Figure 4.11. TCP throughput of the P/A/2/2 link operating under normal, long-term and short-term blockage conditions for the auto MCS.



(a) P/A/2/2 in the normal scenario.

(b) P/A/2/2 in the long-term blocked scenario.

Figure 4.12. TCP throughput of the P/A/2/2 link operating under normal (a) and long-term blockage (b) conditions with fixed MCSs.

4.2.4 Retransmissions

To properly compare the impact of each scenario in the number of TCP retransmissions, the obtained TCP retransmits metric was normalized using the average TCP received rate. This enabled to remove the effect of higher modulations on the transmitted bit rate and, consequently, in the total retransmissions. The following statements summarize the conclusions drawn from the analysis of the data collected for the non-obstructed, long-term blocked and short-term blocked scenarios.

1. The total number of TCP retransmissions is higher in non-blocked links (which supports high MCS) than in blocked links. On average, non-blocked links carry a much higher bit rate than a long-term blocked link, which means that a larger number of packets must be delivered to the receiving STA. Thus, it takes a higher number of retransmissions every second to deliver the information, even though the number of retransmissions required per packet is lower;

2. The normalized number of retransmissions of links under long-term blockage is much higher than in non-blocked environments for a fixed MCS, which implies that, on average, more packet retransmissions are needed in order to deliver a given packet to its destination;

3. In a long-term blocked link with high PER, setting higher MCSs increases the number of TCP retransmissions, and thus, decreases the maximum achievable TCP throughput;

4. Heavily blocked links in the long-term blocked environment have a limited delivery capacity when used with TCP, as it mainly carries traffic from pending retransmissions instead from useful new information;

5. Changing the MCS to higher index MCS with lower code rates in heavily blocked scenarios increases the number of normalized TCP retransmissions;

6. Using MCS with higher coding rates makes it less likely that the TCP performs a high number of retransmissions in a link submitted to short-term blockage;

7. The prediction of the behavior of the number of TCP retransmissions in scenarios where the MCS is adjustable, especially in obstructed environments, is rather challenging as the instants at which the increases in the number of retransmissions occur are not linearly related to the instantaneous signal quality.

Figure 4.13 shows the impact that long-term blockage has on the total and the number of retransmissions per Mbps. At a first glance, it may appear that a non-blocked link is somehow worse than the long-term blocked one, but that is not true. Removing the effect of the maximum achievable throughput enables a correct analysis of the blockage impact. The truth is that a long-term blocked link in which the MCS is fixed is unable to adjust its MCS whenever a high PER is detected. Thus, maintaining the MCS leads to a consistently high PER, which in turn forces TCP to trigger a high number of retransmissions to deliver the packets without error.

In principle, the higher the average packet losses, the higher the number of retransmissions are needed to deliver a packet successfully. However, the collected metric is not referring to the average number of retransmitted packets per transmitted packets, but rather to the mean number of retransmissions required to deliver the totality of the packets. Hence, as shown in figure 4.13, more packets are retransmitted at each second in the normal scenario (which has a higher capacity) than for the long-term one for the same modulation scheme. The latter observation provides evidence that CCS nodes most likely use an in-order delivery strategy for flow control purposes. Similar to that described in [73] for a TP-Link AP based on IEEE 802.11ad, the link-layer cannot aggregate within the same block, frames from both retransmission and transmission queues. Therefore, the only type of traffic transiting in a link with pending retransmissions is the traffic generated from those retransmissions. Thus new information can only circulate after the pending packets are successfully delivered. In our experiments, the transmitted rate was observed to be no more than 60 Mbps for heavily blocked links, suggesting that the network mainly carries traffic from pending retransmissions instead of useful new information.

Furthermore, the experimental results show that, even under normal conditions, the TCP packet retransmissions still exist. According to [73], this is because the timeout value of frames that are waiting on the received queue is not adjusted to account with the time channel that was occupied performing frame transmissions of the IEEE 802.11ad mechanisms such as beamforming and beam tracking.

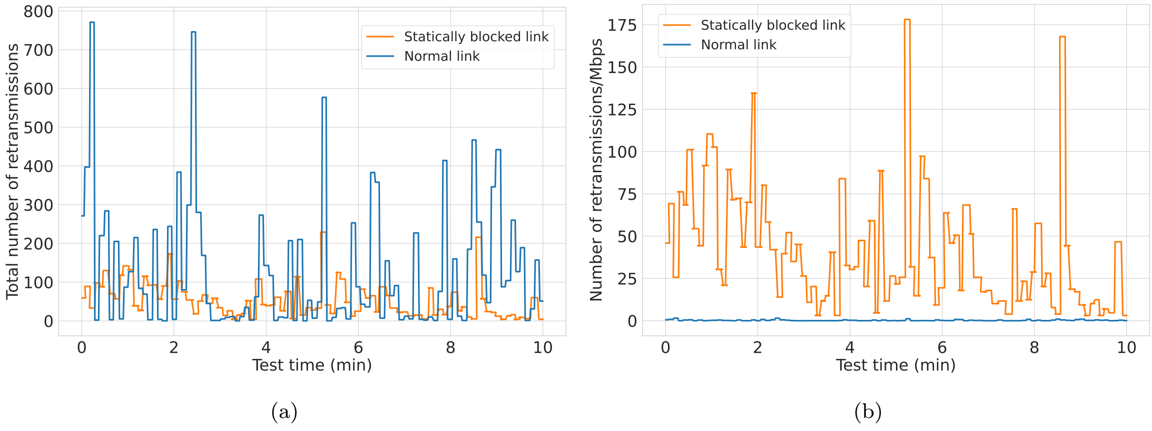
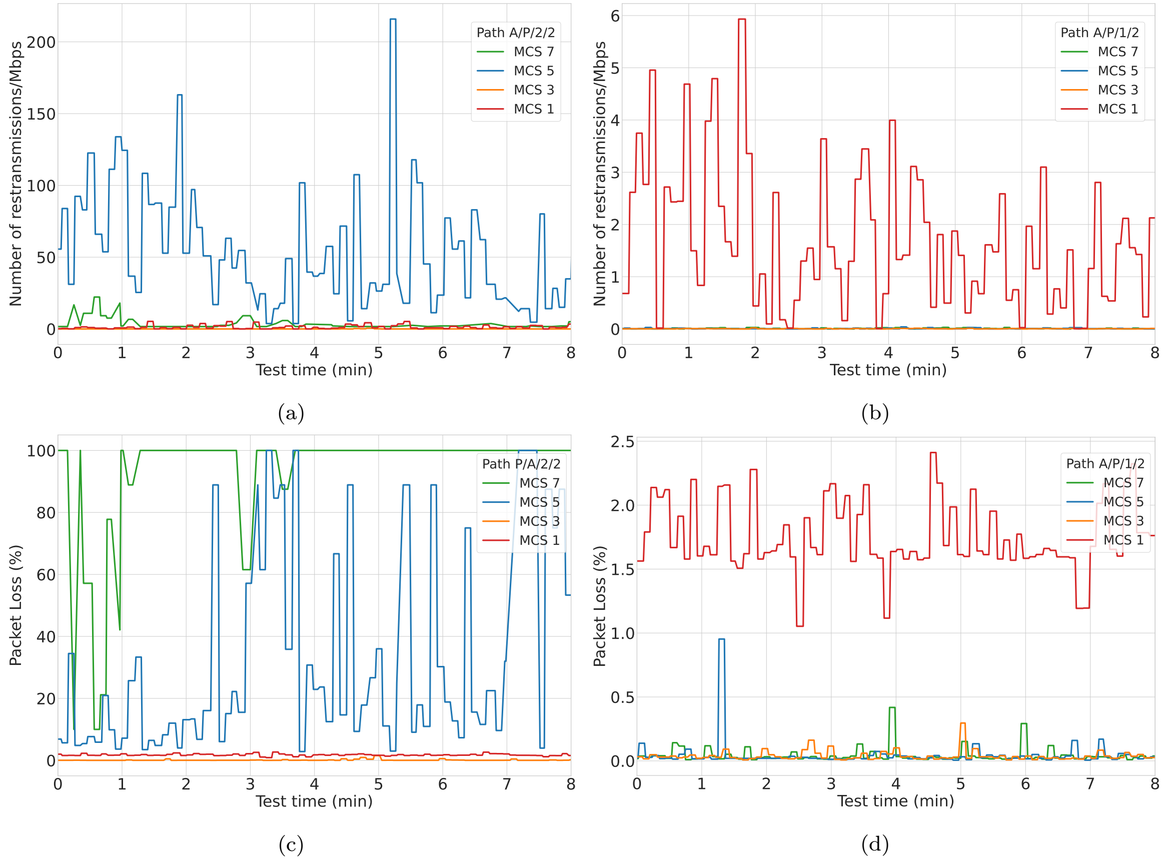


Figure 4.13. The total number of TCP retransmissions (a) and number of retransmission per Mbps (b), measured for the P/A/2/2 under long-term blockage when using a fixed MCS of 5.

Figure 4.14 shows the relationship between the normalized number of retransmissions and the PL for two links suffering from different levels of long-term obstruction. The P/A/2/2 link has higher SNR than the P/A/1/2 link in all fixed MCS tested, resulting in a more heavily blocked channel. In a highly obstructed link like P/A/2/2, using higher MCSs such as 5 and



(a)

(b)

(c)

(d)

Figure 4.14. Number of TCP retransmissions per Mbps measured during long-term blockage in the uplink A/P/2/2 (a) and A/P/1/2 (b) for five different MCS levels. Packet Loss measured for the uplink A/P/2/2 (c) and A/P/1/2 (d) for five fixed MCS levels

7 causes an average PL of 49% and 97%, respectively, and maximums that can reach the 100% mark (see table 4.5). The high PLs triggers the TCP retransmission mechanism. On the other hand, in a slightly blocked link similar to P/A/1/2, the number of retransmissions is kept very low for all tested MCSs. Note that, in the A/P/1/2 link case, the maximum retransmissions do not surpass the six mark, whereas in the heavily obstructed link, the value is 33 times higher. This is because A/P/1/2 was observed to comply with the RSSI and SNR requirements of MCS up to 7, which allowed it to operate with negligible PER for all MCS tested. In this case, long-term blockage's only consequence is the limitation of the maximum supported throughput because it only supports MCS 7.

Furthermore, describing the behavior of the TCP retransmissions on a link using an adaptive MCS scheme is a somewhat complex task, as an increase or decrease in the instantaneous value of this metric can be traced to multiple sources. Firstly, since the modulation scheme is dynamically adjusted according to the registered PER and SNR, it is expected that most of the time, a significant SNR decrease and/or high PER will cause the MCS to be changed to a more robust scheme. Nonetheless, suppose the code rate of the FEC mechanism used in the MCS prior to the PER peak can correct the errors introduced by obstruction. In that case, there are not as much losses in the channel, and thereby, the number of retransmissions should remain controlled. On the other hand, an increase in the TCP retransmissions could also be observed due to pending retransmissions from previous unsuccessful instances. Thus, one cannot conclude with certainty whether an increase or decrease in the number of retransmissions will be observed when the MCS is dynamically adjusted.

4.3 Summary

This chapter has characterized the impact of long-term and short-term blockage on a mmWave link (based on IEEE 802.11ad) at multiple layers (physical, network, and transport). The analysis conducted showed that, maintaining higher modulations in long-term blocked channels, might only be possible at the expense of high PERs (between 90% - 100%), a maximum delay that can be in the order of one or a few seconds, high PLs at the transport layer (between 37% - 100%), and a high number of TCP retransmissions which reduces the throughput rates to a few Mbps. Even though dynamically adjusting the MCS leads to more controlled PERs and PLs at the transport layer, it was still shown to reach maximum PERs of 40% even if the SNR remains constant in a long-term blocked link. Moreover, long-term blockage was shown to cause maximum delays of 0.9 seconds in some links, and data rates that vary in a few hundred Mbps.

Furthermore, it has been shown that short-term blocked links are more likely to reach maximum PERs when there are sudden decreases in signal quality and are, therefore, more likely to adapt their MCS to a more robust scheme. Still, the MCS returns to its pre-loss value as soon as the LOS condition is reestablished. Hence, short-term obstruction, not only decreases the maximum throughput at the layer-2, but also causes packet losses up to 8% that will lead to a significant decrease of the TCP congestion window, further contributing to the decrease in the data rate. Moreover, the temporary LOS blockage led to maximum delays of half a second in the worst-case scenario.

These aspects combined create a highly unreliable backhaul network, which comprises the constant data rate and maximum delay requirements of both TCP-based and UDP-based services. The lack of a retransmission mechanism in UDP to circumvent the losses introduced by the mmWave channel makes best-effort services based on this transport protocol, like VoIP or video transmission, suffer even more from reliability problems. Furthermore, the conclusions drawn at the TCP-based transport layer level indicate that even support for TCP-based services is compromised. The decrease of the TCP congestion window to values lower than those used during the pre-loss instants, alongside the inability of the transport layer to support new mechanisms implemented in the MAC layer of WiGig networks, makes it even more important to search for a solution that can deal with the propagation challenges in the 60GHz bands in a more proactive manner. Thus, to ensure a truly resilient network, mechanisms should be in place to mitigate the negative impact caused by LOS obstruction at multiple layers. The following chapters propose mechanisms to improve the network reliability and resiliency of a blockage-prone mmWave backhaul.

Chapter 5

Machine Learning Pipeline

Due to the high variability of the IEEE 802.11ad-based networks, a proactive ML-based resilience mechanism is proposed to eliminate the need for constant beamforming training, solving the network reliability issue. Using such approach requires the addressing of two ML problems: 1) prediction of the wireless link quality according to the current metrics; and 2) forecasting relevant KPIs based on past history.

This chapter describes the methodology to address these two ML problems, and the obtained results. It is organized in the following way. The first section describes the preprocessing pipeline used to produce the dataset supplied to the models. The second section describes the methods and techniques used to produce and evaluate a model that can solve the classification and forecasting problems. This last section is subdivided into two parts. The first subsection explores the impact of the time window selection in different classification methods, and compares the training and testing scores obtained with the different classifiers (ML and DL-based). The second and last subsection presents a comparative study of methodologies commonly used in the literature for time-series forecasting. More specifically, it compares the quality of the forecasts made by a statistical-based method like ARIMA with those made by a LSTM-based RNN for a dataset collected from the testbed.

5.1 General Pipeline

Typically, running a machine learning algorithm involves a sequence of steps, including data preprocessing, model fitting, and a model validation stage. These stages are usually encapsulated within a ML pipeline, which formalizes the processes and data transformations required to produce and deploy a functional and well-scored machine learning model.

Although solving the classification and forecasting problems requires different tools, the input data used is the same. Therefore, both tasks share a common ML pipeline, which is illustrated in Figure 5.1. This pipeline is composed of two main components: the preprocessing module and the ML module. The preprocessing module comprises the workflow necessary to transform the raw physical and network data into a new format understandable by most ML models. This stage involves data cleaning, data aggregation, data transformation, and feature selection. On the other hand, the ML module is responsible for all the processes that allow fitting, testing, and validating the developed DL and ML models. This step comprises hyperparameter tuning to select the settings with the best performance, test and validation, and inversion of the transformations made to the data during the preprocessing phase. The only difference between the classification and the KPIs forecasting tasks is related to the models implemented in this latter module, as they require completely different outputs. This module is also responsible for formatting the input data to fit a specific model's shape.

5.2 Data preprocessing module

This section describes the data preprocessing module and its different phases: data cleaning, aggregation, transformation and feature selection.

5.2.1 Data cleaning

The format of the raw metrics gathered in chapter 3 is not understandable as-is by ML algorithms. Data-specific characteristics, such as missing values and duplicate entries, make the work with a handful of popular classification and forecasting algorithms very difficult. To address this, the multi-layer metrics were submitted to a cleaning process, where invalid and duplicated entries were removed, and where the missing values on each entry row were replaced, using time-series interpolation techniques. By not simply removing invalid entries, we are able to preserve as much information as possible. After this process has ended, the raw data from physical, network and transport layers was ready to be aggregated.

5.2.2 Data aggregation

The aggregation process accepts the physical and network data sources, and transforms them into a single dataset. While each physical data metrics is collected with a 1-second sampling frequency, network-related entries were only available every 5 seconds. Thus, direct data aggregation of those two sources was not possible. Instead, the network data was firstly up-sampled to match the 1-second periodicity observed in the physical data, a process which by itself generates missing values. To address this, backfilling methods were used in the up-sampled data source to replace each missing value with the previously observed non-null value.

Only when both datasets shared the same temporal axis, the actual aggregation process could begin. During this stage, the network and physical metrics with the same time indexes were concatenated, thereby making the number of features per entry in the resulting dataset equal to the sum of the features in each dataset type. The data related to each mmWave path

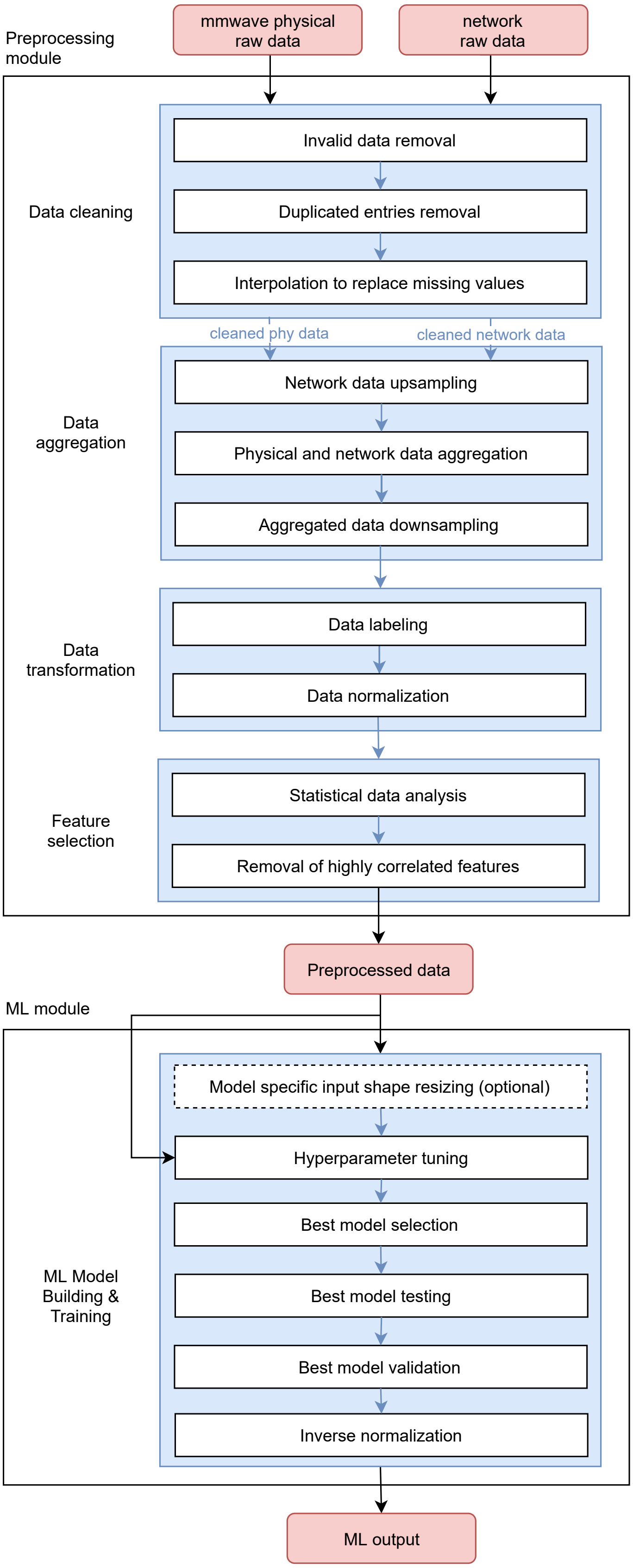


Figure 5.1. ML development pipeline.

and scenario was then grouped into windows of 3, 5, 7, and 10 seconds to form a single entry representing the link state in that time frame. To compute the value of each of the 25 metrics in that time window, aggregation functions expressed in table 5.1 were used.

Table 5.1. Aggregation functions used to produce a single metric for each 1 second entry.

Feature	**Aggregation Function**
Path ID	first()
Timestamp	
Beam Index TX	mean()
Beam Index RX	
Beam Index TX (Min)	
Beam Index TX (Max)	
Beam Index RX (Min)	
Beam Index RX (Max)	
Layer-2 Data Rate	
Layer-2 Data Rate (Max)	
TCP throughput	
Sent Data Rate	
RSSI (dBm)	
SNR (dB)	min()
MCS TX	
MCS RX	
No. of Packet Transmitted	
No. of Packets Received	
Packet Error (%)	max()
TCP Retransmissions	
Packet Loss Count	
Packet Loss Rate	
Packet Duplicate Count	
Packet Duplicate Rate	
RTT (Min)	
RTT (Mean)	
RTT (Max)	

The features that were aggregated using the first() function were set with the value of the first data entry of their respective time window, as it consists of only identifier labels; while features using either mean(), min(), or max() methods were set to be equal to the mean, minimum and maximum values registered in that time window, respectively.

5.2.3 Data transformation

The data transformation stage is responsible for transforming the data aggregated from both sources in a format more suitable for ML algorithms. Data transformations techniques include methods for numeric data transformation such as normalization, removing noisy data such as smoothing, and methods for data discretization. Smoothing data methods are typically used to remove the noise from the dataset by detecting trends. However, applying such tools in our dataset would lead to an inevitable loss of information. The large fluctuations observed, particularly in short-term blocked links, would be treated as noise, and thus, would be removed from the dataset. Under this work scope, the only viable data transformation method is feature scaling/data normalization.

The dataset generated from the aggregation process is composed of several features with different units and ranges. For instance, the transmission MCS can range from 1 to 9, while TCP retransmissions can range from 0 to 1000, as shown in Chapter 4. Furthermore, the acceptable RCPIs are contained in the negative -84 to -59 dBm range. Providing these features as-is to a ML model could lead to a bad score, since distance-based models would assign larger coefficients to features with larger values. In order to ensure that features are all given the same importance, feature scaling was performed via a *min-max* scaler. This scaling operation normalizes each feature to be contained in the range between 0 and 1, converting the data into a more ML friendly-format. This approach was chosen because it does not change the shape of the original distribution, nor reduces the importance of outliers (as sudden signal quality drops would be considered outliers), as other types of scaling do. Although this is not a required step for every model, some ML and DL methods such as SVM, logistic and linear regression, and ANNs require the input to be normalized in order to provide good accuracy [109]. Moreover, scaling the features lead to a faster gradient descent algorithm convergence to a global or local optimum in ANNs, as there is a better weight initialization [110].

Once the data was scaled, the resulting dataframe was submitted to a labeling process with a tag indicating the scenario, in which each entry was collected and added. The tag assignment was performed by correlating the timestamps of each entry with the start and end timestamps registered during the data collection process, as it was explained in section 3.2.2 of Chapter 3. Each data entry was labeled according to the experience; it belongs to one of three possible labels: "0", "1" and "2", which specifies if the link is under normal operation, suffering from short-term blockage or long-term blockage, respectively. Note that these labels are only necessary to solve the link quality classification problem, and therefore, they must be removed from the preprocessed dataset, before it is supplied to a forecasting method.

5.2.4 Feature selection

The aggregation of physical and network data generates a dataset with a total of 25 useful features, which already passed a series of processes designed to improve them. In an attempt to speed up the development process, one may be tempted to provide this information directly to a ML model without any additional consideration. However, passing all these metrics without regard to what is actually there, increases the computational effort excessively beyond what is necessary, and in some cases, may also negatively impact the model performance. Redundant and irrelevant metrics were removed from the aggregated dataset, through a feature selection process to ensure that this would not happen. Feature selection aims to reduce the number of features, by maintaining only those which contribute the most for the expected output. Less redundant data means that, there is less chance to make decisions based on irrelevant metrics and less data to be processed, which helps reduce both overfitting and training time.

Similarly to ML algorithms, the feature selection techniques can be classified as either supervised or unsupervised techniques. In supervised methods, the target variable is used to identify the features which contribute the most for a correct prediction. In contrast, unsupervised methods only use the statistical nature of the input variables, to decide which features are dispensable and which ones are not. This work employs a correlation-based method that belongs to the latter category. This method uses the calculated Pearson correlation coefficient for each feature pair, to identify and remove highly correlated input variables. For that purpose, an absolute threshold of 0.95 was established, and metrics correlated with other input variables by an absolute coefficient equal or greater than 0.95 were removed. This correlation threshold could have been set to a lower value. However, that may remove features that might be important, in correctly predicting the output variable under intermittent blockage, thus potentially decreasing the model's performance. Setting higher correlation thresholds lowers the chance of that happening, as only extremely correlated variables (that existing ones can predict) are removed.

In order to discover which features should be eliminated, the correlation matrices were computed for each physical link. Figure 5.2 and 5.3 show the correlation between input variables for links P-A and P-B, respectively. Note that metrics can either be negatively or positively correlated with another variable. The value is positive, if the increase in one variable relates to an increase in the other. It is a negative value, if the increase in one variable leads to a decrease in the other. The analysis of the Pearson's pair correlations for both links show that:

- The receiving and transmitting beam indexes are highly correlated with their maximum and minimum values;
- Features that were calculated based on other features are also highly correlated with one another. This is the case of the packet loss percentage metric, that is calculated as the ratio of the number of transmitted packets to the number of received packets; and the RTT mean deviation, that is calculated using the average RTT and its maximum value.

In addition, features that remained constant for the entire test duration were also removed, as they did not provide any useful information. Hence, the features that were effectively

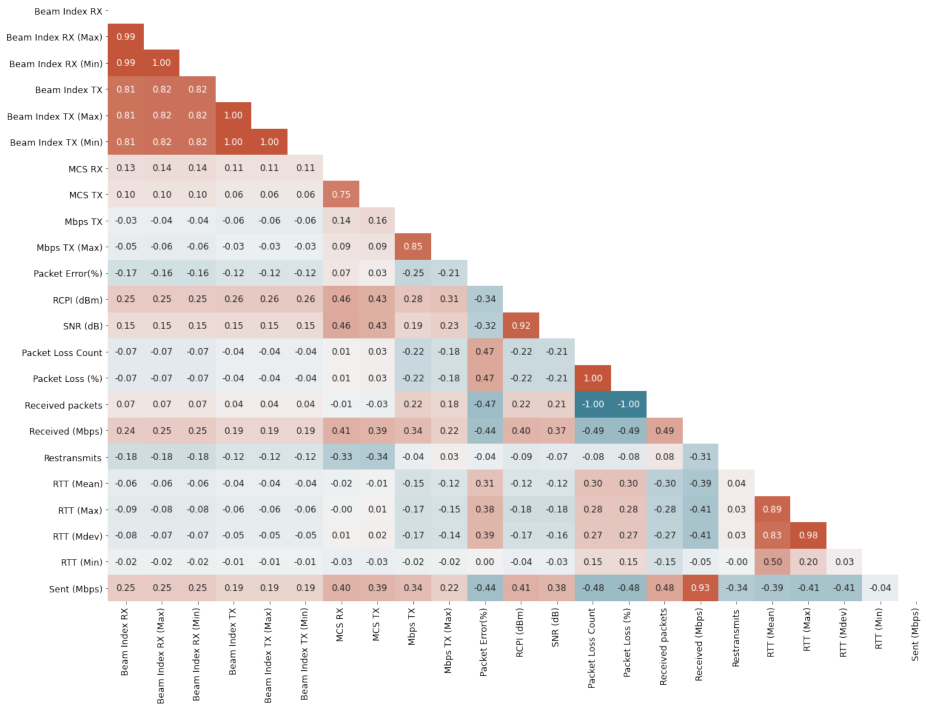


Figure 5.2. Correlation matrix obtained with the P-A link for all experiments.

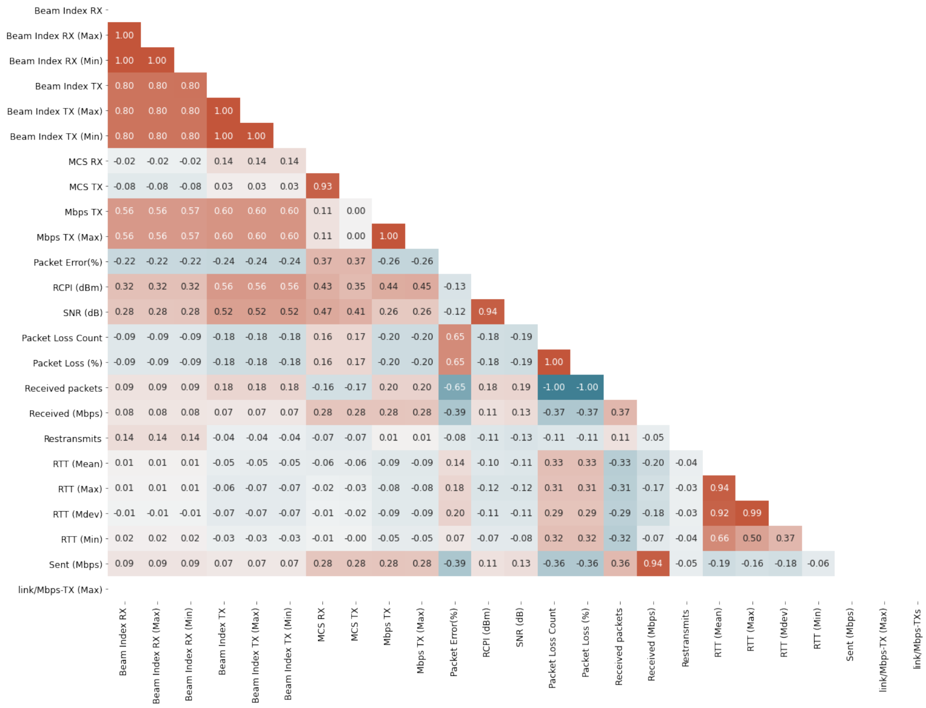


Figure 5.3. Correlation matrix obtained with the P-B link for all experiments.

removed from the final dataset were:

- The minimum and maximum transmitting and receiving beam indexes: i.e., *Beam Index TX (Min)*, *Beam Index RX (Min)*, *Beam Index TX (Max)* and *Beam Index RX (Max)*;
- The total number of transmitted and received packets and packet loss count;
- Packet duplicate count and duplicate rate since their values remained constant.

Once the redundant information was removed from the final dataset, the data was finally ready to be used by the models in the ML module.

5.3 ML module

The ML module uses the preprocessed data to produce and evaluate a model that solves each ML task (classification and forecasting). Still, allocating the entire dataset to the model training is not recommended, as there would be no way to evaluate whether a model is underfitting or overfitting. One way to overcome this is to split the data into training and testing datasets to train and test the model on different data. Testing on unseen data provides insight into how well-generalized the model is.

In this work, the data splitting process was performed randomly for all classifiers tested, as well as for the forecasting ANN, using a 70:30 ratio. That is, 70% of the data is allocated for training the model, and the remaining 30% is used to evaluate its performance. Because the data fed to ML and DL algorithms is already aggregated into time windows, temporal patterns are maintained within a specific time frame. However, in the case of a statistical-based model like ARIMA, the splitting process must maintain the temporal order of samples, so that the model can predict the target variable. After splitting the dataset into two parts, the hyperparameter tuning process is initiated to find the best-performing hyperparameter settings for both ML and DL algorithms. Similarly, a search is done for the combination of parameters p, d, and q that allowed ARIMA to make effective predictions. At the last step, the best-performing model found during the random search is submitted to a final unbiased evaluation.

The next two sections detail the methodology used to develop models that: 1) classify the wireless link quality, and 2) forecast the link's main KPIs.

5.3.1 Link Quality Classifier

The process of classifying a radio link quality according to three types of scenarios can be classified as a multi-class classification problem. In machine learning, multi-class classification refers to the task of predicting a class label for each input's data sample. Under this work context, it means assigning one of three labels ("normal link", "short-term blocked link", or "long-term blocked link") for each sample. As previously mentioned, the preprocessed data had a 1-second periodicity. However, predicting a class label for a 1-second entry has no interest. Each sample would only contain information regarding each instantaneous KPI, rather than reflect the temporal variations induced in different scenarios. Even if that were not the case, it would be very challenging to perform any traffic redirection action in that short period of time.

Similarly to what happens during the acquisition of most real datasets, it was impossible to acquire an equal number of samples for the three classes. Note that it would be unrealistic to have more than five consecutive minutes of data for each path under a short-term blockage scenario, since the nature of the test itself would require a person to walk back and forth for a very long time. Hence, the generated dataset is imbalanced, as more samples characterize more non-blocked or long-term blocked links than intermittently blocked ones. To correctly evaluate the performance of a ML algorithm that uses unbalanced datasets, it is necessary to use evaluation metrics that, contrary to accuracy, severely penalize misclassification in the minority class' samples such as precision, recall, or F1-score. In this work, the precision, recall, and F1-score values obtained for each class were macro averaged to produce a single-precision value, that reflects the performance of both majority and minority classes.

Both ML and DL approaches were tested to perform wireless link quality classification. According to the exploratory data analysis (EDA) presented in chapter 4, the data points belonging to each scenario are, for the most part, not linearly separable. This is also shown by the superimposed data points agglomerations of figure 5.4, that represents the relationship between each feature pair. Therefore, only algorithms that supported non-linear mapping were developed to compare the performance, which include Support Vector Machine (SVM), Random Forest (RF) and Multi-layer Perceptron (MLP). Since F1-score provides a better measurement of incorrectly classified cases than accuracy in imbalanced datasets, it was the metric chosen to select the best-performing models during hyperparameter tuning, and to evaluate the performance of the different proposed classification models. Also, considering that there was no previous knowledge regarding the window size that provided the best predictions, all algorithms were tested for time windows of size 3, 5, 7, and 10 seconds. The implementations of a Support Vector Machine (SVM), Random Forest (RF) and Multi-layer Perceptron (MLP) classifiers and the results for each window size are now going to be described.

One way to perform hyperparameter tuning is using the popular $scikit'slearn$ python library. This was the library chosen to find the best settings for each classifier tested through k-fold Cross-validation (CV). Under k-fold CV, the dataset is split into k smaller sets, where k-1 folds were used to train the model, and the remaining fold to evaluate if the model is overfitting or underfitting on unseen data. This approach gave a chance for each dataset sample to appear in both test and training sets. The evaluation metrics obtained for each fold were then averaged and compared with the results obtained with other hyperparameters settings to select the best well-performing model for all k folds runs.

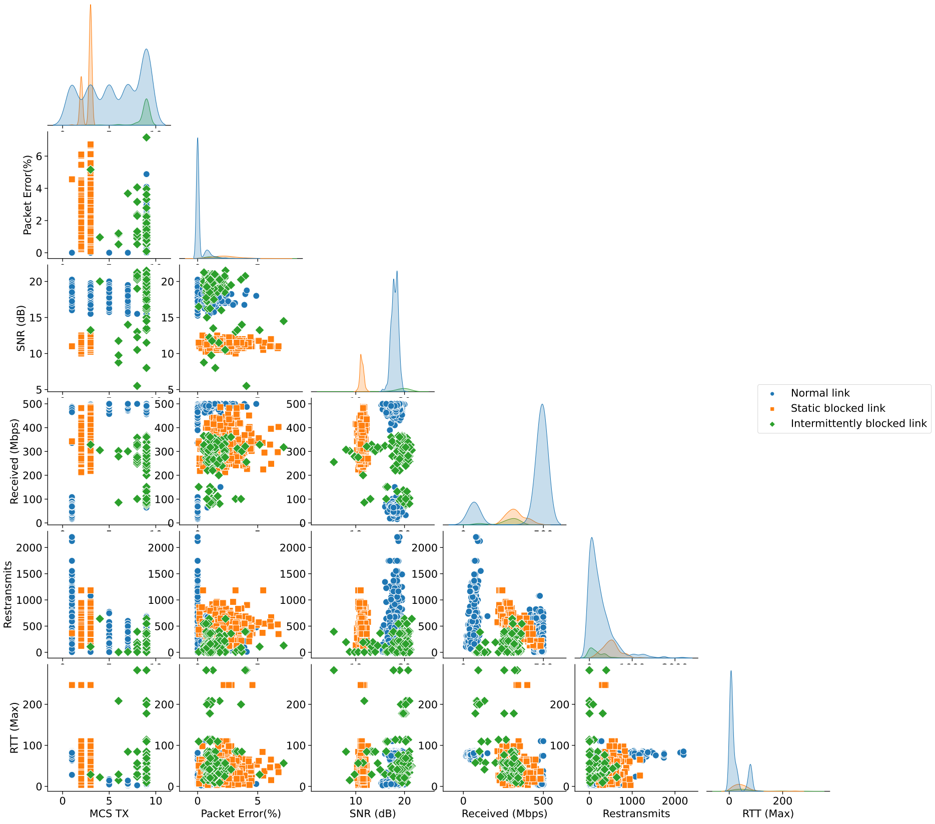


Figure 5.4. Pairwise relationships observed for the uplink P/A/2/2 working in the automatic MCS mode.

Support Vector Machine

SVM is a supervised machine learning method mainly used to solve classification problems, though it can also be used for regression tasks. This algorithm works by finding the optimal decision boundary, commonly called hyper-plane, which best distinguishes the different classes. Instead of searching for the characteristics that differentiate all classes, SVM methods work by finding similarities between samples belonging to the same scenario. In this work, the SVC (C-Support Vector Classification) class was tested using the *scikit learn* python library. This method supports multi-class classification problems by employing a one-vs-one approach in which the original problem is subdivided into multiple smaller binary classification problems, one for each class pair. For instance, a binary classification problem will be generated to classify if a link is non-blocked or statically blocked, and another one to classify if that link is in the non-blocked state or intermittently blocked state. Since the method's input comprises a total of 16 metrics, the best hyper-plane is the 16-dimensional vector, whose distance to the nearest data points (i.e., support vectors) is the largest. However, increasing the distance between the decision boundary and classes is only possible at the expense of more sample misclassifications during training. This makes finding a good balance between the margin distance and the number of allowed misclassifications, perhaps the most critical task during model development. Ensuring that a model is both well-generalized and well-performing often involves allowing more error to be present during training in an attempt to reduce over-fitting.

To model such dependencies, SVM computes the Hinge Loss, a cost function that incorporates the margin distance from the decision boundary in the cost computation. Hinge loss penalizes misclassified samples, and correctly classified ones that do not have a sufficiently large margin from the decision boundary. Keeping this in mind, several model-specific hyperparameters need to be tuned to find the best performing model, including [111]:

- **C - Regularization parameter:** Controls the trade-off between the decision boundary and the number of misclassifications allowed. The smaller the C parameter is, the more tolerant the algorithm is towards misclassifications, and generally, larger margins are achieved. The larger the C parameter is, the more likely the model is to over-fit as the SVM will be more susceptible to noise, resulting in lower margins;
- **Kernel:** The kernel functions are used to transform the data points from a low dimensional space to a higher-dimensional space. Commonly referred to as "kernel tricks", kernel functions are used when the input data is not linearly separable;
- **Gamma:** It does not exist for linear kernels. It is the inverse of the radius of influence of training samples selected by the model as support vectors. A large gamma means that only points in close vicinity influence the computation of the decision-boundary. In contrast, a small gamma is associated with a large area of influence, and thus far away data points are also used to compute the optimal hyper-plane.

The *scikit's learn* random search method was used to build and evaluate models using different combinations of the parameters expressed in table 5.2, by performing k-fold cross-validation. In this case, k was set to 3, and the combination with the highest average F1-score for all *k* runs was selected. Table B.14 of the appendices presents the configurations used by the top three models discovered, during hyperparameter tuning for time windows of 3, 5, 7, and 10 seconds of aggregated data.

Table 5.2. Hyperparameter settings used during random grid search for SVM model.

Kernel	C	Gamma	Degree
Linear	[1, 100] step = 500	–	–
Gaussian		$[10^{-4}, 10^{2}]$ step = 500	
Sigmoid			
Polynomial			2, 3, 5 and 7

Note that the decision boundaries found for the best model are not presented in this document, due to the complexity of visualizing high-dimensional data. In an attempt to solve this, the 16-dimensional data was projected onto a 2D plane using the Principal Component Analysis (PCA). However, reducing dimensionality generated a cluster of points even more dispersed than the one observed in figure 5.4 for each feature pair. Thus, a reliable representation was not being made for higher dimensions. In the N-dimensional space, it is possible to effectively separate the points belonging to the different classes using non-linear kernels, since high test scores were obtained.

Table 5.3 shows the train and test F1-scores obtained using 3-second time windows with some of the SVM models tested during grid search CV with the Radial Basis Function (RBF), sigmoid and polynomial kernel functions. The models presented in this table can be classified as belonging to one of two levels (Best and Worst) according to the performance they achieved. A careful analysis of the results shows that for the RBF kernel, and with the range of C parameters tested, the gamma parameter has a more significant influence on the model's performance than the regularization parameter. This means that, regardless of whether the value of C tested was 1 or 100, choosing an appropriate gamma value will lead to a good performing model. Higher gamma values allow the hyper-plane computation to only consider more densely concentrated point areas, thus ignoring the "outliers" for each class, which explains better training and testing scores. Still, the testing results showed that the combinations tested did not suffer from any visible overfitting problem, since both training and testing scores were similar, differing no more than 2%. The same relationship seems to be established for polynomial kernels, since for a varied range of parameters C and degree of the polynomial, gammas smaller than 0.001 show test F1-scores of 0.22. Conversely, the best performance is achieved with lower gammas in the case of the sigmoid kernel.

As table 5.4 shows, the RBF kernel achieved the best results for all tested window sizes (1, 3, 5, 7 and 10 seconds). Because the data points in all tested window sizes share the same properties (i.e., are constructed in the same manner), they are all not linearly separable. The polynomial kernel also appears as a close third for some window sizes. It was observed that, generally, both kernels could achieve a good test score when properly configured. In contrast, the sigmoid kernel was shown to always lead to lower performances. For instance, the maximum F1-score registered during the hyperparameter tuning process using that kernel was about 0.69. The results obtained for the top three models are presented with more detail in table B.14 of the appendices.

Note that a SVM model that uses a RBF kernel tries to maximize the euclidean distance between the hyperparameter plane and the support vectors. Therefore, if the different features were passed with their original ranges, that would lead to large features dominating over the others in the distance computation. Scaling features to the same range, prior to feeding them to the model, ensures that they all have the same influence in the computation and, thus, the same importance.

Table 5.3. F1-score obtained with cross-validation of some of the models evaluated during grid search for the 3-second window using the RBF, sigmoid and polynomial kernels.

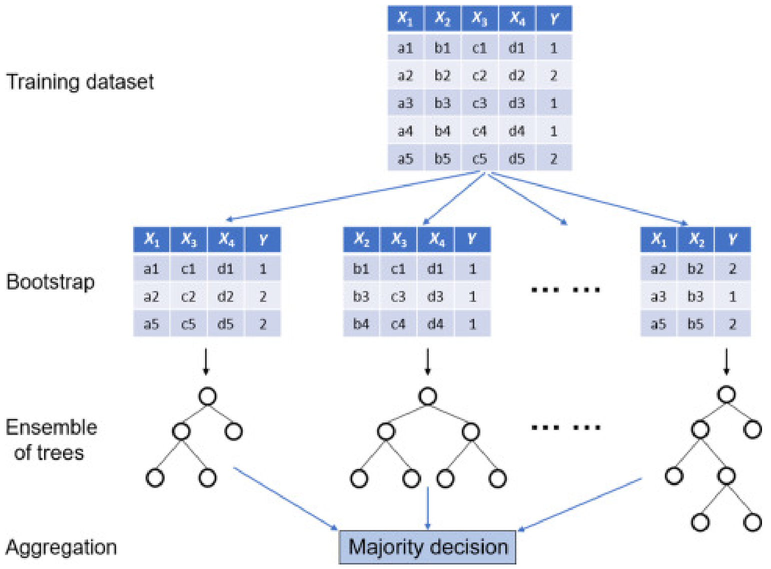
		Parameters			**Results**	
Kernel	**Performance Descriptor**	**C**	**Gamma**	**Degree**	**F1 Train**	**F1 Test**
RBF	Best	81.94	3.81	-	0.99	0.97
		14.94	7.41	-	0.99	0.97
		10.71	3.71	-	0.98	0.97
	Worst	34.60	0.001	-	0.64	0.64
		40.29	0.0004	-	0.56	0.56
		30.69	0.0003	-	0.56	0.56
Sigmoid	Best	57.48	0.009	-	0.71	0.70
		40.67	0.005	-	0.70	0.70
		31.12	0.017	-	0.70	0.70
	Worst	15.57	3.15	-	0.39	0.40
		19.71	6.02	-	0.36	0.36
		56.17	0.69	-	0.32	0.31
Poly	Best	12.08	4.89	3	0.98	0.97
		30.41	0.51	7	0.98	0.96
		36.23	4.36	5	0.98	0.94
	Worst	14.33	0.001	7	0.22	0.22
		10.19	0.0002	2	0.22	0.22
		33.50	0.0004	7	0.22	0.22

Table 5.4. Best settings found during the tuning process for the SVM method for using time windows with size 3, 5, 7 and 10 seconds.

Window Size	**Best Hyperparameters**	**Mean fit time**	**F1-score Macro**	
			Train	**Test**
3 s	kernel: rbf gamma: 3.381 C: 81.249	3.3 s	0.988 ± 0.001	0.971 ± 0.002
5 s	kernel: rbf gamma: 5.122 C 18.051	1.95 s	0.987 ± 0.002	0.967 ± 0.006
7 s	kernel: rbf gamma: 3.460 C: 93.744	0.76 s	0.993 ± 0.002	0.971 ± 0.002
10 s	kernel: rbf gamma: 4.161 C: 12.889	0.695	0.990 ± 0.001	0.968 ± 0.007

Random Forest

The Random Forest (RF) is a supervised machine learning algorithm that uses an ensemble of decision trees to solve complex regression and classification problems. Each individual classifier's prediction is used in this ensemble to find the class with the most votes, which will be the final model's class prediction. In most scenarios, this method produces good results, even without hyperparameter tuning in non-linear problems such as ours. The secret lies in the employment of many uncorrelated classifiers that outperform any of its single constituents. To ensure that individual classifiers are in fact not correlated, bagging (i.e., bootstrap followed by aggregation) is used while growing the trees. In this method, which is graphically represented in figure 5.5 [112], each tree is grown in parallel using a new dataset generated from the original dataset, using only a random subset of features from the entire feature set. In this case, instead of splitting the data into smaller portions that are supplied to each classifier, the new datasets are created by randomly sampling with replacement the original values. Thus, each tree is trained with the same number of samples of the initial dataset [113].



X_1	X_2	X_3	X_4	Y
a1	b1	c1	d1	1
a2	b2	c2	d2	2
a3	b3	c3	d3	1
a4	b4	c4	d4	1
a5	b5	c5	d5	2

X_1	X_3	X_4	Y
a1	c1	d1	1
a2	c2	d2	2
a5	c5	d5	2

X_2	X_3	X_4	Y
b1	c1	d1	1
b3	c3	d3	1
b4	c4	d4	1

X_1	X_2	Y
a2	b2	2
a3	b3	1
a5	b5	2

Figure 5.5. Implementation of RF classifier on a dataset that has four features (X_1, X_2, X_3, and X_4) and two classes (Y = 1 and 2). RF classifier is an ensemble method that trains several decision trees in parallel with bootstrapping followed by aggregation. Each tree is trained on different subsets of training sample and features [112].

As shown in figure 5.5, a decision tree is composed of a series of nodes that start at the root node, and that extend to various leaf nodes, representing classes to classify. A tree is grown by iteratively splitting a node into two or more nodes according to some decision criterion defined by the specific decision tree algorithm. In this work, the *scikit's learn* random forest algorithm employs an optimized version of the Classification and Regression Trees (CART) algorithm. In brief, a node is split by selecting from a random subset of features of the input variable, a single variable and its appropriate cut-off boundary, that leads to increased homogeneity in the data samples remaining after each split. The trees are grown until the first stopping criteria is met, which in this work can either be the maximum depth of the tree, the minimum number of samples necessary to split an internal node, or the minimum number of samples required in each leaf node [114]. Similar to SVM, k-fold cross-validation was performed to search for the optimal hyperparameters. The hyperparameters setting ranges used are presented in table 5.5. The number of features to consider when looking for the best split was set to automatic mode (i.e., $max_{features} = \sqrt{17}$), or an integer value in the range of 1 to 17 (i.e., maximum of available input variables). In order to compare the efficiency of both approaches, the model was tested using both bootstrap and no bootstrap approaches. When bootstrapping was not active, the whole dataset was used to build each tree, as previously explained.

Table 5.5. Hyperparameter settings used during random grid search for random forest assemble.

No. of estimators	[100, 5000] with step=1000
Max. no. of features	'auto', 7, 7, 10, 12, 15 and 17
Max. tree depth	[10, 1000] with step=100
Min samples per split	2, 5, 10, 16, 64, 10, 150 and 256
Min. samples per leaf	1, 2, 4, 50, 64, 100, 128, 150 and 256
Bootstrap flag	True and False

Unlike other classifiers, with RF it is very easy to measure the relative importance of each feature on the prediction, by finding the variables that caused the most significant impurity reduction in the nodes, that used that feature as a decision variable. This information is available after training, and thus can be accessed. Figure 5.6 shows the computed feature importance observed for the 3-second time window dataset.

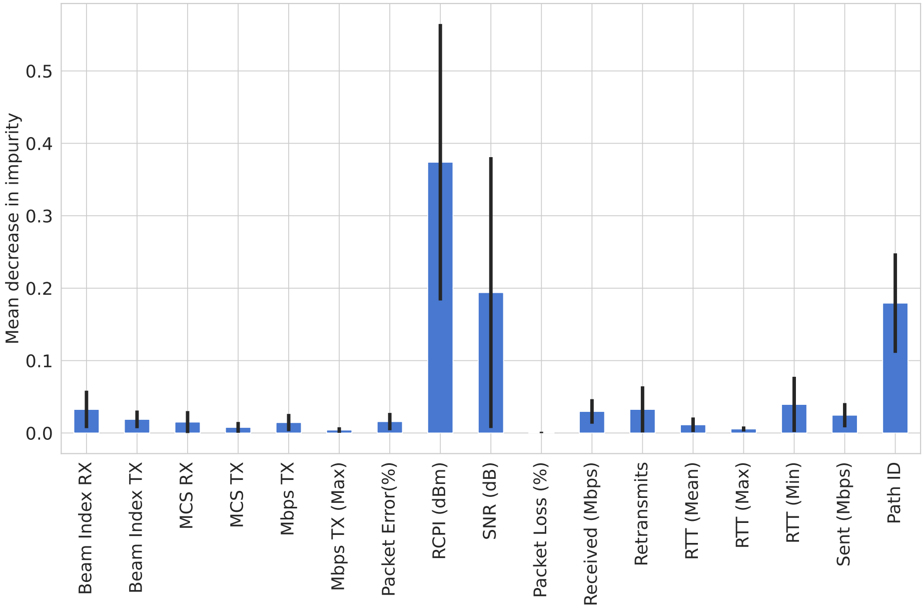


Figure 5.6. Feature importance using the mean decrease in impurity (MDI).

Table B.15 in the appendices displays the results obtained with the three best performing models found in the hyperparameter search. Similar to the SVM model, the best performing models obtained for the 3, 5, 7, and 10-second datasets achieved similar performances. Moreover, it can be seen that, with and without bootstrap, similar test scores are achieved. Also, it was found that the number of estimators does not have to be large to get good results. Both models with few (e.g., 100) and many (e.g., 4000) estimators can perform well, as long as the hyperparameters are correctly adjusted. However, increasing the number of estimators has the downside of significantly increasing training times. Therefore, models that use a lower number of estimators must be preferred in the final comparison. The results obtained with the best performing model for each window size are summarized in table 5.6.

Table 5.6. The best model found during the tuning process for the RF method with 3, 5, 7 and 10-seconds time windows.

Window Size	Fit Time	Hyperparameters							F1-Score	
		Warm Start	**No. estimators**	**Min samples per leaf**	**Max features**	**Min samples per split**	**Max dept**	**Bootstrap Flag**	**Train**	**Test**
3 s	14.6 s	True	109	1	7	16	340	False	0.980	0.955
5 s	1.13 min	False	816	2	12	10	840	True	0.969	0.949
7 s	1.14 min	False	1046	10	15	9	470	True	0.947	0.954
10 s	46.9 s	False	1046	9	15	10	470	True	0.938	0.953

Figure 5.7 shows part of the process used to grow the 50th decision tree of the best RF configuration. As the figure illustrates, each tree is built by further subdividing the original dataset at each decision node based on the specific decision boundary computed. Whenever a decision node is able to fully separate the samples belonging to the different classes, it originates leaf nodes (one per class) that end that specific branch.

Although data normalization does not affect the RF's output because it is not a distance-based algorithm, its employment may decrease the computational effort.

Multi-layer Perceptron

A Multi-layer Perceptron (MLP) is a class of feed-forward neural networks. It comprises three types of layers: the input layer, an arbitrary number of hidden layers, and an output layer. These layers' main building block is an element called a neuron. A neuron is a computational unit that receives weighted inputs and produces an output by combining them according to a specific activation function. By employing non-linear relationships, neurons can combine their inputs in more complex ways. In principle, they can learn any mapping function between the inputs and the desired outputs if properly configured [115].

The first layer that appears in the multi-layer structure of a MLP is an input layer, also called the visible layer. This layer is responsible for the input data signal processing, and it is composed of a number of neurons equal to the number of independent variables. In

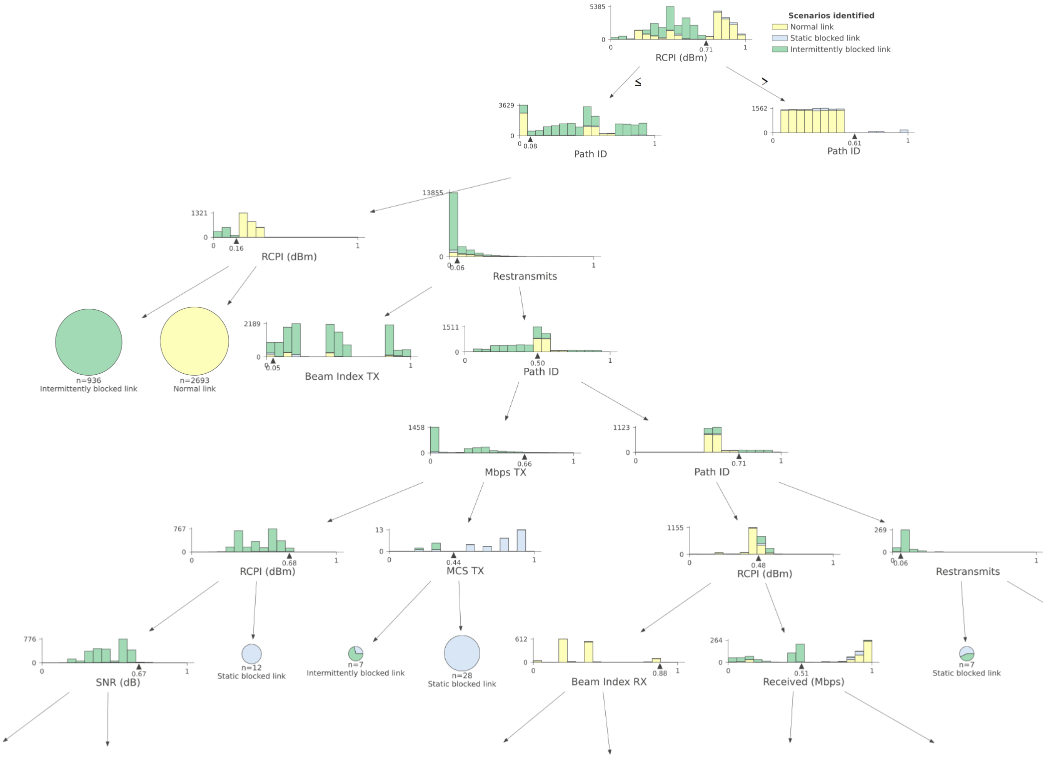


Figure 5.7. Branch of a decision tree used by the best random forest model.

this case, there are 17 preprocessing units. The input layer's neurons forward the inputs to the subsequent layers, rather than performing any non-linear modeling. The last layer that appears in the structure is the output layer. This layer is responsible for effectively performing the classification task at hand [49]. The choice of activation function is only dependent on the problem type: regression problem, binary classification, multi-class classification. Given that the problem of classifying the wireless link quality falls under the multi-class classification, there was the need of having three outputs (one per class), and thus, the $Softmax$ function was used. $Softmax$ computes the probability of a given input sample belonging to each class, and outputs the class with the highest probability as the model's final prediction [116]. Finally, the layers that are placed between the input and output layers are called hidden layers. These are the true computational engine of MLP, as they are responsible for properly modeling non linearly separable data, similar to the task in hand in this work. This is possible due to the ability of MLPs to, similarly to SVMs, find shapes/decision boundaries that allow separating classes in high-dimensional spaces [49]. The general settings used during hyperparameter tuning for the MLP Classifier are presented in table 5.7.

Because MLP neural networks use fully-connected layers, they are more likely to suffer from overfitting. In an attempt to reduce this, an L2 regularization term was added to the list of hyperparameter settings to optimize during random search. L2 regularization decreases

Table 5.7. General hyperparameter settings used for MLP Classifier with three different solvers.

	Hyperparameters	Values
Model	Input Layer Size	17
	Number of hidden layers	1, 2, 3
	Neurons	2, 4, 8, 16, 32, 64, 100, 128, 256
	Output Layer Size	1
	Batch Size	Auto, 2, 4, 8, 16, 32, 64, 128, 256, 512
	Activation Function	ReLu, Tanh, Logistic
	L2 regularization coefficient	0, 0.00001, 0.0001, 0.01, 0.1, 0.2, 0.3
Model Compile	Optimizers	Adam, Sigmoid, L-BFGS
	Losses	Cross-entropy

the likelihood of overfitting, by adding a penalty term in the cost function used during the weight updates, so that the algorithm favors smaller weights [57].

Since we did not know which weight optimization algorithm would be better for our data, all possible solvers were considered as an option: the Limited-memory Broyden–Fletcher -Goldfarb–Shannon (L-BFGS), an optimizer belonging to the family of quasi-Newton methods; the sigmoid method, the classical stochastic gradient descent algorithm; and the adam solver, a stochastic gradient-based optimizer proposed by Kingma, Diederik, and Jimmy Ba [117].

The solver-specific hyperparameters settings tested are also presented in table 5.8. In the gradient descent-based solvers, such as *Adam* and *Sigmoid*, the learning rate controls how much the weights are updated during training. Finding an optimal value for the step-wise weight update is crucial, as significant learning rates may result in unstable models, and low learning rates have a large computational effort. For the *Adam* solver, only the initial learning rate value and the number of total epochs was tuned. There was also an option for the sigmoid solver to adjust the learning rate schedule, i.e., how the learning rate should vary along the training process. For instance, a constant learning rate schedule meant that the value remained unchanged through the training process, while in an *Invscaling* schedule, the learning rate decreases at each time step t. For the L-BFGS solver, the only solver-specific configuration that could be optimized was the maximum number of gradient steps. The best performing neural network found for the MLP Classifier with these settings are presented in figure 5.9. Table B.16 of the appendices presents the results obtained with the top three settings found during hyperparameter tuning.

In our initial analysis, the results obtained for all time window sizes tested showed that the best results were achieved using the L-BFGS solver. It was suspected that it may have outperformed the typical gradient descent algorithm chosen, because it is a second-order optimization algorithm and not a first-order method such as gradient descent. Note that, besides using the gradient for performing weight updates, L-BFGS uses the second derivative (Hessian matrix) to account for the curvature of the cost function, allowing it to locate the

Table 5.8. Tested configurations specific to each of the solvers used in the development of the MPLC classifier.

Solver	**Adam**	**Epochs**	20, 30, 40, 50, 60, 80, 100
		Initial Learning Rate	0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2
		$Beta_1$	Default = 0.9
		$Beta_2$	Default = 0.99
		Epsilon	Default = 1^{-8}
	Sigmoid	**Epochs**	20, 30, 40, 50, 60, 80, 100
		Learning Rate Schedule	Constant, Invscaling, Adaptive
		Initial Learning Rate	0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2
		Momentum for Nesterov's gradient descent update	Default = 0.9
	LBFGS	**Maximum number of gradient steps**	200, 300, 400, 500

Table 5.9. The best model found during the tuning process for the MLP Classifier method with 3, 5, 7 and 10-seconds time windows.

Window Size	**Mean Fit Time**	**Hyperparameters**									**F1-Score**	
		Solver	**Hidden Layers Neurons**	**Batch Size**	**Epochs**	**Activation Function**	**Alpha**	**Warm Start**	**Shuffle Data**	**Initial Learning Rate**	**Train**	**Test**
3 s	11.3 s	Adam	(32, 32, 32)	256	100	Tanh	0	True	True	0.01	0.963	0.955
5 s	14.2 s		(128, 128)	64	60	ReLu	0	True	False	0.005	0.960	0.949
7 s	15.0 s		(128, 128)	512	100	Tanh	0	True	False	0.01	0.965	0.949
10 s	4.9 s		(128, 128, 128)	Auto	60	ReLu	0	True	False	0.005	0.967	0.949

minimum more easily [118]. A second analysis of the cost function convergency showed that was not the case. The optimization algorithm did not converge with any of the configurations used (even when the number of maximum steps was high).

Furthermore, our results show that, increasing the number of hidden layers and neurons per hidden layer required a larger computational effort that did not lead to significant changes in the test scores. Thus, models with lower training times were be preferred for the final method comparison.

Final Results

For final comparison purposes, our focus is on the 3-second window. Table 5.10 presents the training times and evaluation metrics of the model that had the best trade-off between the test F1-score and training time for the SVM, RF and MLP classifiers using a 3-second time window. More specifically, this table displays the F1-scores obtained for the three wireless link quality classes (C1 - normal link, C2 - short-term blocked link, C3 - long-term blocked link). Also, it shows the macro-averaged value of other secondary performance metrics, such as accuracy, precision and recall.

The analysis of this table shows that all three estimators were able to complete the classification task successfully, as the evaluation metrics were all high. Nonetheless, as one would expect, it is easier to distinguish situations where the link is either not blocked or fully blocked, than identifying an intermittently blocked link. The F1-scores observed in the normal or statically blocked examples are very close to 1 in both training and testing datasets, while in short-term blocked samples, the test F1-score decreases to 0.93%. This is because, as was found during the extensive data analysis presented in chapter 4, recognizing patterns in a link subject to abrupt signal quality variations (which do not always have the same impact on the network metrics) is a complex task. This is further exacerbated by the fact that the classifier is generalized to recognize patterns on different links, each with their own slightly different set of physical properties, and that, therefore, experience blockages at different intensities.

Still, the SVM and RF algorithms performed better than the neural network-based approach, especially in the minority class. They both reached a 93% F1-score in the testing dataset for this class, which is 3% higher than what was achieved with the neural network. Note that the hyperparameter tuning process is especially hard in neural networks, as more hyperparameters must be configured. For instance, to properly classify the link quality, one has to determine the optimal number of hidden layers, the number of neurons per layer, the batch size, and many other configurations, so that both performance and algorithmic convergence requirements are met. Besides, MLP are more likely to suffer from overfitting due to the fully-connected nature of layers, especially if a good L2 regularization coefficient is not being used. It is more than likely that the best settings found for the neural network are still not the optimal ones for the specific dataset, which is causing some level of overfitting to happen, as shown by the discrepancy between the training and testing curves of figure 5.8.

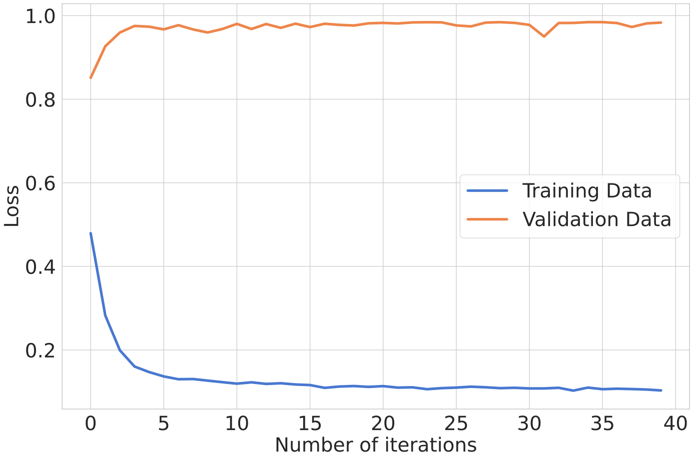


Figure 5.8. Multi-layer Perceptron (MLP) Classifier cross entropy vs number of epochs.

Therefore, less complex methods such as SVM should be preferred over MLPC neural networks for solving this particular classification task, since it was proven that it achieves the best F1-score, specifically in the minority class, in the shortest amount of time.

Table 5.10. Comparison of the results obtained for a 3 second time window with the best models found for SVM, random forest and MLPC.

		Fit Time	**F1-score**		**Accuracy**		**Precision**		**Recall**	
			Train	**Testing**	**Training**	**Testing**	**Training**	**Testing**	**Training**	**Testing**
Methods	**SVM**	3.3 s	0.997 - C1 0.971 - C2 0.999 - C3	0.993 - C1 0.932 - C2 0.995 - C3	0.997	0.991	0.9778	0.962	0.997	0.987
	Random Forest	14.6 s	0.999 - C1 0.997 - C2 0.999 - C3	0.995 - C1 0.937 - C2 0.997 - C3	0.999	0.993	0.999	0.967	0.999	0.9987
	MLPC	11.3 s	0.993 - C1 0.921 - C2 0.996 - C3	0.992 - C1 0.903 - C2 0.993 - C3	0.991	0.988	0.965	0.957	0.975	0.968

5.3.2 KPIs Forecasting

The forecasting work started with the development of an ARIMA model, to determine the suitability of the statistical-based models in making accurate multi-step ahead predictions of the most relevant KPI (SNR), in a blockage-prone mmWave link. For that purpose, the open-source python library *Sktime* library was used alongside the popular *statsmodels*. This *statsmodels* compatible library was developed as an individual initiative by Király et al. [119] to provide an unified, easy-to-use interface between specific ML models. In this work, the library was used to discover the optimal set of parameters of an *statsmodels's* ARIMA model, that could potentially complete the SNR forecast task.

The search for the optimal parameters can be described as follows. The method adopted starts by conducting a Distributed Coordination Function (DCF), to determine whether the time series is stationary or not. If the obtained test's p-value is superior to a defined threshold (0.05), the null hypothesis is rejected, and the series is considered non-stationary. In that case, the method will start testing different differencing orders d, until it finds the value that makes it stationary. When the optimal differencing order is found, the models seek the best set of p and q parameters, by searching through the ranges of orders that are defined in table 5.11. In the end, the Akaike Information Criterion (AIC) is used to select the best performing ARIMA model out of the tested combinations. The method uses L-BFGS optimizer to find the ARIMA model, which minimizes the AIC metric. However, due to data stationarity issues, the method did not find a suitable model that converged, leading to very high AICs as can be seen in table C.17 of the appendices. According to [120], this is due to ARIMA's inability to make the series stationary, even through high-order differentiation. Due to the SNR's large non-periodical variability, the method could not differentiate the series, regardless of the order. Figure 5.9 illustrates exactly that, as the forecasts made by the best performing ARIMA model are unable to capture the rapid changes caused by the link obstruction.

Table 5.11. Configurations details of ARIMA.

Configurations			**Metric to be minimized**
p	**d**	**q**	
[2, 10]	automatically selected using the ACF test	[2, 10]	AIC

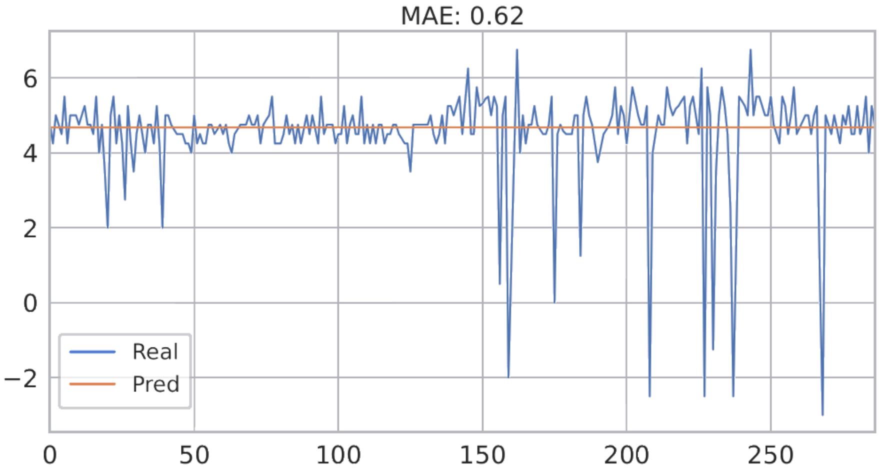


Figure 5.9. Forecasted data by the best performing ARIMA model for one of the mmWave paths.

Furthermore, employing ARIMA models that are univariate by nature to forecast each metric on a multivariate dataset (in this case, 17) is not a scalable option, especially in large backhaul networks. It would be necessary to fit an ARIMA model per metric and per path existent in the network, which would be impractical. Thus, the second part of the forecasting work focused on investigating scalable alternatives (LSTM-based RNNs), that are capable of detecting unexpected significant fluctuations on the relevant link's KPIs. These neural network-based approaches are more well-generalized alternatives, due to their ability to extract information from data collected across several links, some measured simultaneously, to perform more accurate forecasts of future KPIs. Moreover, DL-based models usually perform better than statistical-based ones in complex forecasting tasks, especially in non-linear and non-stationary datasets where no apparent trend and/or seasonality component can be identified. RNNs including LSTMs have a higher computational cost than statistical-based methods because they work on a sequential manner basis [54], and thus are not commonly used for predicting a single metric such what was done with ARIMA model. However, they are extremely useful in cases where a large number of time-series is to be predicted. Since the focus of this study is to determine whether an accurate prediction of the most relevant KPIs that characterize a link can be made in advance, a LSTM neural network approach was taken.

By providing all the relevant input metrics to the model, the LSTM neural network is able to recognize the interactions between features under different blockage conditions, to produce more accurate forecasts than those performed by statistical-based approaches. Rather than only using past records of a given metric to predict its future values (i.e., the univariate approach), several link KPIs are used by the network to estimate their future values. In theoretical terms, this means that the task of exploring the relationship between past link metrics to predict in advance metrics, such as SNR and Packet Error Rate, can be described as a multivariate and multi-step time series forecasting problem; and the task of predicting a metric for a given time window can be classified as a multiple-step problem.

However, predicting metrics one step ahead has no practical interest for the type of applications this is envisioned for. The reality is that it would be hard to perform any decision at the SDN controller (e.g., re-directing priority traffic) in such a short amount of time. Besides given that, the channel propagation conditions can change so rapidly, that it is more than likely that by the time the action would have been taken on the network, it would no longer be suitable. Therefore, a longer time window had to be selected as the model's input. In this case, each metric was predicted for the following 3-second time window.

In order to explore the ability of LSTM neural networks to model temporal dependencies, the 1-second dataset was framed as a supervised learning problem. That is, the original dataset was converted to a new format, where each new entry related to a given metric is composed by its 12-second lagged versions. This process enabled the neural network to use the patterns it found during training, in the other previous 12-second time windows, to predict the next 3-seconds. The converted dataset was then split into two parts. The first part was used during the model's training, to find the optimal set of weights that minimize the error, and the remaining data was saved for evaluating the model.

The LSTM-based RNN developed in this work uses the *Tensorflow Keras* Python framework [121] to create a model which is capable of predicting the 16 most relevant link metrics throughout a given time window. Recall that the final dataset collected is composed of 16 features (actually 17, but the path label that identifies each path is not being predicted) that represent a given window size. The forecasting network is composed of an input layer that accepts the input data; one or more adjacent hidden LSTM layers with several LSTM memory cells, that are responsible for performing the actual mapping between the input and outputs; followed by a standard feed-forward output layer that outputs the computed forecasts. In order to find a good-performing model, the optimal number of LSTM units, hidden LSTM layers, and many other parameters have to be optimized through a hyperparameter tuning process. For that purpose, *Talos* [122], a *TensorFlow*-compatible library, was used to automate the tuning and model evaluation processes. Among the alternatives studied, this library was found to be the simplest, yet most effective method for rapid hyperparameter optimization, while also providing full control of the entire process. The hyperparameters optimized and the ranges evaluated during the tuning process are depicted in table 5.12. Dropout regularization was applied during the best hyperparameters search at both input-to-hidden layers, hidden-to-hidden, and hidden-to-output connections, as shown in figure 5.10, to prevent overfitting. Dropout is a technique where neurons are temporarily dropped from the network in every training batch. In its simplest form, dropout is achieved by dropping each neuron with an independent probability p [123]. Note that, while there was also the possibility to drop recurrent connections in LSTM cells, that may cost the ability to properly model time dependencies, and thus this approach was avoided [124].

Table 5.12. Settings used during hyperparameter tuning of the LSTM-based model.

Model	**Epochs**	100
	Batch Size	2, 4, 8, 16, 64, 128, 256 and 512
	Number of Stacked LSTM Layers	1, 2, 3 and 4
	First LSTM layer's neurons	16, 32, 64, 100, 128, 200 and 256
	Other Hidden Layers' Neurons	8, 16, 32, 64, 100, 128, 200 and 256
	Dropout Rate	0.005, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 and 0.9
	Layer weight initializers	Uniform and normal distributions
	Learning Rate	Default, 0.0003, 0.0005, 0.003, 0.005, 0.01, 0.03, 0.05 and 0.1
Model Compile	**Optimizer**	Adam, RMSprop, Nadam and Sigmoid
	Losses	Mean Squared Error (MSE)

After an extensive tuning process, the model which obtained the best MSE and MAE error was selected to perform the final predictions. For exemplification purposes, table C.20 in the appendices presents results obtained with some of the hyperparameter combinations tested, including the one with the best performance. Table 5.13 summarizes the best hyperparameter settings and values of other test-related parameters used to validate the model. Figure 5.10 illustrates how these hyperparameters are used to produce a LSTM architecture capable of forecasting future 3-second time windows. This model is designed to accept multivariate inputs (the previous 12-second time window for each of the metrics), and path ID that helps the algorithm to differentiate between paths. After training, the neural network can use the discovered mapping function to convert the test data into a new output sequence, that is composed of the forecasted values of these 16 metrics for the next 3 seconds.

The losses of the best performing model are represented for each training and validation epoch in figure 5.11a. Since the model converged at the 60 epochs mark, one may choose to only train the model until this point in future developments. Given that the MSE train and testing curves are almost superimposed, there is no evidence of overfitting. In addition, it can be seen in figure 5.11b that the MAE also achieved convergency.

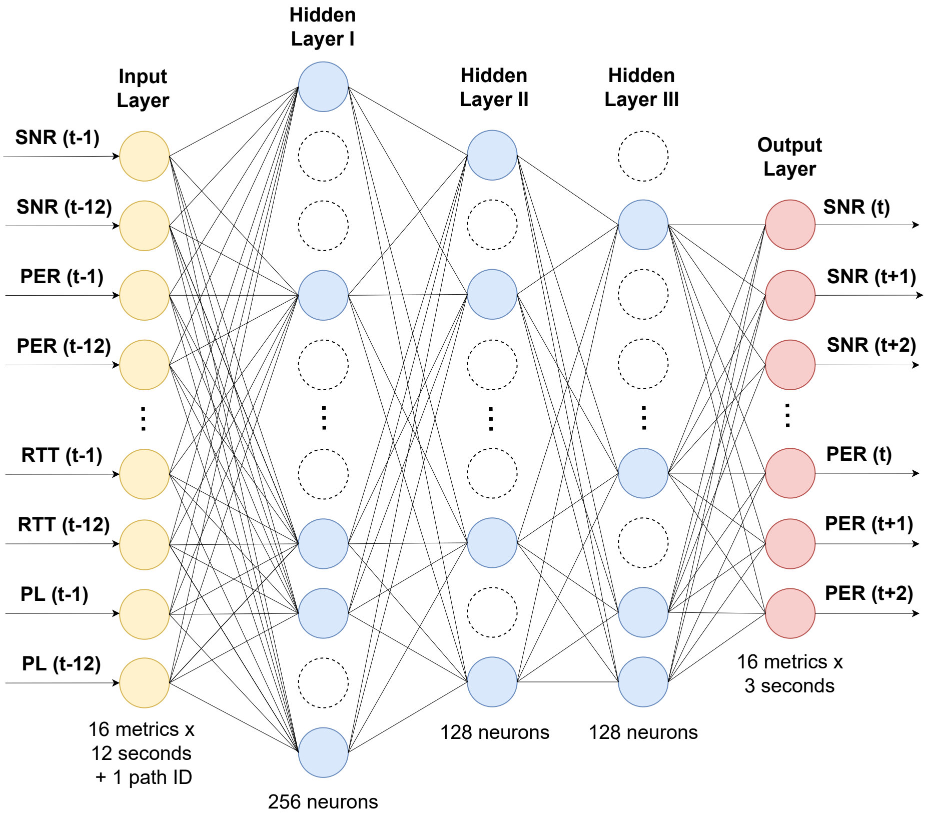
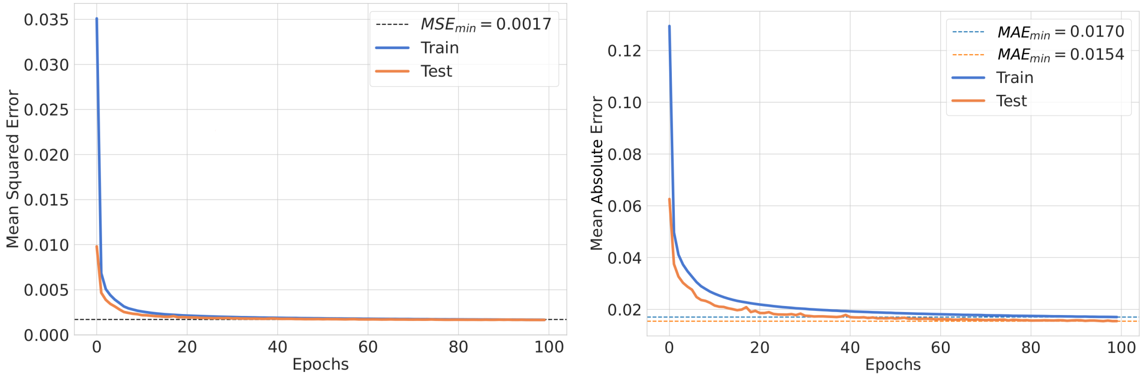


Figure 5.10. LSTM-based RNN neural network architecture used for multi-step metrics forecasting.

Table 5.13. Settings, training and testing time of the best performing LSTM model.

Data	**Training Dataset Size**	79 555
	Testing Dataset Size	24 862
Model	**Epochs**	100
	Batch Size	128
	Number of Stacked LSTM Layers	3
	First LSTM Layer's Neurons	256
	Other hidden Layers' Neurons	128
	Dropout Rate	0.2
	Layer weight initializers	Uniform distribution
Model Compile	**Optimizer**	Nadam
	Loss Function	MSE
	Learning Rate	0.001 (default)
Performance	**Training time**	5 min and 21 s
	Testing time	2.15 s



(a) Mean Squared Error

(b) Mean Absolute Error

Figure 5.11. Loss function (MSE) and MAE in training and testing of the best performing model.

The Mean Absolute Error obtained with the best performing LSTM model, is shown in table 5.14 for the 1, 2 and 3 second ahead metrics predictions. These results show that the forecasting ANN was able to predict with average errors of less than two units (except the average and maximum data rate), most of the physical metrics, for the totality of the 20 paths present in the dataset. Still, there are some small errors that occur for all metrics, that associated with noise the channel introduces even under normal conditions. Recall, for example, that even on a non-blocked link, small variations in signal quality and beam variations (within the previous beam's neighborhood) are observed. Therefore, to analyze the quality of the prediction, one must have to account for normal propagation channel variability. For instance:

- An average prediction error of 1.3 units for the RX beam index in the 3-seconds ahead predictions indicates that the algorithm can accurately predict abrupt transitions of the receive beam when an obstruction occurs. If that were not the case, errors that may have exceeded 50 units would have been observed;
- An 0.63 average prediction error for the SNR at $t + 2$ means that, on average, for the totality of the 20 links, our predictions do not incur a more extreme change than, for example, a drop from 20 to 19.37 dB;
- The MAE of 32 ms at $t + 2$ for the maximum RTT is within the 36 ms variation range observed for the links in normal scenario (see table A.5 in the appendices);
- An average error of 12.45 Mbps at the 3-second ahead received rate prediction is also again contained within the normal scenario variability of about 50 Mbps (see table A.11 in the appendices);
- The average error of 1.13% at $t + 2$ for the packet loss is again included in the normal scenario variability of maximum of 11.5% (see table A.8 in the appendices).

Moreover, as expected, forecasts further ahead in time have larger associated errors, although they are pretty similar.

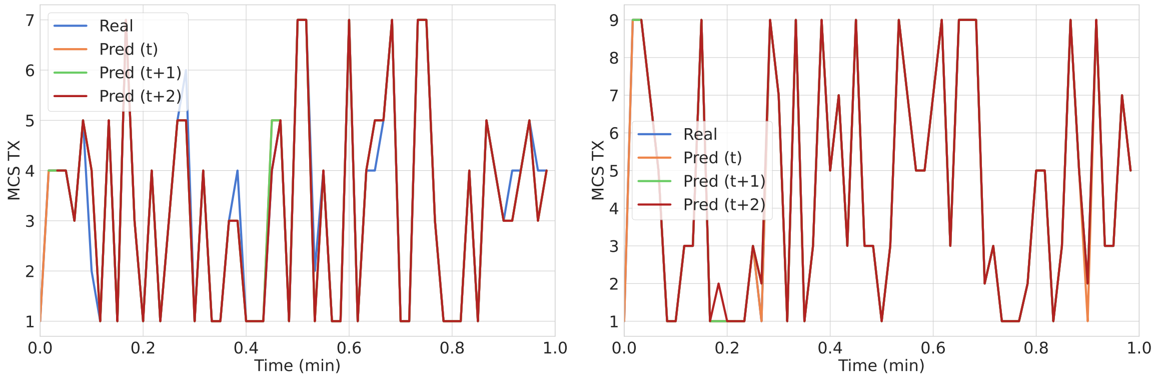
Table 5.14. Mean Absolute Error obtained with the best performing LSTM model.

		MAE (t)	**MAE (t+1)**	**MAE (t+2)**
Physical metrics	Beam Index RX	1.30 ± 3.80	1.37 ± 3.97	1.34 ± 3.96
	Beam Index TX	0.55 ± 0.27	0.61 ± 0.33	0.59 ± 0.30
	MCS RX	0.17 ± 0.18	0.17 ± 0.18	0.18 ± 0.19
	MCS TX	0.15 ± 0.16	0.17 ± 0.18	0.18 ± 0.19
	Mbps TX	32.41 ± 30.56	34.76 ± 33.16	37.48 ± 35.89
	Mbps TX (Max)	34.35 ± 32.34	36.63 ± 35.02	39.00 ± 37.59
	Packet Error (%)	1.22 ± 1.05	1.32 ± 1.10	1.39 ± 1.21
	rcpi (dBm)	0.55 ± 0.17	0.56 ± 0.19	0.58 ± 0.23
	SNR (dB)	0.60 ± 0.13	0.61 ± 0.14	0.63 ± 0.19
Network metrics	Sent Mbps	8.86 ± 3.18	10.49 ± 4.04	12.21 ± 5.25
	Received Mbps	9.03 ± 3.09	10.67 ± 4.00	12.45 ± 5.28
	TCP retransmissions	54.73 ± 22.30	72.46 ± 31.43	91.78 ± 41.42
	RTT (Mean)	10.31 ± 5.79	11.15 ± 7.87	12.46 ± 11.34
	RTT (Max)	26.80 ± 16.71	29.24 ± 20.28	32.33 ± 26.17
	RTT (Min)	1.24 ± 0.83	1.55 ± 1.65	1.88 ± 2.65
	Packet Loss (%)	0.75 ± 0.48	0.95 ± 0.63	1.13 ± 0.78

Tables C.18 and C.19 of the appendices also present the MSE and RMSE, respectively, of the predictions. However, the interpretation of these results is not as direct as the MAE, since they do not represent the actual forecast error (the difference between the real and predicted values), but rather the average of the squared errors and squared root of the average squared errors, respectively. MSE greatly penalizes errors, no matter how small they are, and regardless of whether they are acceptable in the specific context or not.

Our observation is further reinforced by the graphical comparative analysis carried out. Figure 5.12a and 5.12b illustrate forecasts made, for two different paths. These graphs show that it is possible to accurately detect sudden transmit MCS changes one, two and three seconds in advance. Similarly, figures 5.13 and 5.17 show that the developed model is able to accurately estimate SNR and received data rate, respectively, for two uplink paths suffering from different levels of obstruction. This is also true for the Packet Loss Rate, Average RTT and Number of TCP Retransmissions as shown by the small prediction errors of figures 5.15, 5.16 and 5.18, respectively. Nonetheless, the PER of figure 5.14 shows that, even if the average error remained below two units for most links' samples, there are (at least in this specific case) samples in which the peak value prediction is not accurate. For more graphical examples of the quality of the predictions made by the algorithm, please refer to section C of the appendices.

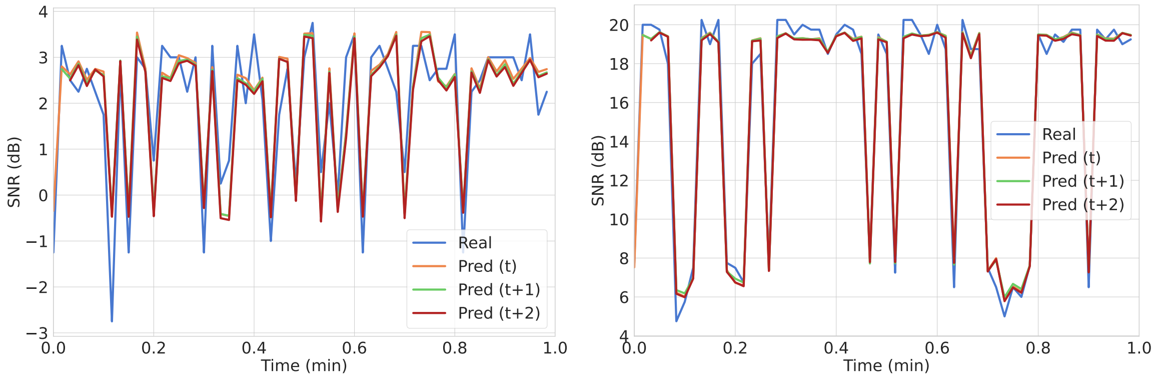
Recall that, as with most time-series forecasting methods, predicting the exact value of peaks is usually not possible for this case due to the channel noise. However, this is not a limiting factor. Under the specific context of this work, it is more critical to identify when significant changes in channel conditions will occur than their exact impact on each network metric.



(a) Path 1.

(b) Path 4.

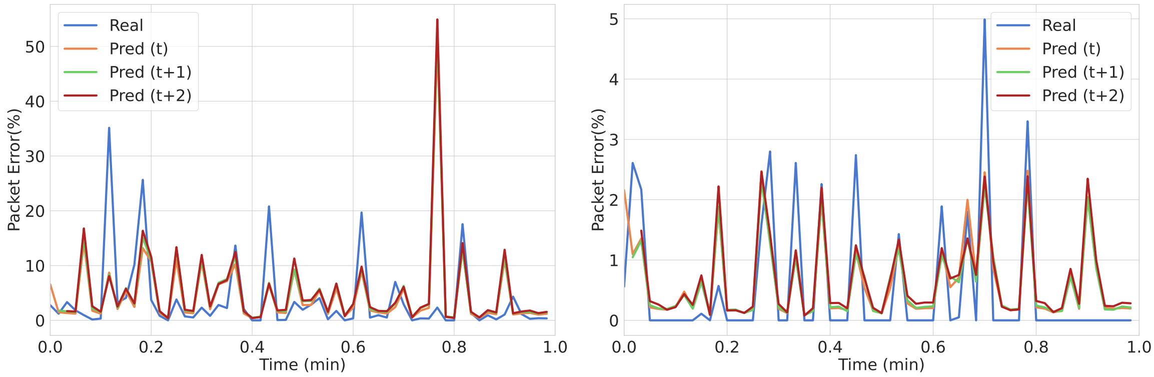
Figure 5.12. MCS predicted by the forecasting algorithm for the uplink of two mmWave paths.



(a) Path 1.

(b) Path 4.

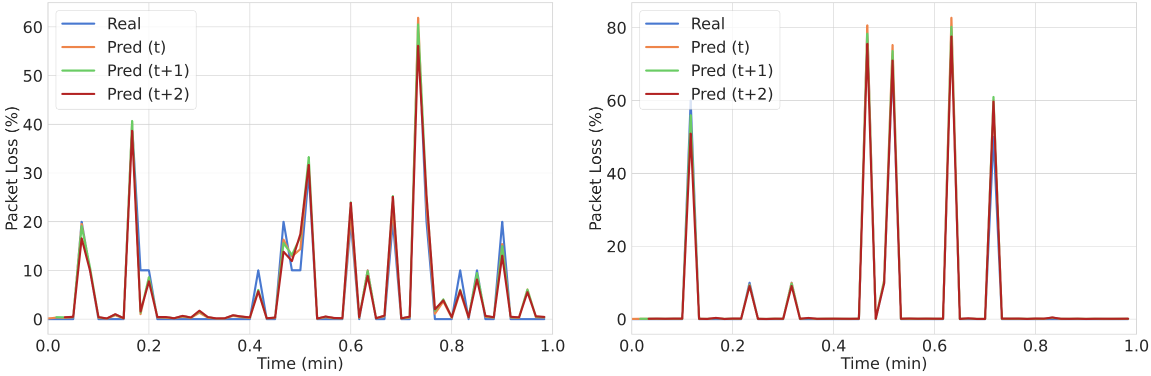
Figure 5.13. SNR predicted by the forecasting algorithm for two mmWave paths.



(a) Path 1.

(b) Path 4.

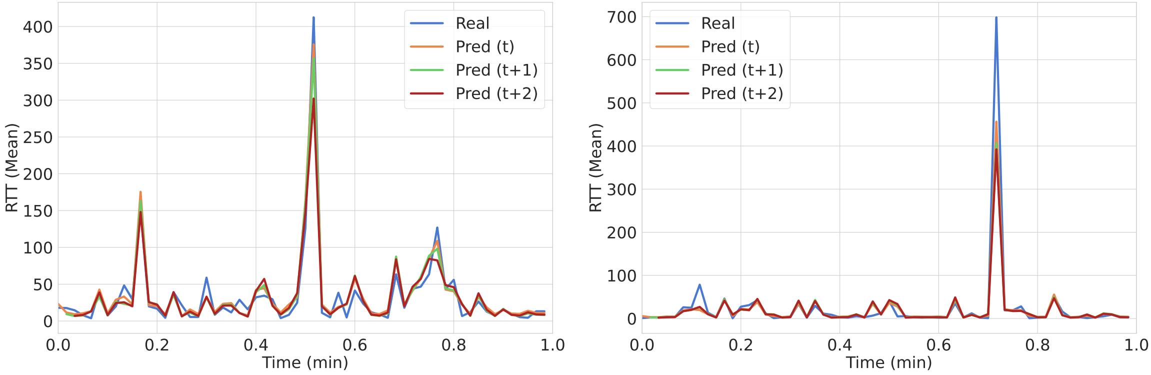
Figure 5.14. PER predicted by the forecasting algorithm for two mmWave paths.



(a) Path 1.

(b) Path 4.

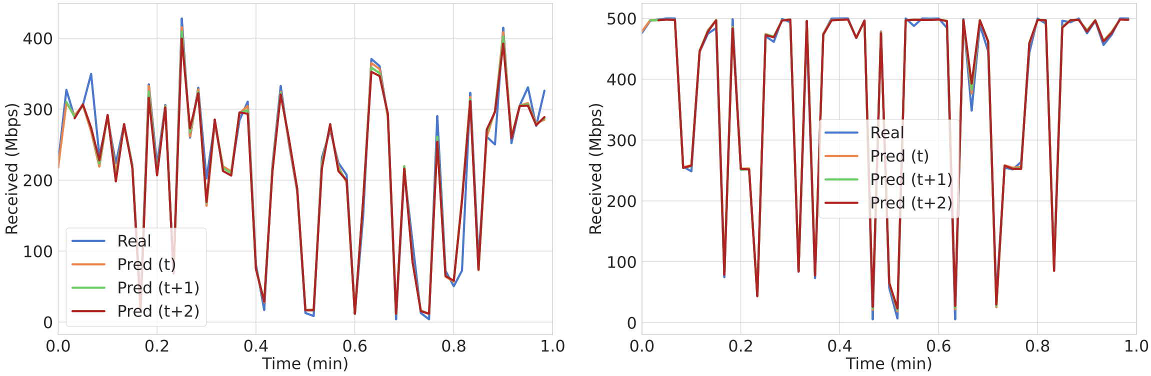
Figure 5.15. Packet loss rate predicted by the forecasting algorithm for two mmWave paths.



(a) Path 1.

(b) Path 4.

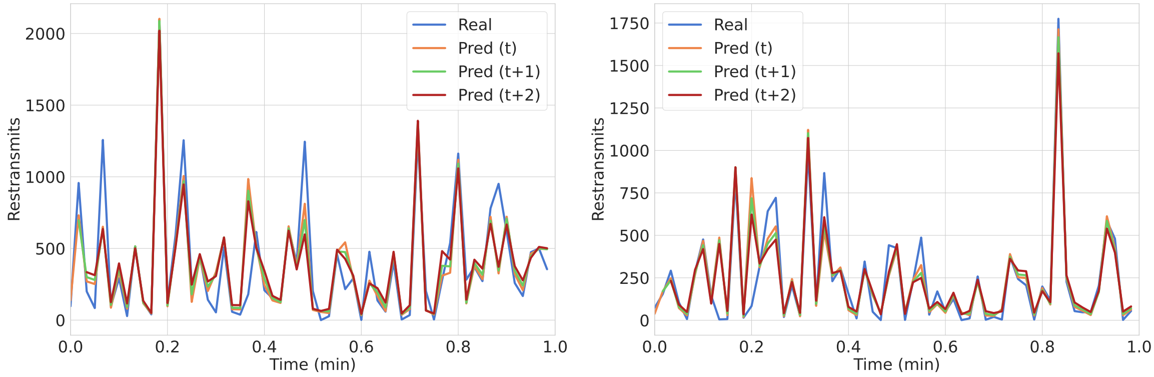
Figure 5.16. Average RTT predicted by the forecasting algorithm for two mmWave paths.



(a) Path 1.

(b) Path 4.

Figure 5.17. TCP throughput predicted by the forecasting algorithm for two mmWave paths.



(a) Path 1.

(b) Path 4.

Figure 5.18. TCP retransmissions predicted by the forecasting algorithm for two mmWave paths.

Thus, we can conclude that the LSTM-based RNN developed is able to capture the behavior of the link's KPIs suffering from long-term and short-term blockage to perform accurate 3-steps ahead predictions. This includes metrics such as the number of TCP retransmissions and PER (for most cases), which did not show very well-defined patterns in automatic mode during our first analysis.

5.4 Summary

To ensure reliable network performance, an ML-based approach can be taken, which exploits the data collected in links, under different scenarios, to predict when significant link quality decreases are to be expected.

In this chapter, solutions were proposed to solve two specific problems associated with blockage-prone networks: 1) how to classify wireless link quality according to the available metrics; and 2) how to use past records to predict future link KPIs. A common ML pipeline was proposed to solve both of these problems, which consists of a preprocessing module and a problem-specific ML module.

In order to perform wireless link quality classification, three different approaches were proposed, which include SVM, RF ensemble and MLP Classifier. The performance obtained with the best settings found during hyperparameter tuning, for the three classifiers, was compared for datasets consisting of time windows of 3, 5, 7, and 10 seconds. The test scores achieved with these classifiers were similar, regardless of the window size, differing no more than 2% in the F1-score of the minority class. Still, SVM was shown to reach superior performance at a lower computational cost, surpassing even a Multi-layer Perceptron neural network performance-wise. SVM achieved a macro-averaged F1-score of 0.989 and 0.973 for the training and testing datasets, respectively, with a 3.3 second mean fitting time.

In the second part of the chapter, the quality of forecasts of a traditional statistical-based model such as ARIMA was compared, to those made by a LSTM-based RNN. It was found that, while ARIMA models provide a lightweight approach to time series forecasting, their inability to extract patterns from an extreme non-linear and non-stationary dataset results in a poor-performing model. On the other hand, the LSTM-based forecasting method proposed was shown to capture the interactions between past feature observations under different environments to produce accurate forecasts, for 16 KPIs in the next 3-seconds time widow. Still, some errors exist, due to the propagation channel's variability, even in normal conditions.

The output of the proposed ML-based approaches brings many possibilities to improve network reliability, in different field domains, improving the performance of network coding techniques or SDN controllers. These two topics are explored in the following chapter.

Chapter 6

mmWave Modelling and Network Coding

Being able to test in real-life mmWave networks and their links enables the possibility to use the data in supporting the research community. From one side, it is possible to test and validate resilience mechanisms that address the challenges of mmWave-based backhauls, by providing the possibility to develop a mmWave model to a simulator, such as ns-3. From the other side, it can also be used to assess the potential of novel techniques, such as Network Coding (NC), to improve the performance of these networks.

The structure of this chapter is as follows. The first section presents a brief description of the extension of an existent IEEE 802.11ad/ay ns-3 module to simulate our real-life testbed. In the second and last section, the potential of NC techniques, such as Random Linear Network Coding (RLNC) are evaluated in the physical infrastructure.

6.1 Simulating a real WigGig-based backhaul

Commercial solutions, such as the CSS Metnet one, make the task of testing and validating the network mechanisms proposed in the literature, a time-consuming, if not impossible, process. Using proprietary software and firmware in each device limits the number of configurations that can be made at both a physical and network level. Also, there is the issue that this equipment is relatively recent and still has features under development, such as its Self-organising Network (SON) algorithm. For instance, currently, changing the initial network topology is only possible in the case of a link outage through the beamforming training process triggered at the MAC layer. Furthermore, this process was observed to operate in a conservative manner. That is, it was not possible to trigger a new beamforming training process any time the link quality decreases to a value lower than a defined threshold as that option is left to the manufacturer.

The firmware version 5.8.0 released on the first of June of 2021 is the right step in that direction. It introduced the ability to periodically scan and measure radio links, which is expected to be used in future releases to dynamically adjust the network topology from a network manager perspective. However, at the time this dissertation was developed, that feature was not implemented yet. Thus, the over-complicated testing process alongside the inability to test large-scale networks led to the search for an accurate IEEE 802.11ad simulator that could be used to quickly validate protocols, coding techniques, resilience mechanisms, among many other possibilities. This kind of approach will allow the validation of the current and future mechanisms that are being developed using IEEE 802.11ad devices in the next generation WiGig networks (802.11ay).

Among the several network simulators that were studied, only ns-3 had a module that was equipped to simulate WiGig WLANs. NS-3 is a free-to-use discrete-event network simulator [125] [126], widely used for research and education purposes. It is composed of core components and modules which are written in C++ language and that use Python for binding purposes. However, this simulator lacks a graphical user interface, since it is not a high-level modeling language. Instead, each simulation event is a C++ function call organized by a scheduler.

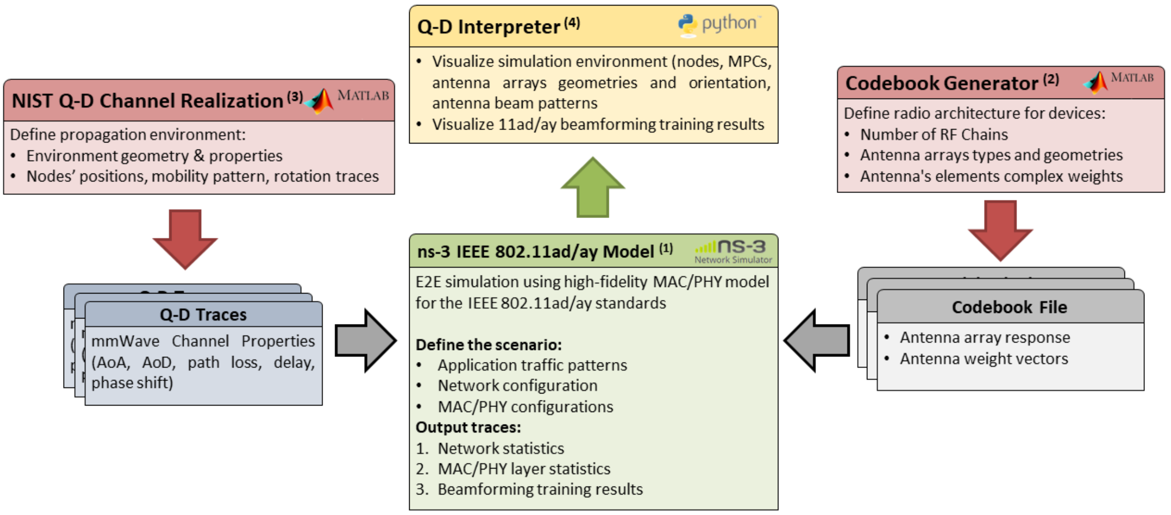
In this work, the ns-3 IEEE 802.11ad module implemented by Assasa et al. [127] is used. This open-source simulation framework provides a high-fidelity tool for building and testing dense wireless networks based on the IEEE 802.11ad/ay standards. As can be seen in figure 6.1, the framework consists of four main components that are [127]:

- **ns-3 IEEE 802.11ad/ay model:** the actual component responsible for accurately simulating the standards' MAC and PHY layers. It also includes the implementation of a quasi-deterministic channel that, based on the profiles generated by 60 GHz ray-tracing software, enables modeling real mmWave propagation environments [128];

- **Codebook Generator:** a MATLAB-based application that generates a file, called beam codebook, that contains the information related with the radios. It is in this file that the number of antenna arrays per RF chain, the number of elements per antenna array and the radiation patterns of the phased arrays are specified. This app is not yet available to the general public. However, it was kindly made available to us for research purposes by Dr. Hany. This module was adapted to simulate our specific context;

- **NIST Q-D Channel Realization:** a MATLAB three-dimensional ray-tracing application for accurate point-to-point indoor and outdoor mmWave channel modeling [129]. Through this approach, complex environments with multipath propagation can be simulated. A channel Q-D trace is generated per pair of communicating stations and specifies the following parameters: number of multipath components, the path loss of the channel in dB, the phase shift due to propagation, reflection and Doppler effect (in case of mobility), the Angle of Arrival (AoA) and Angle of Departure (AoD) for each ray [128]. This file alongside the codebook is used by the ns-3 simulator to compute the estimated received power;

- **Q-D interpreter:** python application for visualizing the simulated scenario (nodes position and rotation, as well as the antenna array geometry and antenna beam patterns) and the results of beamforming training [130].


Q-D Interpreter (4)
python
Visualize simulation environment (nodes, MPCs, antenna arrays geometries and orientation, antenna beam patterns
Visualize 11ad/ay beamforming training results
NIST Q-D Channel Realization (3)
MATLAB
Define propagation environment:
Environment geometry & properties
Nodes' positions, mobility pattern, rotation traces
Codebook Generator (2)
MATLAB
Define radio architecture for devices:
Number of RF Chains
Antenna arrays types and geometries
Antenna's elements complex weights
Q-D Traces
mmWave Channel Properties (AoA, AoD, path loss, delay, phase shift)
ns-3 IEEE 802.11ad/ay Model (1)
ns-3 Network Simulator
E2E simulation using high-fidelity MAC/PHY model for the IEEE 802.11ad/ay standards
Define the scenario:
Application traffic patterns
Network configuration
MAC/PHY configurations
Output traces:
1. Network statistics
2. MAC/PHY layer statistics
3. Beamforming training results
Codebook File
Antenna array response
Antenna weight vectors

(1) Developed by Hany Assasa and Nina Grosheva at IMDEA Networks Institute WNG Group
(2) Developed by Hany Assasa
(3) Developed by NIST CTL Group in collaboration with University of Padova SIGNET Group
(4) Developed by NIST CTL Group

Figure 6.1. The WiGig framework used [127].

As it was shown in the measurement campaign of [131], the Q-D methodology can be used to model the CIR of a mmWave channel. In this approach, the signal power is considered to be primarily concentrated in a set of strong direct rays and some relatively lower rays (NLOS components) generated by reflections with a predefined distribution. Hence, D-rays are obtained using ray tracing techniques and the R-rays by using a statistical distribution [128].

The WiGig module [127] features every functional aspect described earlier for the IEEE 802.11ad standard. This includes the support for receiving and transmitting beamforming training in both the BHI and DTI periods, support for beam refinement operations, support for different MCS at the PHY layer, quasi-deterministic channel modeling, dynamic and static service period allocation, beamforming link maintenance, spatial sharing, and interference assessment.

However, some limitations were observed during the process of simulating our testbed. Currently, the simulator lacks support for WiGig devices with multiple antenna arrays when using the parametric codebook class. Our tests showed that the receive beamforming training process was only being correctly performed with one antenna phased-array. Furthermore, the mechanism that had been previously implemented to introduce blockage worked based on equally decreasing the signal quality for all sectors of a particular blocked communication path. Intuitively, it is easy to understand that, depending on the blocker dimensions, blockage of one beam may also affect the performance of others, especially beams in close vicinity. Thus, a more accurate obstacle module was developed for adding interference to a mmWave link in ns-3. The pipeline used to simulate the mmWave backhaul can be summarized as follows:

1. **Generation of a parametric codebook:** the "IEEE 802.11ad Codebook Generator app" was used to create a beam codebook that describes the radiation properties of the four radios that compose the CCS nodes. To calculate the directivity of each phased-array, a circular microstrip patch at 60GHz was considered, as it was the element used in the real device (see figure 6.2). The generated file specifies the following parameters:
    - Number of RF chains: 1;
    - Number of antenna arrays per RF chain: 4;
    - Number of elements per phased-array: 16x2;
    - Azimuth and Elevation angles covered by each element: [-180; 180] degrees and [-90; 90], respectively;
    - The directivity of each element (for each azimuth and elevation combination);
    - The steering vectors associated with each element (for each azimuth and elevation combination): the steering matrix is composed by (361 x (2 x 181)) x (16x2) elements;
    - The quasi-omnidirectional AWVs for each element;
    - The directional AWVs used in each element to generated each of the 64 sectors.
2. **Generation of the Q-D files:** the MATLAB 3D ray-tracing application was used to generate a Q-D file for each pair of stations. Each Q-D file models a quasi-deterministic mmWave channel with a given propagation loss (calculated as the free space loss) and delay;
3. **Development of a blockage module**: this module is responsible for introducing obstruction in a link formed between a pair of stations. It allows configuring the blockage position, introduced loss according to the blocker's material, the blocker's width and thickness, as well as specifying which beams are affected;
4. **Simulation using the ns-3:** This step entails many operations. From a WiGig perspective, it includes configuring the number, type and position of nodes; the operation frequency (60.48 GHz); the channel bandwidth (2.16 GHz); the transmit power (19 dBm) and the MCS. Also, it includes the configuration of the appropriate modules to simulate the PHY, MAC, and network layers. Besides the typical object instances that belong to the WiGig module which allows us to simulate the PHY and MAC layers [127] (such as *MacLow*, *UpperMac*, *WifiMac*, *DmgStaWifiMac*, *DmgApWifiMac*, *ParametricCodebook*, *SpectrumDmgWifiPhy, QdPropagationLoss*), other typical ns-3 classes had to be used (*NodeContainer*, *NetDeviceContainer*, *BridgeNetworkDevice*, *OnOffApplication*) to simulate the network. In specific, the *BridgeNetworkDevice* was used to connect the wired (SDN switches) and wireless part of the network (mmWave nodes). In addition, the *OnOffApplication* was used to inject packets in the network.

Table 6.1. CCS node radio properties that were used to generate the parametric codebook

Simulated properties of CCS nodes	
Number of RF chains per node	1
Number of phased-arrays/ radios per node	4
Node's field of view (with total of 4 radios)	Horizontal 300º electronic steerable
Gain of each phased-array	19 dBi
Gain of each element	9 dBi
Number of elements per antenna array	16 x 2
Radio Azimuth field of view	90º
Radio Elevation field of view	180º
Number of sector per radio	64
Azimuth 3dB beamwidth	5º
Elevation 3dB beamwidth	~20° (+/-10°)

Moreover, it is in the ns-3 *ParametricCodebook* class that the parsing of Q-D file occurs for each link as well as the computation of the directivity of each of the 64 sectors according to the AWVs of the codebook. Additionally, changes (mainly in the *DmgWifiMac*) were made to correct the lack of support for receiving beamforming training with multiple antennas when using the Parametric codebook approach.

The radiation pattern obtained with the generated codebook and the values provided by the manufacture for 5 sequential beams are shown in figure 6.3. These results indicate that the simulated directional beams have similar behavior to the data obtained in the real-life measurements. In the quasi-omnidirectional case, the simulated and real patterns vary slightly due to the lack of knowledge of which elements were active and inactive in the manufacture's quasi-omnidirectional pattern. Nonetheless, the overall shape is similar. This small difference is expected to be corrected in a near future.

A more detailed description of the modules and the process required to simulate our physical mmWave backhaul is presented in a paper being developed alongside other SNOB-5G project researchers. This author's contribution on this front mainly focused on developing a codebook that could accurately simulate the physical properties of radios, as explained previously in this section.

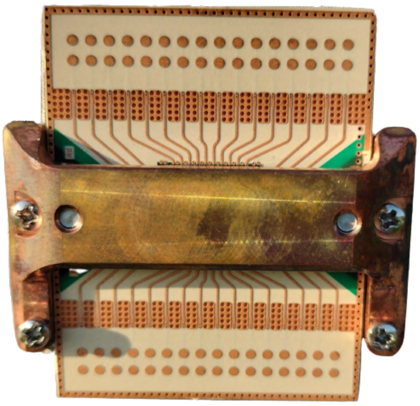
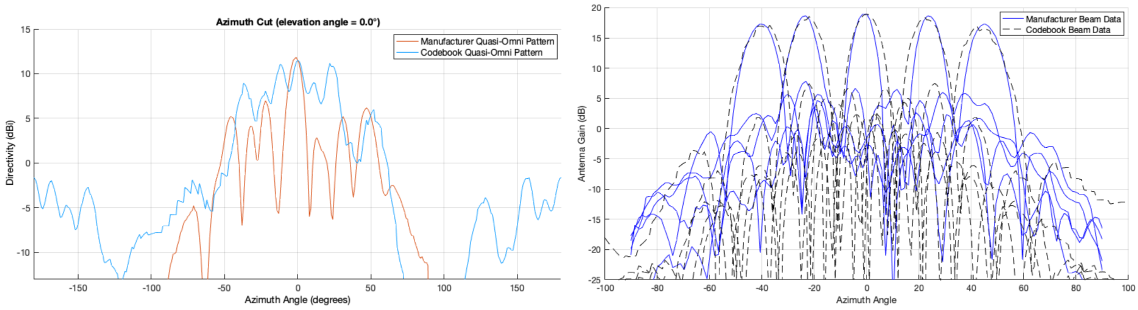


Figure 6.2. Phased-array antenna employed at each node's radio.



(a) Quasi-omnidirectional patterns.

(b) Directional patterns.

Figure 6.3. Radiation patterns measured by the manufacturer for the quasi-omni mode (left) and for 5 directional beams (right) vs the patterns of the simulated codebook

6.2 Improving network coding performance

The application of network coding concepts, in particular RLNC, has already been shown through simulations to improve both reliability and efficiency of mmWave systems in [102] and [103]. Still, so far, the study has mostly focused on cellular networks evaluated through a simulator like ns-3. In this work, the rateless approach of RLNC was implemented using the Kodo library [104] to evaluate the actual gains that NC techniques can have on a real WiGig backhaul network.

Kodo [104] is an open-source C++ library aimed to be used in practical research studies of network coding algorithms. To access the basic functionalities of Kodo RLNC, the high-level API of the open-source Kodo-RLNC-C library [132] was used. Two separate modules (an encoder and decoder) were developed with this library in the context of the SNOB-5G project, one for each node. Additionally, a non-coded version was also deployed to compare the performance of traditional forwarding approaches and network coding-based ones in a link suffering from signal quality degradation due to long-term blockage.

Both the coded and uncoded approaches require an UDP session to be established between the sender and the receiver. This UDP session is first used to inform the receiver of the test-related settings and later to send the coded and uncoded packets or symbols. Overall, the proposed network coding architecture can be described as follows:

1. Send the test-related parameters that the receiver needs to decode, the received packets, as well as to detect when the stopping criterion is met. In the coded approach, this includes reporting the symbol size and the number of symbols that is going to be used in the encoding process;
2. Generate random information symbols, encode those information symbols using RLNC and transmit them on top of UDP;
3. Transmit the encoded symbols over the mmWave link tested and decode the received symbols to retrieve the original information;
4. On the receiver side, increase the overall packet count whenever enough coded packets have been received to successfully decode one packet;
5. End the experiment in the case the total number of symbols (of ten 1MB downloads) have been successfully received or if a socket timeout is detected;
6. Compute the success rate, packet loss and the cumulative download time for each experience as follows:

$$\text{Success Rate} = \frac{\text{Completed downloads}}{\text{Total downloads}}$$

$$\text{Packet Loss} = \frac{\sum_{n=1}^{n=10} \text{Received symbols/packets}}{\text{Total symbols/packets}}$$

7. Repeat steps 2 through 6 until the total test duration for all MCS has passed.

On the other hand, in the non-coded approach, randomly generated packets are forwarded on top of UDP over the same network link without any additional processing. In that case, an experiment (10x1MB downloads) may end due to receiving the total number of pre-established packets or, similarly to the Kodo-based approach, due to socket timeout if data is not being received. Again any time an experience ends, a new one begins, and that process is repeated until all MCSs have been tested for a 30-minute long window. The general architecture used to test the Kodo-based and uncoded approaches is presented in figure 6.4.

Note that the concept of symbols is only valid under a coding perspective to refer to a smaller portion of the information source. In contrast, in a traditional network, there are only packets. Hence, to ensure a fair comparison between the two implementations, the number and size of packets sent in the non-coded approach had to be matched to the number and size of symbols transmitted using RLNC. These parameters are depicted for both approaches in table 6.2. Note also that in UDP it is not possible to mitigate the losses associated with an

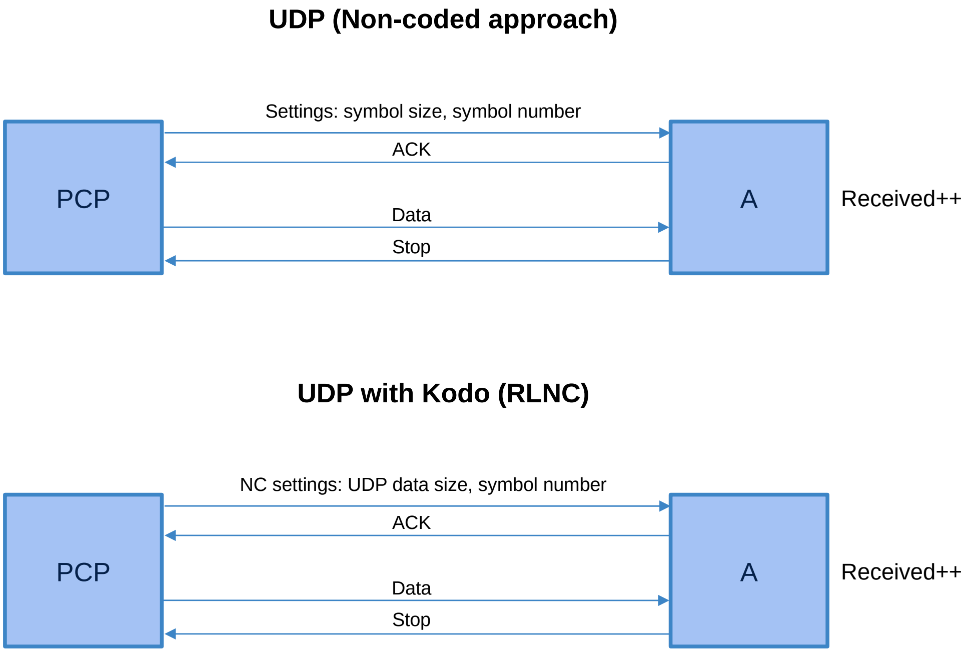


Figure 6.4. Architecture used for testing the performance of a mmWave link using a simply forwarding scheme (top) and a fixed-blocked RLNC technique (bottom).

mmWave channel since it does not employ retransmission mechanisms such as in TCP. As can be seen in table 6.2, the RLNC approach employed uses a fixed blocked length (i.e., a fixed code rate) throughout the entire test duration.

Table 6.2. Test-related parameters configured for the coded and uncoded approaches.

Network Test Scenarios **MCS = 1, 3, 5, 7, 9 and Auto during a 30-minute time-window** **in a statically blocked mmWave link**	
UDP with Kodo (RLNC)	**UDP**
Symbol Size: 8192 btyes	Packet Size: 8192
Symbol Number: 122	Packet Number: 122
Download Size ∼1 MB	Download Size ∼1 MB

The relevant PHY layer (RCPI, SNR, and PER) and transport layer metrics (packet loss) measured throughout the testing process for different MCS modes (fixed and automatic) are presented in Figure 6.5. In addition, this figure shows the minimum required levels of SNR and RCPI for maintaining each tested MCSs. These thresholds were reported by the node's manufacturer in its documentation (for more details regarding the MCSs and associated thresholds, see table 4.1). To simplify the analysis, the metallic obstacle presented earlier was maintained at the same position (blocking the Line-of-sight signal) for the entire test duration of all MCS modes.

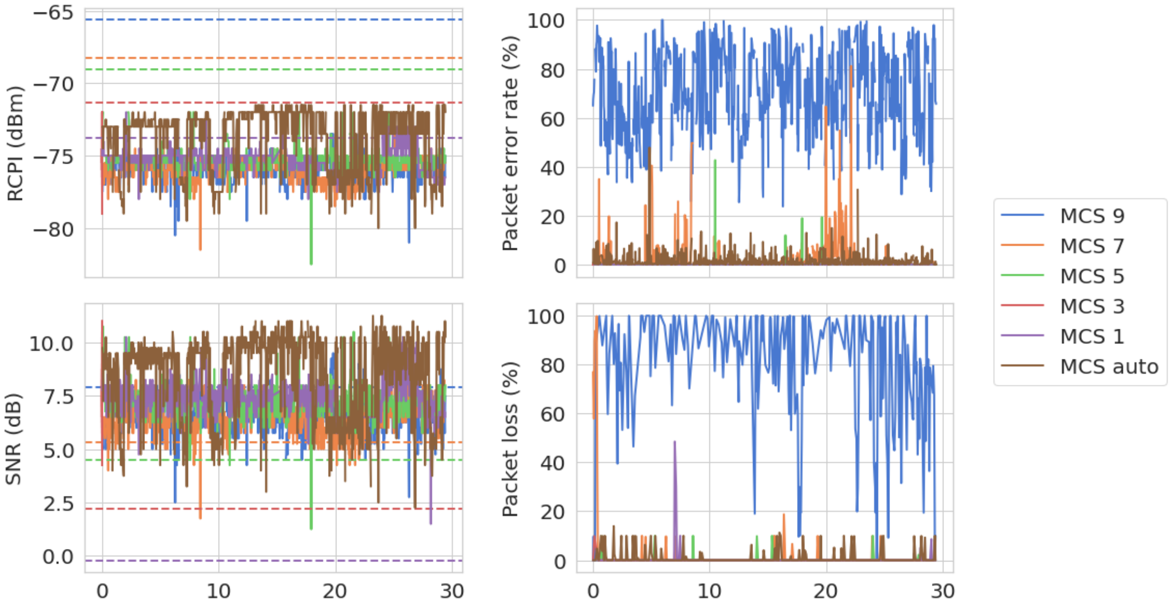


Figure 6.5. PHY and transport layer metrics measured for MCS 1, 3, 5, 7, 9 and Auto in mmWave link suffering from long-term obstruction.

A quick analysis of figure 6.5 shows that the RCPI signal level is lower than the defined threshold for almost all MCSs (except for a MCS fixed of 1 and the automatic MCS). However, the SNR requirement is still being fulfilled most of the time. Note that sudden decreases of the RCPI and, consequently, SNR still occurred due to slight obstacle movements caused by strong wind. Note that these large fluctuations are way higher than those observed in 4 for the long-term blocked, to the extent that it resembles the behavior observed for short-term blocked links.

The results obtained in Figure 6.6 show that using RLNC increases the likelihood of achieving a success rate closer to 100%. The improvements introduced in terms of delivery reliability are clear for the MCS 7, where it was observed that linearly combining different packets through RLNC allows overcoming the high packet losses, which ultimately contributes to reducing the amount of packet errors introduced by the channel even under extreme conditions. Moreover, network coding techniques have improved the success rate even in automatic mode, where it is possible to adjust the MCS to reflect the propagation channel variations. However, compensating the erasure in mmWave channels with Network Coding techniques is only possible at the expense of a higher number of coded packet transmissions. For instance, it can be seen in Figure 6.8 that a larger number of transmissions is needed to consistently achieve a success rate closer or equal to 100% with network coding, which ultimately increases the cumulative download time, as shown in figure 6.9. Note that the higher point density observed in figure 6.9 for the coded approach comes from the fact that coding is more likely to achieve 100% success rates. Nonetheless, the increase in the download time was shown not to be significant, as it was only necessary to introduce a maximum of 20 additional packets to overcome high packet losses in this scenario. In fact, the additional overhead introduced by the encoding operation was demonstrated to be no more than 2% in the automatic mode (see figure 6.7).

Overall, it is shown that a simple RLNC technique, which uses an apriori fixed block size, is already able to improve the performance of real mmWave links suffering from high packet errors rates caused by obstruction.

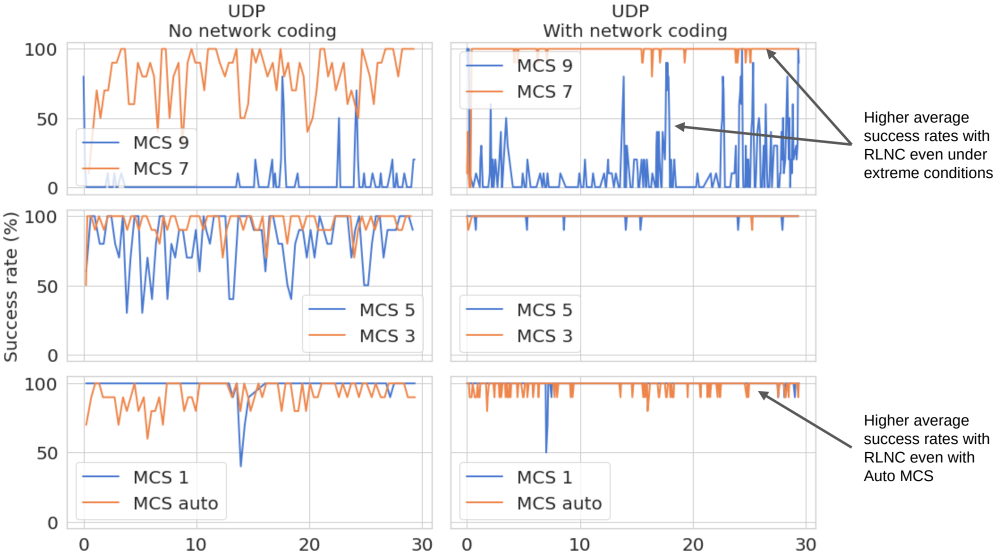


Figure 6.6. Success rate obtained with 1 experience (ten 1MB downloads) with the uncoded (right) and coded (left) approaches for all MCS tested.

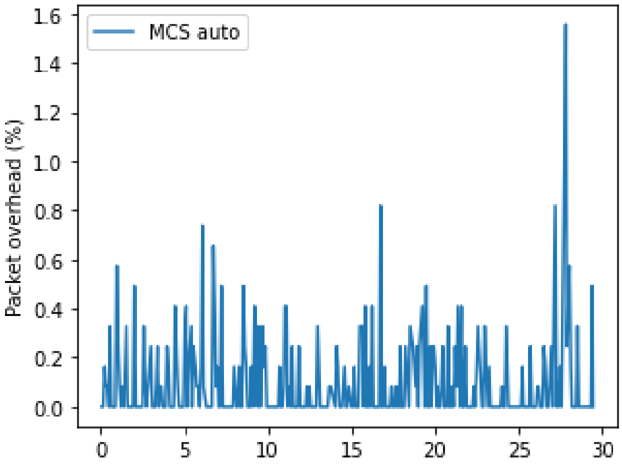
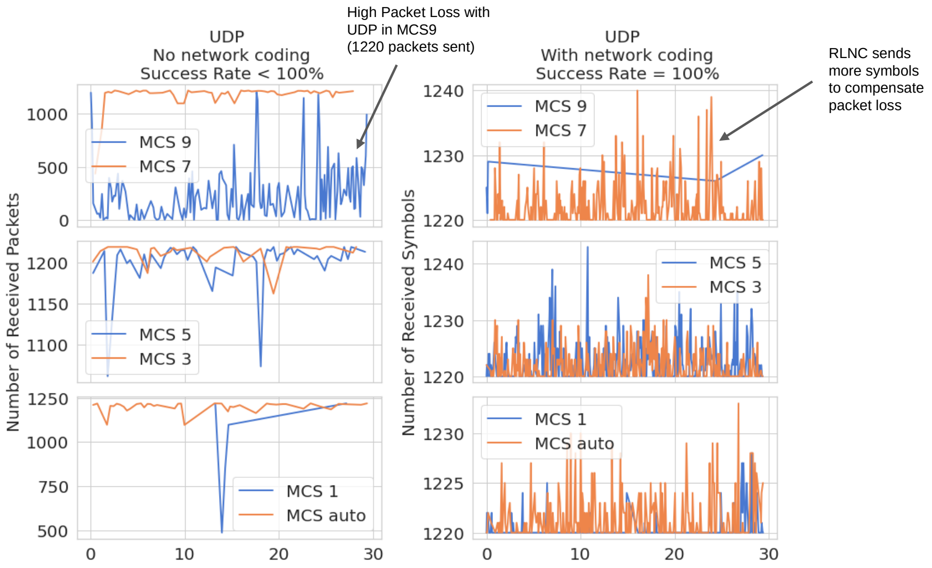


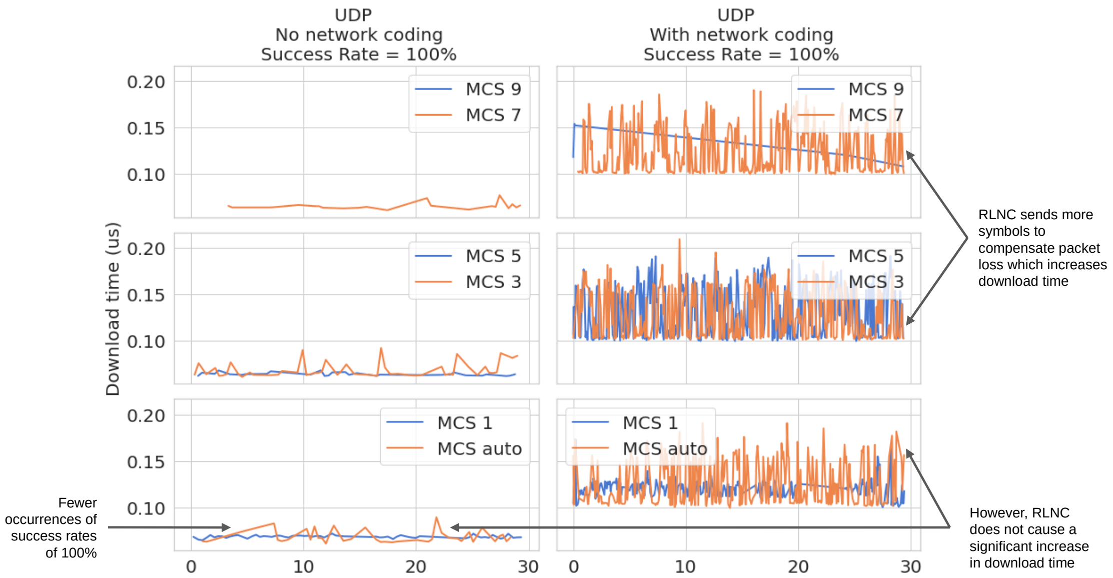
Figure 6.7. Packet overhead inserted by the encoding process in the automatic MCS for experiences with 100% success rate.



High Packet Loss with UDP in MCS9 (1220 packets sent)

RLNC sends more symbols to compensate packet loss

Figure 6.8. Number of total packets transmitted with 1 experience (ten 1MB downloads) with the uncoded (right) and number of transmitted symbols with the coded approach (left) for all MCS tested.



RLNC sends more symbols to compensate packet loss which increases download time

Fewer occurrences of success rates of 100%

However, RLNC does not cause a significant increase in download time

Figure 6.9. Cumulative download time measured with 1 experience (ten 1MB downloads) with the uncoded (right) and coded (left) approaches for all MCS tested.

However, this approach still has its limitations, since it is not able to adapt to RTT fluctuations and loss rate. In order to support real-time transmissions in highly variable propagation channels, the Adaptive Causal Network Coding with Feedback technique proposed by MIT is currently being deployed in our testbed. By relying on packet-level FEC and on the feedback it receives from the receiver, the sender can track the channel erasure pattern to dynamically adjust the code rate, thus providing a good trade-off between throughput and in-order delay. However, adjusting the code rate requires waiting for a full RTT for the receiver's feedback. To reduce the associated overhead, a novel approach to AC-RLNC is being proposed where the code rate is adjusted solely based on forecasted metrics by the LSTM-based RNN. The computation of the expected average RTT and Packet Loss three seconds ahead of time further improves the performance of AC-RLNC by controlling the amount of inserted apriori and posteri redundancy before the impact of sudden link quality drops is felt.

The novel approach described will generate a scientific paper in collaboration with MIT and other IT researchers, to explore the potential of using metric forecasts (based on past observations) to improve the performance of real-life mmWave backhauls employing AC-RLNC.

6.3 Summary

This chapter presented the extension of an existent ns-3 WiGig module to simulate our backhaul network, and the utilization of RLNC and AC-RLNC techniques to further improve network resilience.

This chapter began with the description of the process used to replicate the behavior of our testbed using the WiGig ns-3 module proposed by Assasa et al. [127]. For that purpose, a codebook was generated to define the specific radiation properties of our nodes. Overall, the simulated directional and quasi-omnidirectional radiation patterns were shown to be in agreement with the patterns disclosed by the radio manufacture. Also, an obstruction module was developed from scratch to introduce blockage with different properties into a mmWave link.

The second part of this chapter shows the potential of a fixed-blocked RLNC approach in improving the reliability of a physical mmWave backhaul suffering from high packet loss due to obstruction. Specifically, the test results show that linearly combining different packets through RLNC allows overcoming the high packet losses in highly blocked channels and increases the likelihood of achieving a success rate closer to 100%, even in the cases where MCS can be adjusted to reflect the propagation channel variations. Moreover, it has been shown that the overhead associated with the coding operation is no more than 2% for a network operating in automatic MCS mode, and does not significantly increase the download time compared with the non-coded approach.

At last, the shortcomings of a fixed-blocked RLNC were presented. We found that, although a fixed-blocked RLNC approach can improve the success rate under extreme conditions, it can only improve it to some extent. Because the proposed approach could not adapt the code rate to fit the varying channel conditions, reliable delivery of all transmitted symbols is not guaranteed. To address this, an enhanced version of AC-RLNC is being discussed where metrics forecasted by the LSTM-based neural network, such as packet error, packet loss, and average RTT, are used to adapt the code rate before a significant drop in signal quality occurs.

Chapter 7

Conclusions and future work

5G mobile networks are currently being deployed to meet challenging multi-gigabit data rates and low latency of current demands. With the introduction of the millimeter-wave spectrum in ultra-dense scenarios, the wireless backhaul has become a key enabler for 5G technology, by presenting a cost-effective and scalable alternative to the typical fiber backhaul. By leveraging the frequently unused millimeter-wave spectrum, data rates comparable to those offered by fiber connections can be obtained, but at the expense of higher susceptibility to blockage.

In this work, an SDN-controlled backhaul using IEEE 802.11ad was successfully deployed in a controlled outdoor scenario, which allowed collecting physical, network, and transport layer metrics of millimeter-wave links suffering from long-term and short-term blockage. In this work, the most extensive multi-layer measurement campaign performed to this date (to the best of our knowledge), in WiGig-based networks was performed, which includes an extensive cross-layer analysis of the behavior of long-term and short-term blocked 60GHz links. The analysis conducted provided several insights into the inner operation of WiGig networks. It was found that maintaining fixed high MCSs in long-term blocked channels may induce packet errors as high as 100%, an RTT that can be in the order of a few seconds, and packet losses as high 90% due to the errors. Even if dynamically adjusting the MCS is possible, instantaneous PERs can reach the 40%, maximum delays can reach the one-second mark, and data rates can vary by a hundred Mbps. This effect was exacerbated in short-term links, as they suffered from more extreme MCS changes upon sudden obstructions. Temporary LOS obstruction was shown to cause maximum delays of half a second, and PER maximum usually around 20%, and in more extreme cases, it even led to temporary link failure.

The higher susceptibility to blockage makes it challenging to meet critical applications with constant data rate and maximum delay requirements, in a backhaul scenario. Thus, the work developed in the second part of this thesis focused on researching mechanisms at multiple layers that could improve network reliability and resilience of these blockage-prone mmWave backhauls. These include a wireless link quality classifier, a deep-learning KPI forecaster, and some insights of how lower layer can relax, when using random linear network coding techniques.

A general pipeline composed of a preprocessing and an ML-specific module was proposed to address the wireless link quality prediction and the forecasting of future KPIs. This pipeline accepts the multi-layer metrics collected at different sampling rates under short-term and long-term obstruction and outputs a problem-specific variable. In the case of the classification, its output is a label that describes whether a mmWave link is under normal operation or suffering from long-term or short-term blockage. In the case of the forecasting task, it outputs multiple continuous values that represent the estimations for the link's KPIs for the next 3-seconds. In the preprocessing module, the raw data passed a series of preprocessing steps involving data cleaning, aggregation, scaling, and feature selection. The preprocessed was then split using a 70:30 ratio for performing training and the final evaluation of the models. During the training stage, hyperparameter tuning was used alongside cross-validation to find the model-specific hyperparameter settings that maximize performance. For the wireless link quality classification, the performance of three popular ML and DL-based classifiers (SVM, RF and MLP) was compared for datasets composed of 3, 5, 7 and 10-second aggregation windows. The test scores achieved with these classifiers were similar, regardless of the window size, differing no more than 2% in the F1-score of the minority class. SVM still was able to achieve the best performance in the shortest amount of time. The results obtained found that it is possible to use a SVM to classify with a 98% F1-score the quality of a millimeter-wave link (in a 3-second time window), according to the type of obstruction it faces (non-blocked, short-term blocked, and long-term blocked links). For the metrics forecasting task, the estimations performed by a statistical-based model like ARIMA were compared with those performed by a LSTM-based neural network. It was found that ARIMA is unable to extract complex time dependencies, due to the non-stationary nature of the SNR time series. Moreover, employing ARIMA models that are univariate by nature implies fitting a model to forecast each KPI, thus having scalability issues. Besides, it cannot leverage the correlation between metrics to enhance its prediction capabilities. On the other hand, an LSTM-based neural network was shown to accurately detect temporal patterns for multiple link KPIs under normal, long-term blocked and short-term blocked scenarios, but at the cost of increased complexity. The average prediction error of the developed forecaster was contained within the maximum absolute deviation, presented by links operating in a non-obstructed environment.

The ability to test in a real-life mmWave network enabled exploring, for the first time, the performance of NC techniques in a experimental mmWave channel suffering from erasures. In this work, the rateless approach of RLNC was deployed in the nodes of a mmWave link using the Kodo library, and was compared with the performance obtained with traditional forwarding approaches using UDP. Our preliminary results show that the coding approach allows overcoming the high packet losses in highly blocked links, and increases the likelihood of achieving a success rate closer to 100%, even in the cases where the MCS is adjusted accordingly to the channel fluctuations.

Furthermore, the data obtained in a real-life mmWave network was leveraged to produce an accurate model of our outdoor testbed that employs IEEE 802.11ad, using the ns-3 simulator. This provides a way to quickly test and validate protocols, coding techniques, resilience mechanisms, and many other possibilities, without the constraints of proprietary-based solutions. Overall, the simulated directional and quasi-omni patterns are aligned with the radiation patterns disclosed by the manufacturer.

The work developed throughout this thesis allows to draw valuable conclusions about the behavior of mmWave networks. It was demonstrated the potential of network coding techniques to increase network robustness. It was also shown that it is possible to detect patterns in multi-layer metrics to detect short-term and long-term blockage, with the help of ML techniques, even in situations indistinguishable to the human eye. The possibility of detecting short-term obstruction, even in temporal windows that do not present significant variations of SNR, leads us to think that the negative impact of obstruction can be felt sometime after it happens. Furthermore, this "system memory" combined with the short-term obstruction history results in the possibility of detecting future unexpected LOS obstructions and their impact on multi-layer KPIs quite precisely.

In the future, the integration of the SVM classifier and the LSTM forecasting methods will enable producing a framework capable of forecasting the future wireless link quality. This will allow SDN switches to redirect the traffic generated by critical applications to radio links, that are predicted to be under normal conditions, and capable to meet its requirements. Beyond that, the ability to perform ahead-of-time forecasts of values such as RTT, SNR, and PER brings many other possibilities to improve network reliability. For instance, the predictions made for the average channel RTT may be used to further increase the performance of AC-RLNC coding techniques when employed in a real millimeter-wave backhaul. Furthermore, the proposed resilience mechanisms can now be easily tested on a network simulator without the physical hardware limitations. In summary, possible directions for future research are:

- Development of a ML-based framework that uses the wireless link quality and KPIs forecaster pipeline to forecast future link states;

- Leveraging the estimates of future links qualities to trigger traffic rerouting actions at the network layer by the SDN controller. This will allow high priority traffic to be rerouted through more reliable links, which can still fulfill the throughput and maximum delay requirements;

- Leveraging the forecasted KPIs to further increase the robustness of MIT's AC-RLNC on mmWave links experiencing obstruction. This approach will be used to adjust the coding rate before the blockage is experienced, no longer requiring waiting a full round-trip time for the receiver's feedback;

- Qualifying in the ns-3 testbed the impact caused by an obstruction in a millimeter-wave backhaul, using the SDN's dynamic rerouting traffic actions;

- Collecting data in a larger deployment installed in an urban setting;
- Extend the proposed mechanisms to the next generation WiGig networks (IEEE 802.11ay).

In the near future, maintaining a reliable connection in a WiGig-based network may require deploying a set of hybrid multi-layer techniques to achieve redundancy. This can be accomplished in multiple ways, for instance, by employing network coding techniques at the transport layer or by using an SDN controller that leverages the expected link quality to dynamically adapt the routes at layer 3. In a more advanced version, physical links with other neighbors may also be established to achieve this purpose. These mechanisms do not replace the beamforming training operations defined in IEEE 802.11ad and, more recently, in IEEE 802.11ay, but rather complement them. Re-training at the MAC layer is the last resort and is expected to be triggered only when all other resilience techniques fail. The new generation of WiGig networks defines new PHY and MAC layers that support technologies to decrease the training overhead, such as group training. Nonetheless, this remains a reactive mechanism, and thus, adjustments are only performed when signal quality is effectively degraded.

Since there are no COTS devices available in the market that support IEEE 802.11ay, it is not possible to evaluate the overhead of the new types of beamforming training. Nevertheless, only sub-optimal adjustments can be performed due to the lack of knowledge of the entire network topology in each node. As such, the resilience mechanisms proposed in this master's thesis should be evaluated in future 60 GHz networks.

References

[1] L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020. DOI: 10.1109/JIOT.2019.2948888.

[2] C.-X. Wang, F. Haider, X. Gao, *et al.*, "Cellular architecture and key technologies for 5g wireless communication networks," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 122–130, 2014. DOI: 10.1109/MCOM.2014.6736752.

[3] J. Andrews, S. Buzzi, W. Choi, *et al.*, "What will 5g be?" English, *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014, ISSN: 0733-8716. DOI: 10.1109/JSAC.2014.2328098.

[4] *Snob-5g - scalable network backhauling for 5g*, Jan. 2021. [Online]. Available: https://snob-5g.com/.

[5] *Ubiwhere leads international consortium with mit, it and uc in new 5g project.* [Online]. Available: https://www.ubiwhere.com/en/news/ubiwhere-leads-project-snob-5g.

[6] A. N. Uwaechia and N. M. Mahyuddin, "A comprehensive survey on millimeter wave communications for fifth-generation wireless networks: Feasibility and challenges," *IEEE Access*, vol. 8, pp. 62 367–62 414, 2020. DOI: 10.1109/ACCESS.2020.2984204.

[7] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. W. Heath, "Spatially sparse precoding in millimeter wave mimo systems," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1499–1513, 2014. DOI: 10.1109/TWC.2014.011714.130846.

[8] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A survey of millimeter wave communications (mmwave) for 5g: Opportunities and challenges," *Wireless networks*, vol. 21, no. 8, pp. 2657–2676, 2015.

[9] *Report itu-r m.2376-0: Technical feasibility of imt in bands above 6 ghz*, Last accessed 2020-11-16. [Online]. Available: https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2376-2015-PDF-E.pdf.

[10] D. Nandi and A. Maitra, "The effects of rain on millimeter wave communication for tropical region," in *2019 URSI Asia-Pacific Radio Science Conference (AP-RASC)*, IEEE, 2019, pp. 1–3.

[11] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Communications Magazine*, vol. 49, no. 6, pp. 101–107, 2011. DOI: 10.1109/MCOM.2011.5783993.

[12] I. Uchendu and J. Kelly, "Survey of beam steering techniques available for millimeter wave applications," *Progress in Electromagnetics Research B*, vol. 68, pp. 35–54, 2016.

[13] S. Singh, F. Ziliotto, U. Madhow, E. Belding, and M. Rodwell, "Blockage and directivity in 60 ghz wireless personal area networks: From cross-layer model to multihop mac design," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, pp. 1400–1413, 2009. DOI: 10.1109/JSAC.2009.091010.

[14] A. Maltsev, R. Maslennikov, A. Sevastyanov, A. Khoryaev, and A. Lomayev, "Experimental investigations of 60 ghz wlan systems in office environment," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, pp. 1488–1499, 2009. DOI: 10.1109/JSAC.2009.091018.

[15] T. Manabe, Y. Miura, and T. Ihara, "Effects of antenna directivity and polarization on indoor multipath propagation characteristics at 60 ghz," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 3, pp. 441–448, 1996. DOI: 10.1109/49.490229.

[16] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra, "Beamspy: Enabling robust 60 ghz links under blockage," USENIX Association, 2016, ISBN: 9781931971294.

[17] *High rate 60 ghz phy, mac and hdmi pal.* ECMA standard 387, ECMC TC48, 2008. [Online]. Available: https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-387.pdf.

[18] *Ieee 802.15 wpan millimeter wave alternative phy task group 3c (tg3c)*, 2005. [Online]. Available: https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-387.pdf.

[19] "Ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 3: Enhancements for very high throughput in the 60 ghz band," *IEEE Std 802.11ad-2012 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012 and IEEE Std 802.11aa-2012)*, pp. 1–628, 2012. DOI: 10.1109/IEEESTD.2012.6392842.

[20] *Wirelesshd: Wirelesshd specification overview*, 2009. [Online]. Available: https://www.ecma-international.org/publicationsS/files/ECMA-ST/ECMA-387.pdf.

[21] *Wireless gigabit alliance.* [Online]. Available: http://wirelessgigabitalliance.org/.

[22] X. Wang, L. Kong, F. Kong, *et al.*, "Millimeter wave communication: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 1616–1653, 2018. DOI: 10.1109/COMST.2018.2844322.

[23] *Wirelesshd, wigig to compete in the future*, Apr. 2010. [Online]. Available: `https://archive.eetindia.co.in/www.eetindia.co.in/ART_8800603045_1800005_NT_d2147c75.HTM`.

[24] T. Baykas, C. Sum, Z. Lan, *et al.*, "Ieee 802.15.3c: The first ieee wireless standard for data rates over 1 gb/s," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 114–121, 2011. DOI: `10.1109/MCOM.2011.5936164`.

[25] Y. Ghasempour, C. Silva, C. Cordeiro, and E. Knightly, "Ieee 802.11ay: Next-generation 60 ghz communication for 100 gb/s wi-fi," *IEEE Communications Magazine*, vol. PP, pp. 1–7, Oct. 2017. DOI: `10.1109/MCOM.2017.1700393`.

[26] "Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 2: Enhanced throughput for operation in license-exempt bands above 45 ghz," *IEEE Std 802.11ay-2021 (Amendment to IEEE Std 802.11-2020 as amendment by IEEE Std 802.11ax-2021)*, pp. 1–768, 2021. DOI: `10.1109/IEEESTD.2021.9502046`.

[27] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, and J. Zhang, "Overview of millimeter wave communications for fifth-generation (5g) wireless networks-with a focus on propagation models," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, pp. 6213–6230, 2017. DOI: `10.1109/TAP.2017.2734243`.

[28] H. Assasa and J. Widmer, "Extending the ieee 802.11 ad model: Scheduled access, spatial reuse, clustering, and relaying," in *Proceedings of the Workshop on Ns-3*, 2017, pp. 39–46.

[29] I. Paudel, "Qos provisioning in future wireless local area networks," Ph.D. dissertation, Institut National des Télécommunications, Paris, 2015. [Online]. Available: `https://tel.archives-ouvertes.fr/tel-01217195/document`.

[30] C. Emna, L. Fourati, and L. Kamoun, "Phy/mac enhancements and qos mechanisms for very high throughput wlans: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 15, pp. 1714–1735, Feb. 2013. DOI: `10.1109/SURV.2013.013013.00084`.

[31] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "Ieee 802.11 ad: Directional 60 ghz communication for multi-gigabit-per-second wi-fi," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 132–141, 2014.

[32] B. Schultz, "802.11 ad - wlan at 60 ghz - a technology introduction," *Rohde & Schwarz*, 2013.

[33] C. Cordeiro, D. Akhmetov, and M. Park, "Ieee 802.11 ad: Introduction and performance evaluation of the first multi-gbps wifi technology," in *Proceedings of the 2010 ACM international workshop on mmWave communications: from circuits to networks*, 2010, pp. 3–8.

[34] H. Assasa and J. Widmer, "Implementation and evaluation of a wlan ieee 802.11ad model in ns-3," in *Proceedings of the Workshop on Ns-3*, 2016, pp. 57–64.

[35] S. Choi, J. Del Prado, S. Mangold, *et al.*, "Ieee 802.11 e contention-based channel access (edcf) performance evaluation," in *IEEE International Conference on Communications, 2003. ICC'03.*, IEEE, vol. 2, 2003, pp. 1151–1156.

[36] G. H. Sim, T. Nitsche, and J. C. Widmer, "Addressing mac layer inefficiency and deafness of ieee802. 11ad millimeter wave networks using a multi-band approach," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, 2016, pp. 1–6.

[37] C. Pielli, T. Ropitault, N. Golmie, and M. Zorzi, "An analytical model for cbap allocations in ieee 802.11 ad," *IEEE Transactions on Communications*, 2020.

[38] P. Zhou, K. Cheng, X. Han, *et al.*, "Ieee 802.11ay-based mmwave wlans: Design challenges and solutions," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 1654–1681, 2018. DOI: 10.1109/COMST.2018.2816920.

[39] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with eyes closed: Mm-wave beam steering without in-band measurement," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2416–2424. DOI: 10.1109/INFOCOM.2015.7218630.

[40] C. R. C. M. Da Silva, J. Kosloff, C. Chen, A. Lomayev, and C. Cordeiro, "Beamforming training for ieee 802.11 ay millimeter wave systems," in *2018 Information Theory and Applications Workshop (ITA)*, 2018, pp. 1–9. DOI: 10.1109/ITA.2018.8503112.

[41] W. Feng, Y. Li, D. Jin, L. Su, and S. Chen, "Millimetre-wave backhaul for 5g networks: Challenges and solutions," *Sensors*, vol. 16, no. 6, 2016, ISSN: 1424-8220. DOI: 10.3390/s16060892. [Online]. Available: https://www.mdpi.com/1424-8220/16/6/892.

[42] K. Sakaguchi, E. Mohamed, H. Kusano, *et al.*, "Millimeter-wave wireless lan and its extension toward 5g heterogeneous networks," *IEICE Transactions on Communications*, vol. E98.B, Jul. 2015. DOI: 10.1587/transcom.E98.B.1932.

[43] R. Chen, *An introduction to the 5g small cell*, Aug. 2021. [Online]. Available: https://www.litepoint.com/blog/an-introduction-to-the-5g-small-cell/.

[44] Z. Pi, J. Choi, and R. Heath, "Millimeter-wave gigabit broadband evolution toward 5g: Fixed access and backhaul," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 138–144, 2016. DOI: 10.1109/MCOM.2016.7452278.

[45] P. Legg and R. McConnell, "Meshed backhauling of small cells using ieee802.11ad at 60ghz," in *2018 European Conference on Networks and Communications (EuCNC)*, 2018, pp. 393–397. DOI: 10.1109/EuCNC.2018.8442433.

[46] Alain, *Edgehaul™ millimeter wave small cell backhaul system*, Jan. 2016. [Online]. Available: https://www.interdigital.com/presentations/edgehaul-millimeter-wave-small-cell-backhaul-system.

[47] *Metnet 60g unlicensed mmwave mesh datasheet - ccsl.com*, Feb. 2020. [Online]. Available: https://www.ccsl.com/v1/uploads/files/Metnet-60G-Mesh-datasheet.pdf.

[48] *Forecasting: Principles and practice (2nd ed).* [Online]. Available: https://otexts.com/fpp2/stationarity.html.

[49] Y.-S. Park and S. Lek, "Artificial neural networks: Multilayer perceptron for ecological modeling," in *Developments in environmental modelling*, vol. 28, Elsevier, 2016, pp. 123–140.

[50] T. M. Leite, *Neural networks, multilayer perceptron and the backpropagation algorithm*, Oct. 2018. [Online]. Available: https://medium.com/@tiago.tmleite/neural-networks-multilayer-perceptron-and-the-backpropagation-algorithm-a5cd5b904fde.

[51] S. Siami Namini, N. Tavakoli, and A. Siami Namin, "A comparison of arima and lstm in forecasting time series," Dec. 2018, pp. 1394–1401. DOI: 10.1109/ICMLA.2018.00227.

[52] S. Siami-Namini and A. S. Namin, *Forecasting economics and financial time series: Arima vs. lstm*, 2018. arXiv: 1803.06386 [cs.LG].

[53] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, pp. 107–116, Apr. 1998. DOI: 10.1142/S0218488598000094.

[54] *Understanding lstm networks*, Aug. 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[55] B. Rocca, *Handling imbalanced datasets in machine learning*, Feb. 2021. [Online]. Available: https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28.

[56] A. Chugh, *Mae, mse, rmse, coefficient of determination, adjusted r squared - which metric is better?* Dec. 2020. [Online]. Available: https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e.

[57] J. Brownlee, *Probabilistic model selection with aic, bic, and mdl*, Aug. 2020. [Online]. Available: https://machinelearningmastery.com/probabilistic-model-selection-measures/.

[58] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE transactions on dependable and secure computing*, vol. 1, no. 1, pp. 11–33, 2004.

[59] K. C. Joshi, R. Hersyandika, and R. V. Prasad, "Association, blockage, and handoffs in ieee 802.11ad-based 60-ghz picocells - a closer look," *IEEE Systems Journal*, vol. 14, pp. 2144–2153, 2020.

[60] R. Hersyandika, "Characterization of human blockage in 60 ghz communication," The Netherlands, 2016. [Online]. Available: http://resolver.tudelft.nl/uuid:2912a5b2-2cea-42ba-9ba4-1ca06b08d9d6.

[61] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, “The benefits of coding over routing in a randomized setting,” in *IEEE International Symposium on Information Theory, 2003. Proceedings.*, 2003, pp. 442–. DOI: `10.1109/ISIT.2003.1228459`.

[62] D. S. Lun, M. Médard, and R. Koetter, “Efficient operation of wireless packet networks using network coding,” 2005. [Online]. Available: `https://www.crab.rutgers.edu/~dslun/docs/iwct2005.pdf`.

[63] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000. DOI: `10.1109/18.850663`.

[64] R. Koetter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003. DOI: `10.1109/TNET.2003.818197`.

[65] S.-Y. Li, R. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003. DOI: `10.1109/TIT.2002.807285`.

[66] D. F. Pinto, “Network coding data planes with programmable switches,” Ph.D. dissertation, Faculdade de Ciências da Universidade de Lisboa, Lisboa, Portugal, 2017. [Online]. Available: `http://hdl.handle.net/10451/30687`.

[67] J. Heide, S. Shi, K. Fouli, M. Medard, and V. Chook, “Random linear network coding (rlnc)-based symbol representation,” *Working Draft, IETF Secretariat, Internet-Draft draft-heide-nwcrg-rlnc-02, July 2019*, 2018.

[68] D. F. V. Gonçalves, “Network coding switch,” Ph.D. dissertation, Faculdade de Ciências da Universidade de Lisboa, Lisboa, Portugal, 2019. [Online]. Available: `http://hdl.handle.net/10451/40494`.

[69] A. Cohen, D. Malak, V. B. Bracha, and M. Medard, *Adaptive causal network coding with feedback*, 2019. arXiv: `1905.02870 [cs.IT]`.

[70] J. Cloud, D. Leith, and M. Médard, “A coded generalization of selective repeat arq,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2155–2163. DOI: `10.1109/INFOCOM.2015.7218601`.

[71] A. Cohen, G. Thiran, V. B. Bracha, and M. Médard, “Adaptive causal network coding with feedback for multipath multi-hop communications,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7. DOI: `10.1109/ICC40277.2020.9149421`.

[72] M. Zhang, M. Mezzavilla, R. Ford, *et al.*, “Transport layer performance in 5g mmwave cellular,” in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016, pp. 730–735. DOI: `10.1109/INFCOMW.2016.7562173`.

[73] M. Dahhani, G. Jakllari, and A.-L. Beylot, “Association and reliability in 802.11ad networks: An experimental study,” in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, 2019, pp. 398–405. DOI: `10.1109/LCN44214.2019.8990684`.

[74] S. K. Saha, S. Aggarwal, R. Pathak, D. Koutsonikolas, and J. Widmer, “Musher: An agile multipath-tcp scheduler for dual-band 802.11 ad/ac wireless lans,” in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[75] *Software-defined networking (sdn) definition*, Jun. 2020. [Online]. Available: `https://opennetworking.org/sdn-definition/`.

[76] Z. Zaidi, V. Friderikos, Z. Yousaf, S. Fletcher, M. Dohler, and H. Aghvami, “Will sdn be part of 5g?” *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3220–3258, 2018. DOI: `10.1109/COMST.2018.2836315`.

[77] B. K. Thapa, B. Dikici, and J. Schönwälder, “Reactive forwarding applications in onos,” *no. January*, 2018. [Online]. Available: `https://www.thapabishwa.de/onos-report/paper.pdf`.

[78] U. Seidel, *An introduction to sdn " admin magazine.* [Online]. Available: `https://www.admin-magazine.com/Archive/2016/34/SDN-and-the-future-of-networking`.

[79] Mar. 2015. [Online]. Available: `https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf`.

[80] F. Kuliesius and M. Giedraitis, “Sdn/legacy hybrid network control system,” in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, 2019, pp. 504–509. DOI: `10.1109/ICUFN.2019.8806028`.

[81] X. Huang, S. Cheng, K. Cao, P. Cong, T. Wei, and S. Hu, “A survey of deployment solutions and optimization strategies for hybrid sdn networks,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1483–1507, 2019. DOI: `10.1109/COMST.2018.2871061`.

[82] S. K. Saha, V. V. Vira, A. Garg, and D. Koutsonikolas, “60 ghz multi-gigabit indoor wlans: Dream or reality?” *arXiv preprint arXiv:1509.04274*, 2015. [Online]. Available: `https://arxiv.org/pdf/1509.04274.pdf`.

[83] M. Kacou, V. Guillet, G. El Zein, and G. Zaharia, “Coverage and throughput analysis at 60 ghz for indoor wlan with indirect paths,” in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 1–5. DOI: `10.1109/PIMRC.2018.8580903`.

[84] A. Loch, G. Bielsa, and J. Widmer, “Practical lower layer 60 ghz measurements using commercial off-the-shelf hardware,” Oct. 2016. DOI: `10.1145/2980159.2980165`.

[85] Guillermo, “Analysis and performance improvement of consumer-grade millimeter wave wireless networks,” Ph.D. dissertation, Jul. 2019. [Online]. Available: `http://hdl.handle.net/20.500.12761/745`.

[86] Y. Zhu, Z. Zhang, Z. Marzi, *et al.*, “Demystifying 60ghz outdoor picocells,” ser. MobiCom ’14, Maui, Hawaii, USA: Association for Computing Machinery, 2014, pp. 5–16, ISBN: 9781450327831. DOI: `10.1145/2639108.2639121`. [Online]. Available: `https://doi.org/10.1145/2639108.2639121`.

[87] G. K. Tran, M. Nakamura, H. Nishiuchi, K. Sakaguchi, R. Santos, and K. Koslowski, "Outdoor experiment of mmwave meshed backhaul for realtime edge content delivery," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, 2019, pp. 1–6. DOI: 10.1109/WCNCW.2019.8902699.

[88] V. Va, T. Shimizu, G. Bansal, and R. W. H. Jr., "Millimeter wave vehicular communications: A survey," *Foundations and Trends in Networking*, vol. 10, no. 1, pp. 1–113, 2016, ISSN: 1554-057X. DOI: 10.1561/1300000054. [Online]. Available: http://dx.doi.org/10.1561/1300000054.

[89] W. Wu, Q. Shen, M. Wang, and X. Shen, "Performance analysis of ieee 802.11.ad downlink hybrid beamforming," May 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7996975.

[90] M. K. Haider and E. W. Knightly, "Mobility resilience and overhead constrained adaptation in directional 60 ghz wlans: Protocol design and system implementation," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '16, Paderborn, Germany: Association for Computing Machinery, 2016, pp. 61–70, ISBN: 9781450341844. DOI: 10.1145/2942358.2942380. [Online]. Available: https://doi.org/10.1145/2942358.2942380.

[91] S. Shao, H. Zhang, D. Koutsonikolas, and A. Khreishah, "Two-dimensional reduction of beam training overhead in crowded 802.11ad based networks," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 680–685. DOI: 10.1109/INFCOMW.2018.8407013.

[92] T. Gu, Z. Yang, D. Basu, and P. Mohapatra, "Beamsniff: Enabling seamless communication under mobility and blockage in 60 ghz networks," in *2019 IFIP Networking Conference (IFIP Networking)*, IEEE, 2019, pp. 1–9.

[93] M. Jasim, N. Siasi, A. Aldalbahi, and N. Ghani, "Soft self-handover scheme for mmwave communications," *2019 SoutheastCon*, pp. 1–6, 2019.

[94] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan, "60 ghz indoor networking through flexible beams: A link-level profiling," ser. SIGMETRICS '15, Portland, Oregon, USA: Association for Computing Machinery, 2015, pp. 71–84, ISBN: 9781450334860. DOI: 10.1145/2745844.2745858. [Online]. Available: https://doi.org/10.1145/2745844.2745858.

[95] A. Aldalbahi, F. Shahabi, and M. Jasim, "Instantaneous beam prediction scheme against link blockage in mmWave communications," *Applied Sciences*, vol. 11, no. 12, p. 5601, Jun. 2021. DOI: 10.3390/app11125601. [Online]. Available: https://doi.org/10.3390/app11125601.

[96] Z. M. Ali, A. Duel-Hallen, and H. D. Hallen, "Early warning of mmwave signal blockage and aoa transition using sub-6 ghz observations," *IEEE Communications Letters*, vol. 24, pp. 207–211, 2020.

[97] M. Alrabeiah and A. Alkhateeb, "Deep learning for mmwave beam and blockage prediction using sub-6ghz channels," *IEEE Transactions on Communications*, vol. PP, pp. 1–1, Jun. 2020. DOI: 10.1109/TCOMM.2020.3003670.

[98] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan, "60 ghz indoor networking through flexible beams: A link-level profiling," in *SIGMETRICS 2015*, 2015.

[99] M. Zarifneshat, L. Xiao, J. Tang, and X. Zhang, "Learning-based blockage prediction for robust links in dynamic millimeter wave networks," *Wireless Networks*, vol. 27, no. 7, pp. 4693–4714, 2021.

[100] J. Palacios, D. Steinmetzer, A. Loch, M. Hollick, and J. Widmer, "Adaptive codebook optimization for beam training on off-the-shelf ieee 802.11ad devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '18, New Delhi, India: Association for Computing Machinery, 2018, pp. 241–255, ISBN: 9781450359030. DOI: 10.1145/3241539.3241576. [Online]. Available: https://doi.org/10.1145/3241539.3241576.

[101] S. Aggarwal, Z. Kong, M. Ghoshal, Y. Hu, and D. Koutsonikolas, "Throughput prediction on 60 ghz mobile devices for high-bandwidth, latency-sensitive applications," in Mar. 2021, pp. 513–528, ISBN: 978-3-030-72581-5. DOI: 10.1007/978-3-030-72582-2_30.

[102] M. Narasimha and H. Bagheri, "Network coding applications for 5g millimeter-wave communications," *ArXiv*, vol. abs/1512.03031, 2015.

[103] M. Drago, T. Azzino, M. Polese, Č. Stefanović, and M. Zorzi, "Reliable video streaming over mmwave with multi connectivity and network coding," in *2018 International Conference on Computing, Networking and Communications (ICNC)*, 2018, pp. 508–512. DOI: 10.1109/ICCNC.2018.8390387.

[104] M. V. Pedersen, J. Heide, and F. H. P. Fitzek, "Kodo: An open and research oriented network coding library," in *NETWORKING 2011 Workshops*, V. Casares-Giner, P. Manzoni, and A. Pont, Eds., Springer Berlin Heidelberg, 2011, pp. 145–152, ISBN: 978-3-642-23041-7.

[105] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan, "60 ghz indoor networking through flexible beams: A link-level profiling," *SIGMETRICS Perform. Eval. Rev.*, vol. 43, no. 1, pp. 71–84, Jun. 2015, ISSN: 0163-5999. DOI: 10.1145/2796314.2745858. [Online]. Available: https://doi.org/10.1145/2796314.2745858.

[106] A. R. Broadnet and J. Brady. [Online]. Available: https://www.bcba.ca/application/files/5515/5812/9724/CCS_Metnet__Broadnet.pdf.

[107] *Why use mongodb and when to use it?* [Online]. Available: https://www.mongodb.com/why-use-mongodb.

[108] V. GUEANT, *Iperf - the ultimate speed test tool for tcp, udp and sctptest the limits of your network + internet neutrality test.* [Online]. Available: https://iperf.fr/.

[109] J. Hale, *Scale, standardize, or normalize with scikit-learn*, Sep. 2021. [Online]. Available: https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02.

[110] B. Roy, *All about feature scaling*, Apr. 2020. [Online]. Available: https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35.

[111] A. Kumar, *Svm rbf kernel parameters with code examples*, Jul. 2020. [Online]. Available: https://vitalflux.com/svm-rbf-kernel-parameters-code-sample/.

[112] S. Misra and H. Li, "Chapter 9 - noninvasive fracture characterization based on the classification of sonic wave travel times," in *Machine Learning for Subsurface Characterization*, S. Misra, H. Li, and J. He, Eds., Gulf Professional Publishing, 2020, pp. 243–287, ISBN: 978-0-12-817736-5. DOI: https://doi.org/10.1016/B978-0-12-817736-5.00009-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128177365000090.

[113] T. Yiu, *Understanding random forest*, Sep. 2021. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

[114] B. I. C. Education, *What is random forest?* [Online]. Available: https://www.ibm.com/cloud/learn/random-forest.

[115] *Multilayer perceptron.* [Online]. Available: https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron.

[116] J. Brownlee, *How to choose an activation function for deep learning*, Jan. 2021. [Online]. Available: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/.

[117] *Sklearn neural_network.mlpclassifier.* [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.

[118] L. Karlsson and O. Bonde, *A comparison of selected optimization methods for neural networks*, Sweden, 2020. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1438308/FULLTEXT01.pdf.

[119] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. J. Király, "Sktime: A unified interface for machine learning with time series," *arXiv preprint arXiv:1909.07872*, 2019.

[120] *Autoarima - sktime documentation.* [Online]. Available: https://www.sktime.org/en/v0.5.2/api_reference/modules/auto_generated/sktime.forecasting.arima.AutoARIMA.html.

[121] *Recurrent neural networks (rnn) with keras.* [Online]. Available: https://www.tensorflow.org/guide/keras/rnn.

[122] *Autonomio talos [computer software].* [Online]. Available: http://github.com/autonomio/talos.

[123] J. Brownlee, *A gentle introduction to dropout for regularizing deep neural networks*, Aug. 2019. [Online]. Available: `https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/`.

[124] S. Semeniuta, A. Severyn, and E. Barth, "Recurrent dropout without memory loss," *arXiv preprint arXiv:1603.05118*, 2016.

[125] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration*, vol. 14, no. 14, p. 527, 2008.

[126] *Ns-3 tutorial.* [Online]. Available: `https://www.nsnam.org/docs/tutorial/singlehtml/`.

[127] *High-fidelity implementation of the ieee 802.11ad/ay standards in network simulator ns-3.* [Online]. Available: `https://github.com/wigig-tools/wigig-module`.

[128] H. Assasa, J. Widmer, T. Ropitault, and N. Golmie, "Enhancing the ns-3 ieee 802.11ad model fidelity: Beam codebooks, multi-antenna beamforming training, and quasi-deterministic mmwave channel," ser. WNS3 2019, Florence, Italy: Association for Computing Machinery, 2019, pp. 33–40, ISBN: 9781450371407. DOI: `10.1145/3321349.3321354`. [Online]. Available: `https://doi.org/10.1145/3321349.3321354`.

[129] A. Bodi, S. Blandino, N. Varshney, and T. Ropitault, *Q-d realization software*, Aug. 2021. [Online]. Available: `https://github.com/wigig-tools/qd-realization`.

[130] H. Assasa, N. Grosheva, and S. Blandino, *The nist q-d interpreter software*, Oct. 2021. [Online]. Available: `https://github.com/wigig-tools/qd-interpreter`.

[131] A. Maltsev, A. Pudeyev, I. Karls, *et al.*, "Quasi-deterministic approach to mmwave channel modeling in a non-stationary environment," in *2014 IEEE Globecom Workshops (GC Wkshps)*, 2014, pp. 966–971. DOI: `10.1109/GLOCOMW.2014.7063558`.

[132] Steinwurf, *Steinwurf/kodo-rlnc-c: High-level c bindings for the kodo-rlnc network coding library.* [Online]. Available: `https://github.com/steinwurf/kodo-rlnc-c`.

Appendices

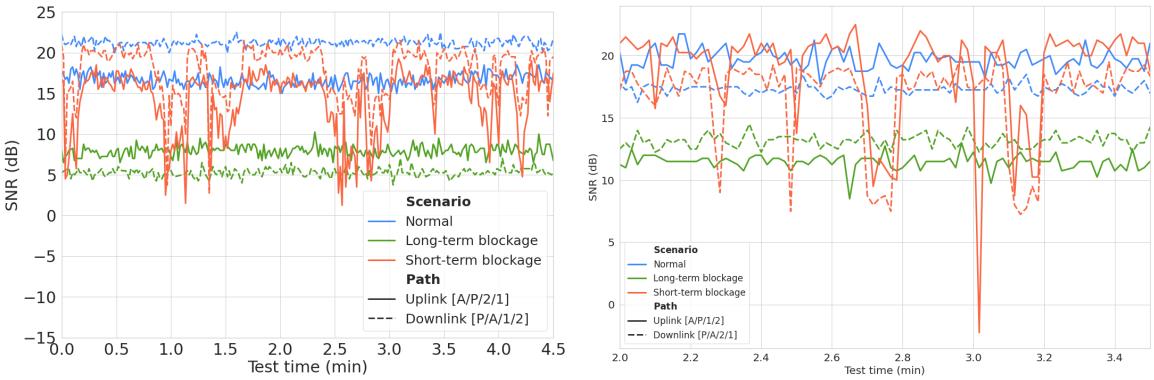
This section presents additional results that further detail the findings described in the previous chapters. It is organised into three main sections. Firstly, some results are presented to complement the multi-layer analysis. In the second section, additional results obtained during the development of the classifier of the quality of a mmWave link are presented. In the last section, results regarding the prediction of the main KPIs of a link are presented.

The main takeaways from the analysis of these results have been described throughout the thesis in their respective chapters.

A Multi-layer Analysis

In the following section results that complement the multi-layer analysis of the chapter 4 are presented.

A.1 Signal to Noise Ratio



(a) Path P/A/1/2.

(b) Path P/A/2/1.

Figure A.1. SNR of two mmWave paths under normal, long-term blocked and short-term blocked scenarios.

A.2 Packet Error Rate

Table A.1. Statistics of the measured PER for the 4 uplink paths between node P and A with different MCS modes (fixed and automatic) for the three defined scenarios.

Path	MCS mode	Scenario	Packet Error (%)			
			Min	Mean	Max	Std dev
A/P/1/1	1	Normal	0	0	0	0
		Static	0	1.3	46.7	5.1
	3	Normal	0	0	0	0
		Static	0	10.2	100	24.7
	5	Normal	0	0	0	0
	7	Normal	0	0	0	0
	9	Normal	0	0.01	0.3	0.003
	Auto	Normal	0	1.8	11.4	1.2
		Static	0	6.4	64.7	9.3
		short-term	0	2.4	88.2	4.8
A/P/1/2	1	Normal	0	0	0	0
		Static	0	0.01	3.6	0.16
	3	Normal	0	0	0	0
		Static	0	0.003	0.5	0.002
	5	Normal	0	0	0	0
		Static	0	0.03	0.7	0.09
	7	Normal	0	0	0	0
		Static	0	0.18	1.5	0.25
	9	Normal	0	0.02	0.3	0.04
		Static	0	1.2	5.4	1.3
	Auto	Normal	0	1.8	15.8	1.4
		Static	0	3.0	25.6	3.1
		short-term	0	1.7	18.0	1.7
A/P/2/1	1	Normal	0	0	0	0
		Static	0	0.002	2.5	0.06
	3	Normal	0	0	0	0
		Static	0	0.02	3.23	0.15
	5	Normal	0	0	0	0
		Static	0	34.5	99.2	36.6
	7	Normal	0	0.0003	1.8	0.006
		Static	0	66.8	100	33.4
	9	Normal	0	0.12	1.22	0.18
	Auto	Normal	0	1.6	16.7	1.3
		Static	0	1.05	17.0	1.6
		short-term	0	2.0	20.8	2.3
A/P/2/2	1	Normal	0	0	0	0
		Static	0	0	0	0
	3	Normal	0	0	0	0
		Static	0	1.4	96.4	8.1
	5	Normal	0	0	0	0
		Static	0	49.1	100	43.2
	7	Normal	0	0	0	0
		Static	0	62.1	100	36.6
	9	Normal	0	0.06	0.7	0.1
	Auto	Normal	0	1.6	19.0	1.4
		Static	0	2.8	64.3	3.7
		short-term	0	1.9	17.3	1.9

Table A.2. Average RTT measured for all paths between node A and P under normal operation.

MCS mode	**Link**	**Average RTT (ms)**				**MCS**	
		Mean	**Std dev**	**Min**	**Max**	**MCS range**	**MCS mean**
MCS 1	A/P/1/1	40.83	7.79	19.17	56.46	-	-
	A/P/1/2	41.04	7.61	24.67	54.15	-	-
	A/P/2/1	41.67	7.93	18.66	58.75	-	-
	A/P/2/2	41.23	7.40	24.67	54.15	-	-
MCS 3	A/P/1/1	9.98	2.87	3.45	17.15	-	-
	A/P/1/2	9.40	2.65	4.43	14.86	-	-
	A/P/2/1	9.54	2.95	3.00	16.49	-	-
	A/P/2/2	9.79	2.90	3.10	16.37	-	-
MCS 5	A/P/1/1	2.93	0.75	2.93	5.25	-	-
	A/P/1/2	2.90	0.58	1.56	4.13	-	-
	A/P/2/1	3.11	0.75	1.90	5.53	-	-
	A/P/2/2	3.09	0.72	1.47	5.83	-	-
MCS 7	A/P/1/1	2.52	0.64	1.27	4.17	-	-
	A/P/1/2	2.62	0.62	1.40	4.42	-	-
	A/P/2/1	2.47	0.59	1.50	4.50	-	-
	A/P/2/2	2.56	0.58	1.28	4.42	-	-
MCS 9	A/P/1/1	2.48	0.66	1.28	4.69	-	-
	A/P/1/2	2.38	0.59	1.23	4.68	-	-
	A/P/2/1	2.44	0.60	1.25	4.21	-	-
	A/P/2/2	2.37	0.56	1.36	3.52	-	-
MCS Auto	A/P/1/1	2.83	1.15	1.19	6.86	8 - 9	8.99
	A/P/1/2	2.95	1.60	1.40	9.72	8 - 9	8.98
	A/P/2/1	2.65	0.98	1.15	7.43	8 - 9	8.99
	A/P/2/2	2.65	1.34	2.65	12.26	8 - 9	8.99

A.3 Round-trip Time

Table A.3. Average RTT measured for all paths between node A and P under long-term blockage.

MCS mode	Link	Average RTT (ms)				MCS	
		Mean	std dev	Min	Max	MCS range	Mean MCS
MCS 1	A/P/1/1	26.64	4.96	16.69	38.44	–	–
	A/P/1/2	42.85	134.53	14.51	1324.96	–	–
	A/P/2/1	27.86	7.38	9.31	48.89	–	–
	A/P/2/2	28.14	6.31	11.16	41.74	–	–
MCS 3	A/P/1/1	167.88	230.26	0.72	819.34	–	–
	A/P/1/2	2.37	0.66	1.36	4.25	–	–
	A/P/2/1	2.34	0.62	1.26	4.53	–	–
	A/P/2/2	6.79	16.95	1.07	114.54	–	–
MCS 5	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	3.63	17.42	0.44	147.59	–	–
	A/P/2/1	208.18	138.27	5.66	554.03	–	–
	A/P/2/2	21.80	35.40	0.51	164.84	–	–
MCS 7	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	1.34	0.33	0.81	2.71	–	–
	A/P/2/1	2.11	5.92	0.46	46.29	–	–
	A/P/2/2	3.80	8.96	0.49	87.32	–	–
MCS Auto	A/P/1/1	1.13	1.37	0.52	8.35	6 – 9	7
	A/P/1/2	5.73	3.88	1.04	20.81	8 – 9	8.98
	A/P/2/1	5.51	15.34	0.46	133.99	2 – 7	4.57
	A/P/2/2	8.90	5.98	1.36	29.89	1 – 6	3

Table A.4. Average RTT measured for all paths between node A and P under short-term blockage.

MCS mode	Link	Average RTT (ms)				MCS	
		Mean	std dev	Min	Max	Range	Mean
MCS Auto	A/P/1/1	7.04	6.73	1.84	34.17	1 – 9	8.39
	A/P/1/2	9.59	9.07	1.43	57.02	1 – 9	8.72
	A/P/2/1	6.76	0.89	1.55	27.60	1 – 9	8.81
	A/P/2/2	25.30	6.46	21.18	93.15	3 – 9	8.99

Table A.5. Maximum RTT measured for all paths between node A and P under normal operation.

MCS mode	Link	Maximum RTT (ms)				MCS	
		Mean	std dev	Min	Max	Range	Mean
MCS 1	A/P/1/1	57.27	10.64	34.59	74.91	–	–
	A/P/1/2	112.5	478.62	36.29	4677.09	–	–
	A/P/2/1	60.51	13.9	15.2	87.8	–	–
	A/P/2/2	61.04	13.04	18.21	81.53	–	–
MCS 3	A/P/1/1	402.37	491.59	1.15	1846.55	–	–
	A/P/1/2	6.6	2.78	2.02	12.4	–	–
	A/P/2/1	7.03	2.94	2.75	18.51	–	–
	A/P/2/2	42.79	143.1	1.94	1066.15	–	–
MCS 5	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	18.57	121.93	0.66	1026.44	–	–
	A/P/2/1	684.92	323.73	13.49	1360.27	–	–
	A/P/2/2	119.63	205.15	0.58	1009.97	–	–
MCS 7	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	3.4	1.86	1.2	13.33	–	–
	A/P/2/1	9.93	44.46	0.54	351.2	–	–
	A/P/2/2	16.53	47.25	0.61	401.3	–	–
MCS 9	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	4.91	4.08	1.3	22.7	–	–
	A/P/2/1	–	–	–	–	–	–
	A/P/2/2	–	–	–	–	–	–
MCS Auto	A/P/1/1	5.45	13.53	0.63	78.42	6 – 9	7.26
	A/P/1/2	29.41	20.48	1.74	106.13	8 – 9	8.98
	A/P/2/1	32.93	105.73	0.59	897.83	2 – 7	4.57
	A/P/2/2	43	33.29	3.21	246.91	1 – 6	2.98

Table A.6. Maximum RTT measured for all paths between node A and P under long-term blockage.

MCS mode	**Link**	**Maximum RTT (ms)**				**MCS**	
		Mean	**std dev**	**Min**	**Max**	**Range**	**Mean**
MCS 1	A/P/1/1	57.27	10.64	34.59	74.91	–	–
	A/P/1/2	112.5	478.62	36.29	4677.09	–	–
	A/P/2/1	60.51	13.9	15.2	87.8	–	–
	A/P/2/2	61.04	13.04	18.21	81.53	–	–
MCS 3	A/P/1/1	402.37	491.59	1.15	1846.55	–	–
	A/P/1/2	6.6	2.78	2.02	12.4	–	–
	A/P/2/1	7.03	2.94	2.75	18.51	–	–
	A/P/2/2	42.79	143.1	1.94	1066.15	–	–
MCS 5	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	18.57	121.93	0.66	1026.44	–	–
	A/P/2/1	684.92	323.73	13.49	1360.27	–	–
	A/P/2/2	119.63	205.15	0.58	1009.97	–	–
MCS 7	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	3.4	1.86	1.2	13.33	–	–
	A/P/2/1	9.93	44.46	0.54	351.2	–	–
	A/P/2/2	16.53	47.25	0.61	401.3	–	–
MCS 9	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	4.91	4.08	1.3	22.7	–	–
	A/P/2/1	–	–	–	–	–	–
	A/P/2/2	–	–	–	–	–	–
MCS Auto	A/P/1/1	5.45	13.53	0.63	78.42	6 – 9	7.26
	A/P/1/2	29.41	20.48	1.74	106.13	8 – 9	8.98
	A/P/2/1	32.93	105.73	0.59	897.83	2 – 7	4.57
	A/P/2/2	43	33.29	3.21	246.91	1 – 6	2.98

Table A.7. Maximum RTT measured for all paths between node A and P under short-term blockage.

MCS mode	**Link**	**Maximum RTT (ms)**				**MCS**	
		Mean	**std dev**	**Min**	**Max**	**Range**	**Mean**
MCS Auto	A/P/1/1	34.45	40.04	3.23	245.70	1 – 9	8.39
	A/P/1/2	35.50	66.94	2.80	529.78	1 – 9	8.72
	A/P/2/1	27.92	28.97	2.92	92.22	1 – 9	8.81
	A/P/2/2	67.08	60.68	9.14	283.11	3 – 9	8.99

A.4 Packet Loss

Table A.8. TCP packet loss measured for all paths between node A and P under normal operation.

MCS mode	**Link**	**Maximum RTT (ms)**				**MCS**	
		Mean	**std dev**	**Min**	**Max**	**Range**	**Mean**
MCS 1	A/P/1/1	57.27	10.64	34.59	74.91	–	–
	A/P/1/2	112.5	478.62	36.29	4677.09	–	–
	A/P/2/1	60.51	13.9	15.2	87.8	–	–
	A/P/2/2	61.04	13.04	18.21	81.53	–	–
MCS 3	A/P/1/1	402.37	491.59	1.15	1846.55	–	–
	A/P/1/2	6.6	2.78	2.02	12.4	–	–
	A/P/2/1	7.03	2.94	2.75	18.51	–	–
	A/P/2/2	42.79	143.1	1.94	1066.15	–	–
MCS 5	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	18.57	121.93	0.66	1026.44	–	–
	A/P/2/1	684.92	323.73	13.49	1360.27	–	–
	A/P/2/2	119.63	205.15	0.58	1009.97	–	–
MCS 7	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	3.4	1.86	1.2	13.33	–	–
	A/P/2/1	9.93	44.46	0.54	351.2	–	–
	A/P/2/2	16.53	47.25	0.61	401.3	–	–
MCS 9	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	4.91	4.08	1.3	22.7	–	–
	A/P/2/1	–	–	–	–	–	–
	A/P/2/2	–	–	–	–	–	–
MCS Auto	A/P/1/1	5.45	13.53	0.63	78.42	8 – 9	8.99
	A/P/1/2	29.41	20.48	1.74	106.13	8 – 9	8.98
	A/P/2/1	32.93	105.73	0.59	897.83	8 – 9	8.99
	A/P/2/2	43	33.29	3.21	246.91	8 – 9	8.99

Table A.9. TCP packet loss measured for all paths between node A and P under long-term blockage.

	Link	Mean	Std dev	Min	Max	MCS Range	MCS mean	Number of samples
MCS 1	A/P/1/1	1.74	0.28	0.96	2.74	–	–	590
	A/P/1/2	1.72	0.29	0.87	2.61	–	–	833
	A/P/2/1	1.77	0.30	1.06	2.93	–	–	830
	A/P/2/2	1.76	0.32	0.89	2.83	–	–	826
MCS 3	A/P/1/1	12.45	7.49	0.01	21.77	–	–	452
	A/P/1/2	0.04	0.04	0.006	0.27	–	–	832
	A/P/2/1	0.04	0.037	0.008	0.29	–	–	832
	A/P/2/2	0.28	0.56	0.008	4.74	–	–	833
MCS 5	A/P/1/1	–	–	–	–	–	–	0
	A/P/1/2	10.66	9.28	1.10	66.7	–	–	836
	A/P/2/1	0.04	0.09	0.006	0.953	–	–	823
	A/P/2/2	49.20	36.46	2.86	100	–	–	839
MCS 7	A/P/1/1	–	–	–	–	–	–	0
	A/P/1/2	97.53	9.09	37.5	100	–	–	872
	A/P/2/1	0.04	0.09	0.006	0.87	–	–	826
	A/P/2/2	60.7	45.17	6.01	100	–	–	839
MCS Auto	A/P/1/1	0.12	0.33	0.007	2.22	6 – 9	7	832
	A/P/1/2	0.19	0.35	0.008	2.59	8 – 9	8.98	834
	A/P/2/1	0.42	0.31	2.51	2.5	2 – 7	4.57	834
	A/P/2/2	0.7	0.40	0.29	2.36	1 – 6	3	836

Table A.10. TCP Packet loss measured for all paths between node A and P under short-term blockage.

	Link	Mean	std dev	Min	Max	MCS Range	MCS mean	Number of samples
MCS Auto	A/P/1/1	1.01	0.78	0.02	3.95	1 – 9	8.39	284
	A/P/1/2	0.67	0.62	0.01	2.41	1 – 9	8.73	285
	A/P/2/1	1.05	0.94	0.01	8.68	1 – 9	8.81	683
	A/P/2/2	1.23	0.66	0.07	3.9	3 – 9	8.90	344

A.5 TCP throughput

Table A.11. Throughput measured for all paths between node A and P under normal operation.

MCS mode	Link	TCP throughput				MCS	
		Mean	**std dev**	**Min**	**Max**	**Range**	**Mean**
MCS 1	A/P/1/1	74.39	37.31	23.74	238.51	–	–
	A/P/1/2	67.39	19.65	10.30	101.69	–	–
	A/P/2/1	66.42	19.80	17.29	101.55	–	–
	A/P/2/2	64.12	19.09	15.95	108.42	–	–
MCS 3	A/P/1/1	472.53	17.03	416.03	498.39	–	–
	A/P/1/2	469.41	18.18	407.71	499.27	–	–
	A/P/2/1	467.79	18.84	423.51	499.08	–	–
	A/P/2/2	472.61	18.22	389.35	499.75	–	–
MCS 5	A/P/1/1	496.32	6.04	471.78	499.88	–	–
	A/P/1/2	495.75	11.10	387.82	499.90	–	–
	A/P/2/1	497.19	4.67	477.26	499.89	–	–
	A/P/2/2	496.44	6.67	457.32	499.85	–	–
MCS 7	A/P/1/1	496.47	5.42	473.16	499.88	–	–
	A/P/1/2	496.20	8.29	441.01	499.87	–	–
	A/P/2/1	496.24	6.64	459.82	499.88	–	–
	A/P/2/2	497.00	5.21	470.81	499.90	–	–
MCS Auto	A/P/1/1	495.12	7.94	451.38	499.91	8 - 9	8.99
	A/P/1/2	494.70	8.15	453.63	499.89	8 – 9	8.98
	A/P/2/1	495.23	7.27	452.59	499.87	8 – 9	8.99
	A/P/2/2	495.68	7.63	461.62	499.89	8 – 9	8.99

Table A.12. Throughput measured for all paths between node A and P under long-term blockage.

MCS mode	Link	TCP throughput				MCS	
		Mean	std dev	Min	Max	Range	Mean
MCS 1	A/P/1/1	252.42	6.92	222.40	265.90	–	–
	A/P/1/2	238.52	5.12	225.42	250.05	–	–
	A/P/2/1	239.47	5.13	226.80	253.19	–	–
	A/P/2/2	238.82	5.62	222.30	251.31	–	–
MCS 3	A/P/1/1	64.55	130.45	4.20	444.74	–	–
	A/P/1/2	499.74	0.18	498.50	499.89	–	–
	A/P/2/1	499.69	0.69	491.94	499.90	–	–
	A/P/2/2	483.37	62.03	19.89	499.88	–	–
MCS 5	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	496.81	34.64	90.43	499.90	–	–
	A/P/2/1	13.76	17.11	0.04	115.61	–	–
	A/P/2/2	1.06	2.14	0.00	14.21	–	–
MCS 7	A/P/1/1	–	–	–	–	–	–
	A/P/1/2	499.72	0.43	495.55	499.91	–	–
	A/P/2/1	0.00	0.01	0.00	0.10	–	–
	A/P/2/2	2.38	2.31	0.00	4.69	–	–
MCS Auto	A/P/1/1	477.51	23.50	409.92	499.91	6 – 9	7
	A/P/1/2	409.86	35.56	329.75	487.57	8 – 9	8.98
	A/P/2/1	495.79	10.44	433.83	499.89	2 – 7	4.57
	A/P/2/2	328.76	59.99	213.14	488.00	1 – 6	3

Table A.13. Throughput measured for all paths between node A and P under short-term blockage.

MCS mode	Link	TCP throughput				MCS	
		Mean	std dev	Min	Max	Range	Mean
MCS Auto	A/P/1/1	417.09	90.55	145.48	499.81	1 – 9	8.39
	A/P/1/2	453.09	42.13	229.46	499.91	1 – 9	8.73
	A/P/2/1	432.21	94.60	167.11	499.87	1 – 9	8.81
	A/P/2/2	273.11	82.15	75.05	365.36	3 – 9	8.90

B Wireless Link Quality Classifier

In the following section results that complement the wireless link quality classification analysis of chapter 5 section 5.3.1 are presented, including intermediate results obtained with the SVM, Random Forest and MLP classifiers.

Table B.14. Settings found for the top three SVM models for windows of size 3, 5, 7 and 10 seconds.

Window Size	**Best Hyperparameters**	**Mean fit time**	**F1-Score**			
			Mean		**std dev**	
			Train	**Test**	**Train**	**Test**
3 s	kernel: rbf gamma: 3.381 C: 81.249	3.276	0.988	0.971	0.001	0.002
	kernel: rbf gamma: 7.409 C: 14.940	5.441	0.987	0.971	0.001	0.004
	kernel: rbf gamma: 3.708 C: 10.717	3.581	0.979	0.968	0.001	0.003
5 s	kernel: rbf gamma: 5.122 C 18.051	1.951	0.987	0.967	0.000	0.006
	kernel: rbf gamma: 0.306 C: 72.397	2.430	0.961	0.951	0.003	0.004
	kernel: poly gamma: 0.867 degree: 5 C: 31.116	16.161	0.990	0.950	0.001	0.006
7 s	kernel: rbf gamma: 3.460 C: 93.744	0.756	0.993	0.971	0.001	0.002
	kernel: rbf gamma: 4.358 C: 12.365	0.852	0.984	0.969	0.001	0.003
	kernel: rbf gamma: 1.813 C: 31.696	0.604	0.983	0.967	0.000	0.003
10 s	kernel: rbf gamma: 4.161 C: 12.889	0.695	0.990	0.968	0.001	0.007
	kernel: rbf gamma: 2.181 C: 37.771	0.521	0.990	0.966	0.001	0.005
	kernel: rbf gamma: 1.989 C: 10.093	0.533	0.983	0.963	0.001	0.005

Table B.15. Settings found for the top three Random Forest models for windows of size 3, 5, 7 and 10 seconds.

Window Size	Best Hyperparameters	Mean fit time	Test F1-Score	
			Mean	std dev
3 s	warm_start: False, n_estimators: 938 min_samples_split: 2, min_samples_leaf: 1 max_features: auto, max_depth: 400 bootstrap: False	55.7 s	0.982	0.000
	warm_start: False, n_estimators: 4779 min_samples_split: 2, min_samples_leaf: 1 max_features: 10, max_depth: 740 bootstrap: False	10.4 min	0.982	0.000
	warm_start: True, n_estimators: 109 min_samples_split: 16, min_samples_leaf: 1 max_features: 7, max_depth: 340 bootstrap: False	14.6 s	0.980	0.001
5 s	warm_start: False, n_estimators: 4862 min_samples_split: 2, min_samples_leaf: 1 max_features: 10, max_depth: 190 bootstrap: True	5.9 min	0.973	0.005
	warm_start: False, n_estimators: 816 min_samples_split: 10, min_samples_leaf: 2 max_features: 12, max_depth: 840 bootstrap: True	1.13 min	0.969	0.005
	warm_start: True, n_estimators: 1978 min_samples_split: 16, min_samples_leaf: 1 max_features: 15, max_depth: 960 bootstrap: False	5.5 min	0.963	0.003
7 s	warm_start: False, n_estimators: 1046 min_samples_split: 10, min_samples_leaf: 9 max_features: 15, max_depth: 470 bootstrap: True	1.14 min	0.947	0.005
	warm_start: True, n_estimators: 4705 min_samples_split: 57, min_samples_leaf: 5 max_features: 7, max_depth: 520 bootstrap: True	2.75 min	0.946	0.006
	warm_start: True, n_estimators: 1909 min_samples_split: 89, min_samples_leaf: 11 max_features: 10, max_depth: 690 bootstrap: False	2.17 min	0.944	0.005
10 s	warm_start: False, n_estimators: 1046 min_samples_split: 10, min_samples_leaf: 9 max_features: 15, max_depth: 470 bootstrap: True	46.9 s	0.938	0.001
	warm_start: True, n_estimators: 4705 min_samples_split: 57, min_samples_leaf: 5 max_features: 7, max_depth: 520 bootstrap: True	1.9 min	0.935	0
	warm_start: True, n_estimators: 1909 min_samples_split: 89, min_samples_leaf: 11 max_features: 10, max_depth: 690 bootstrap: False	1.49 min	0.931	0.002

Table B.16. Settings found for the top three MLPC models for windows of size 3, 5, 7 and 10 seconds.

Window Size	**Mean Fit Time**	**Hyper-parameters**								**F1 Score**			
		Solver	**Hidden Layers Neurons**	**Batch Size**	**Epochs**	**Activation Function**	**Alpha**	**Warm Start**	**Shuffle Data**	**Initial Learning Rate**	**Train Mean**	**Test Mean**	**Train std dev**	**Test std dev**
3 s	16.6 s	Adam	(50, 50)	128	60	Tanh	0.001	True	True	0.003	0.963	0.966	0.005	0.060
	11.3 s		(32, 32, 32)	256	100	Tanh	0	True	True	0.01	0.963	0.955	0.002	0.005
	1.8 min		128	8	60	ReLu	0	False	False	0.001	0.961	0.954	0.002	0.007
5 s	4 min		(128, 128, 128)	8	50	ReLu	1e-5	False	True	0.0001	0.965	0.953	0.000	0.006
	3.9 min		(32, 32, 32)	2	100	Tanh	1e-5	True	True	0.001	0.964	0.953	0.006	0.002
	14.2 s		(128, 128)	64	60	ReLu	0	True	False	0.005	0.960	0.949	0.004	0.002
7 s	15.0 s		(128, 128)	512	100	Tanh	0	True	False	0.01	0.965	0.954	0.008	0.002
	50.9 s		(64, 64)	8	60	ReLu	0	False	True	0.001	0.963	0.950	0.008	0.001
	2.6 min		128	2	50	ReLu	1e-5	True	False	0.0005	0.954	0.943	0.008	0.002
10 s	33.7 s		(64, 64, 64)	4	60	ReLu	1e-5	True	True	0.001	0.965	0.953	0.002	0.012
	1.4 min		(128, 128)	4	80	ReLu	0	True	True	0.001	0.971	0.951	0.004	0.006
	4.9 s		(128, 128, 128)	Auto	60	ReLu	0	True	False	0.005	0.967	0.949	0.005	0.009

C KPIs forecasting

In the following section results that complement the KPIs forecasting analysis of the ARIMA and LSTM-based RNN models, of chapter 5 section 5.3.2, are presented. Examples of predictions made by the LSTM neural network for two other mmWave paths are also shown.

Table C.17. Results obtained with the best performing ARIMA models for predicting the SNR of each mmWave path established.

Path ID	**Path Name**	**Configurations Order**			**Metrics**
		p	**d**	**q**	**AIC**
1	P/A/0/2	1	1	1	6929.644
2	A/P/2/0	0	1	2	8008.633
3	A/P/1/1	2	1	1	15620.335
4	P/A/1/1	0	1	2	9801.333
5	A/P/2/1	1	1	3	17194.189
6	P/A/1/2	1	1	1	17102.137
7	A/P/1/2	3	1	1	24176.161
8	P/A/2/1	1	1	2	23757.950
9	A/P/2/2	0	1	2	16720.983
10	P/A/2/2	0	1	2	16002.159
11	P/A/3/1	2	1	2	13809.592
12	A/P/1/3	0	1	2	10188.548
13	P/B/1/1	0	1	1	2039.078
14	B/P/1/1	0	1	1	680.995
15	B/P/2/1	4	1	5	9437.029
16	P/B/1/2	3	1	1	9185.718
17	B/P/1/2	1	1	1	5130.811
18	P/B/2/1	4	1	3	6518.607
19	B/P/2/2	0	0	0	1027.869
20	P/B/2/2	0	0	0	944.172

Table C.18. Mean Squared Error obtained with the best performing LSTM model.

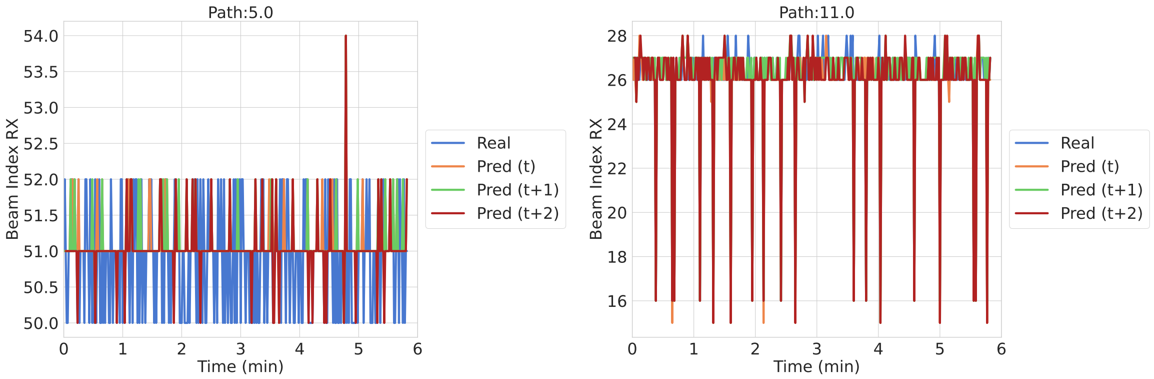
		MSE (t)	**MSE (t+1)**	**MSE (t+2)**
Physical metrics	Beam Index RX	24.09 ± 95.52	25.18 ± 101.62	25.02 ± 100.86
	Beam Index TX	1.38 ± 1.84	2.30 ± 4.32	1.66 ± 2.64
	MCS RX	0.27 ± 0.30	0.31 ± 0.35	0.35 ± 0.43
	MCS TX	0.23 ± 0.28	0.27 ± 0.33	0.29 ± 0.32
	Mbps TX	4409.60 ± 4415.92	5068.61 ± 5215.88	5748.62 ± 5835.69
	Mbps TX (Max)	4729.37 ± 4803.50	5432.87 ± 5563.34	5950.92 ± 6020.35
	Packet Error (%)	18.50 ± 27.85	21.61 ± 28.07	24.40 ± 33.70
	RCPI (dBm)	0.81 ± 0.52	0.90 ± 0.75	1.01 ± 1.07
	SNR (dB)	0.80 ± 0.36	0.93 ± 0.60	1.02 ± 0.80
Network metrics	Sent Mbps	295.44 ± 151.94	448.58 ± 268.61	615.03 ± 383.13
	Received Mbps	297.29 ± 151.70	450.04 ± 266.60	620.93 ± 387.39
	TCP retransmissions	12411.95 ± 10680.55	19932.62 ± 17659.82	28143.20 ± 25000.92
	RTT (Mean)	1332.91 ± 2195.41	4175.40 ± 12178.61	7148.64 ± 23314.91
	RTT (Max)	7876.01 ± 12255.11	17183.57 ± 38121.04	27151.44 ± 70721.41
	RTT (Min)	86.65 ± 259.23	273.00 ± 793.06	489.52 ± 1514.44
	Packet Loss (%)	7.92 ± 6.49	13.82 ± 13.36	17.76 ± 17.64

Table C.19. Root Mean Squared Error obtained with the best performing LSTM model.

		RMSE (t)	**RMSE (t+1)**	**RMSE (t+2)**
Physical metrics	Beam Index RX	2.12 ± 4.43	2.08 ± 4.57	2.05 ± 4.56
	Beam Index TX	1.02 ± 0.59	1.18 ± 0.96	1.08 ± 0.71
	MCS RX	0.47 ± 0.22	0.49 ± 0.26	0.51 ± 0.29
	MCS TX	0.43 ± 0.22	0.46 ± 0.24	0.48 ± 0.23
	Mbps TX	48.42 ± 45.45	51.53 ± 49.13	54.67 ± 52.53
	Mbps TX (Max)	50.24 ± 46.96	53.49 ± 50.72	55.69 ± 53.38
	Packet Error (%)	3.22 ± 2.85	3.61 ± 2.92	3.73 ± 3.24
	RCPI (dBm)	0.86 ± 0.28	0.88 ± 0.35	0.92 ± 0.42
	SNR (dB)	0.88 ± 0.19	0.92 ± 0.29	0.95 ± 0.35
Network metrics	Sent Mbps	16.60 ± 4.46	20.24 ± 6.25	23.63 ± 7.54
	Received Mbps	16.67 ± 4.41	20.29 ± 6.18	23.75 ± 7.54
	TCP retransmissions	100.98 ± 47.06	127.50 ± 60.63	150.94 ± 73.22
	RTT (Mean)	28.29 ± 23.08	39.73 ± 50.96	47.14 ± 70.19
	RTT (Max)	70.90 ± 53.37	93.91 ± 91.45	108.65 ± 123.88
	RTT (Min)	5.18 ± 7.73	8.04 ± 14.43	10.16 ± 19.66
	Packet Loss (%)	2.47 ± 1.35	3.19 ± 1.90	3.61 ± 2.18

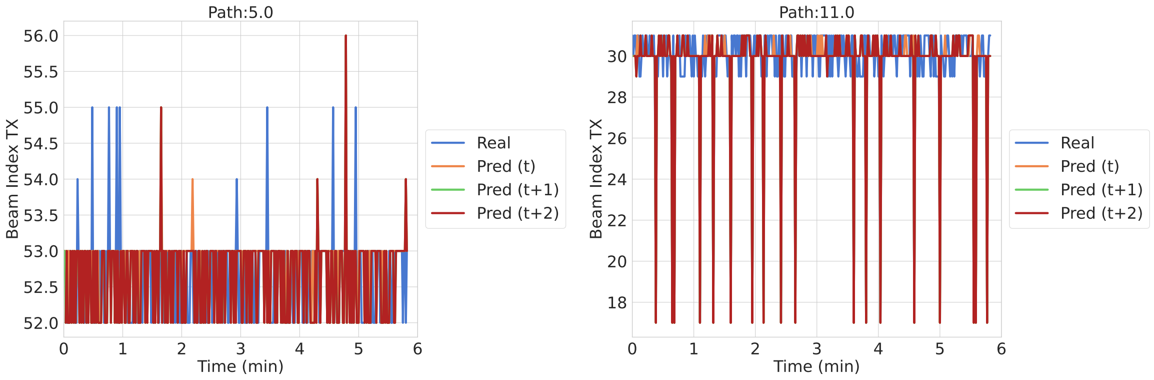
Table C.20. Examples of the results obtained with some of the settings tested and for the best-performing model (green) during hyperparameter tuning for the LSTM-based RNN.

Hyperparameters									Results			
No. hidden Layers	First layer neurons	Other hidden layers neurons	Dropout Rate	Kernel Initializer	Batch Size	Solver	Learn Rate	Round Epochs	Loss (MSE)		MAE	
									Train	Test	Train	Test
4	100	128	0.40	uniform	64	rmsprop	0.1000	5	0.2421	0.2417	0.3344	0.3341
3	64	200	0.05	uniform	512	rmsprop	0.1000	9	0.1492	0.1497	0.2415	0.2420
3	100	64	0.60	normal	32	sgd	0.0001	100	0.1442	0.1441	0.3505	0.3504
1	16	100	0.50	normal	128	nadam	0.1000	100	0.0062	0.0249	0.0420	0.0915
3	32	256	0.70	uniform	8	sgd	0.0003	100	0.0781	0.0779	0.2537	0.2535
2	64	64	0.05	uniform	8	sgd	0.0030	100	0.0585	0.0585	0.1835	0.1836
3	256	64	0.05	uniform	512	adam	0.1000	21	0.0297	0.0285	0.1079	0.1074
3	128	32	0.60	uniform	2	sgd	0.0300	100	0.0279	0.0255	0.1124	0.1070
3	64	8	0.70	normal	512	nadam	0.0100	49	0.0302	0.0253	0.1128	0.1049
5	256	64	0.05	uniform	128	adam	0.0050	43	0.0024	0.0021	0.0220	0.0203
5	32	100	0.20	uniform	64	rmsprop	0.0010	100	0.0027	0.0021	0.0234	0.0202
4	16	200	0.10	normal	64	rmsprop	0.0010	100	0.0021	0.0019	0.0192	0.0175
4	16	200	0.10	normal	64	rmsprop	0.0010	100	0.0021	0.0019	0.0192	0.0175
3	100	128	0.30	normal	32	nadam	0.0003	83	0.0020	0.0018	0.0200	0.0173
3	100	128	0.30	normal	32	nadam	0.0005	83	0.0020	0.0018	0.0200	0.0173
3	128	100	0.20	normal	256	nadam	0.0010	100	0.0018	0.0017	0.0183	0.0162
3	256	128	0.20	uniform	128	nadam	0.0005	100	0.0017	0.0017	0.0170	0.0154



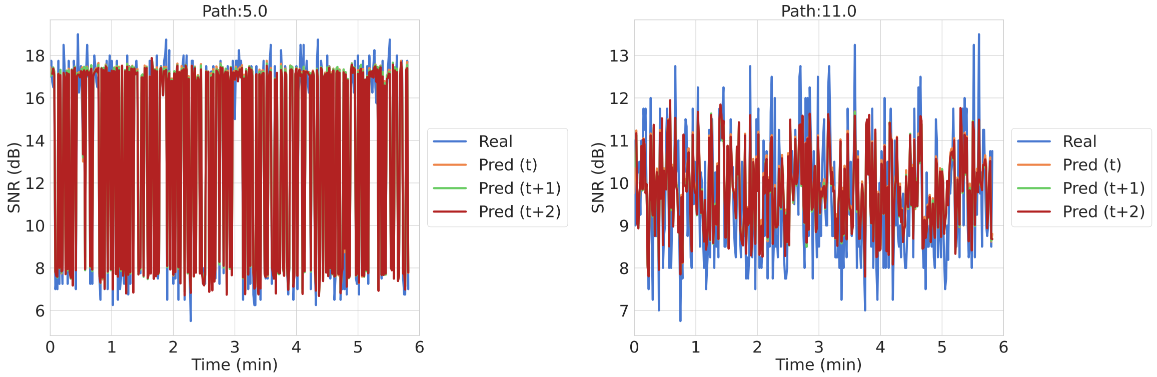
(a) (b)

Figure C.2. Predicted RX beams for two mmWave paths.



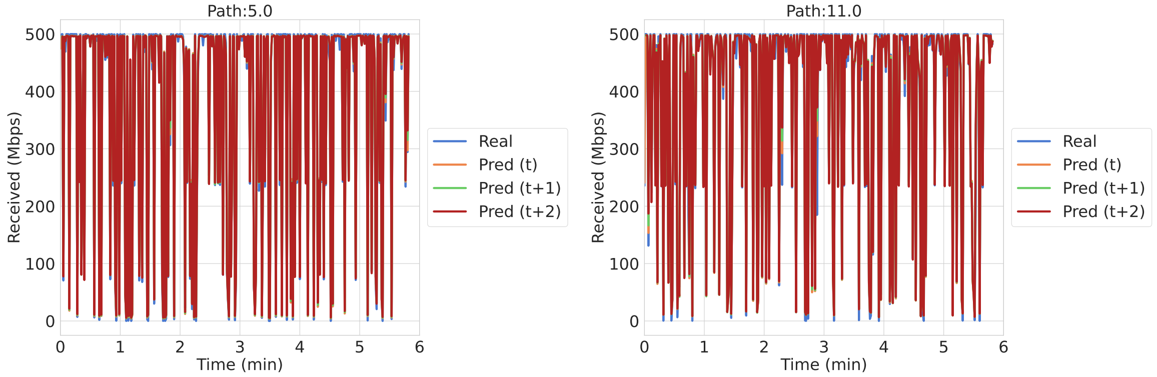
(a) (b)

Figure C.3. Predicted TX beams for two mmWave paths.



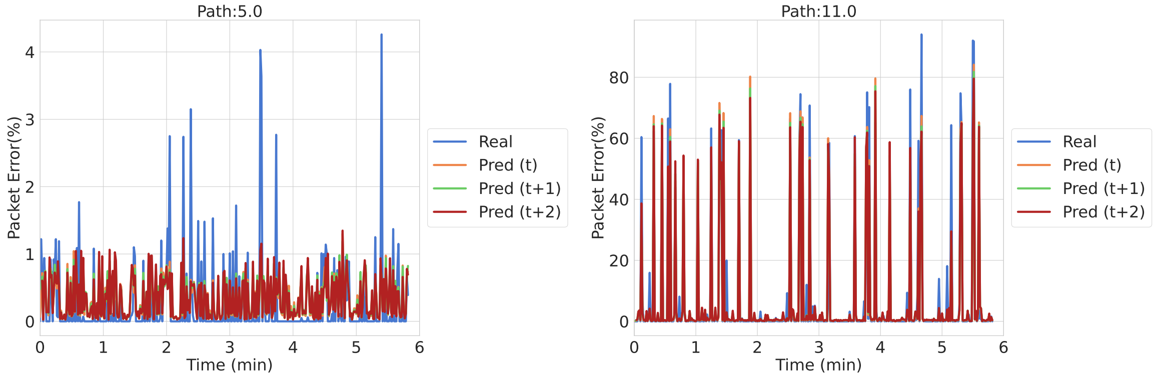
(a) (b)

Figure C.4. Predicted SNR for two mmWave paths.



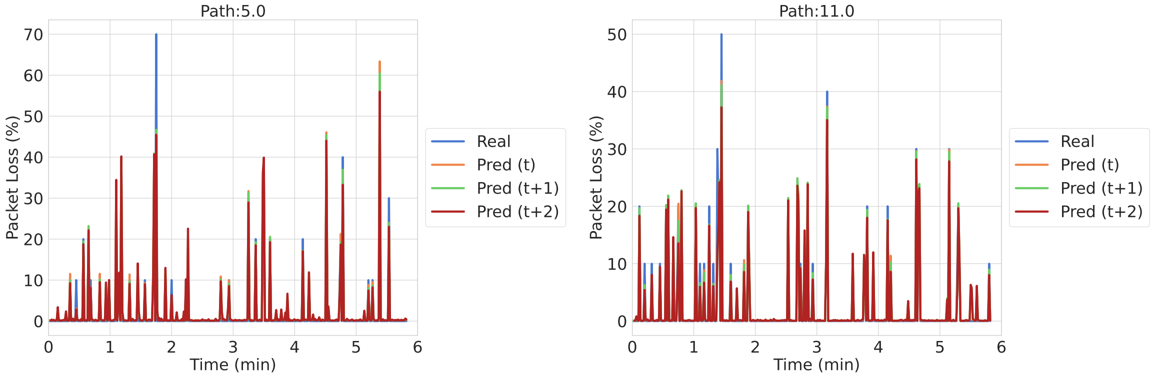
(a) (b)

Figure C.5. Predicted received data rate for two mmWave paths.



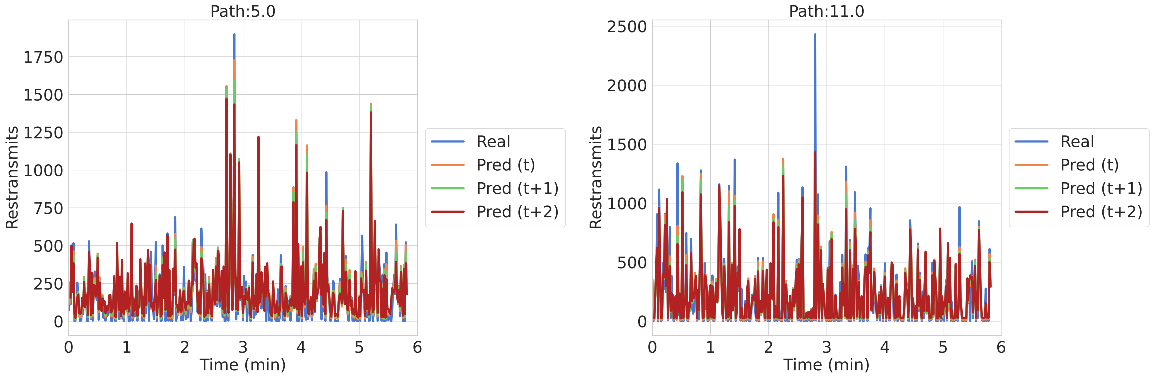
(a) (b)

Figure C.6. Predicted PER for two mmWave paths.



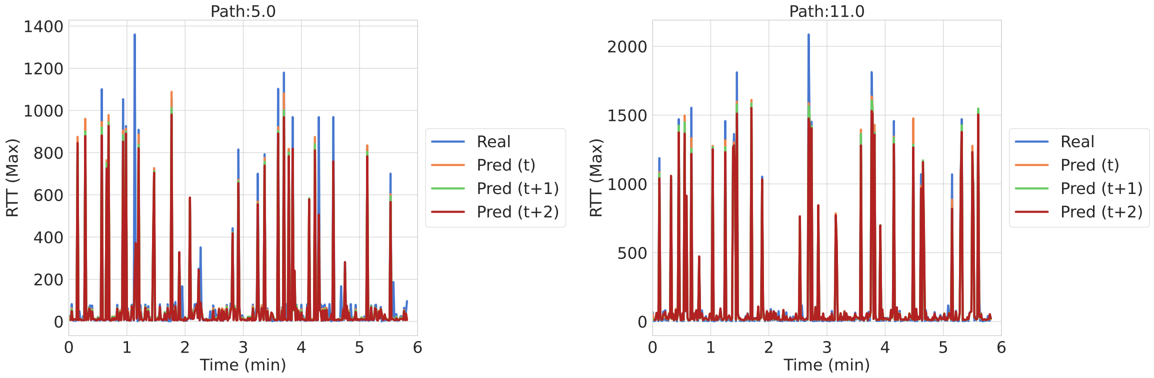
(a) (b)

Figure C.7. Predicted packet loss for two mmWave paths.



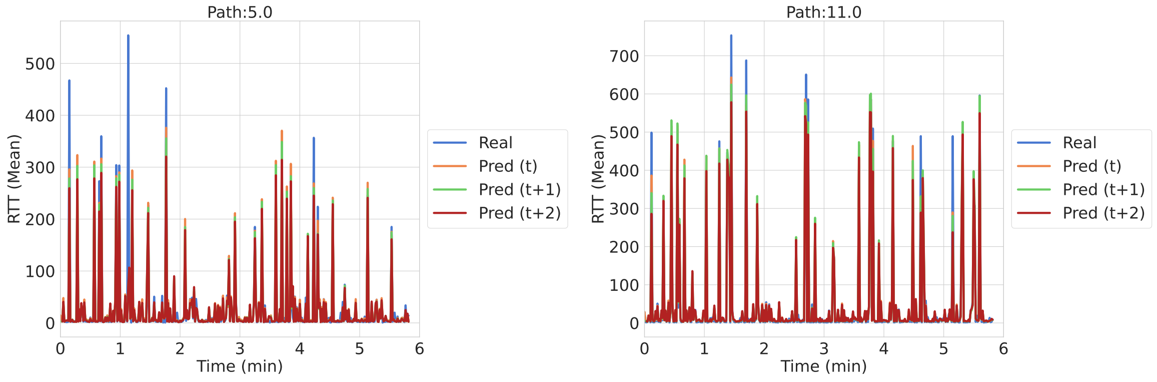
(a) (b)

Figure C.8. Predicted number of TCP retransmits for two mmWave paths.



(a) (b)

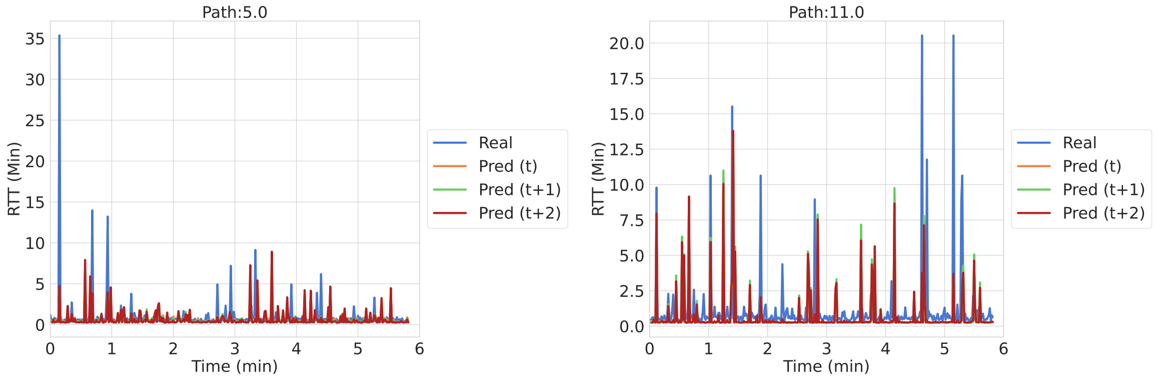
Figure C.9. Predicted maximum RTT for two mmWave paths.



(a)

(b)

Figure C.10. Predicted mean RTT for two mmWave paths.



(a)

(b)

Figure C.11. Predicted minimum RTT for two mmWave paths.