Bandwidth and Power Saving Optimisation of Fronthaul for Beyond-5G Mobile Networks

Mohsan Niaz Chughtai

School of Engineering
University of Kent

A Thesis Submitted for the Degree of Doctor of Philosophy
in Electronic Engineering

May, 2024

Acknowledgements

I extend my heartfelt gratitude to my PhD supervisors, Prof. Nathan J. Gomes and Dr. Philippos Assimakopoulos, for their unwavering technical and moral support throughout my doctoral journey. Their guidance has been instrumental in not only completing my PhD studies but also in honing my critical research and writing skills.

In addition, I am deeply thankful to Prof. Jiangzhou Wang and Dr. Huiling Zhu for their invaluable guidance throughout the OpTIMO project.

I would like to acknowledge the guidance and support from my colleague, Dr. Shabnam Noor, during my initial phase of my PhD studies and throughout the OpTIMO project. Her collaboration and insights have been immensely beneficial.

Abstract

In the fronthaul segment of beyond-5G mobile networks, the bit rate variability leads to varying end-to-end latency, affecting user throughputs managed by coordinated multipoint beamforming (CoMP), which in turn adds to variability in fronthaul bit rates. To address this, a resource allocation scheme is needed to improve both bandwidth utilisation and power efficiency in fibre optic link components. A model is also required to predict percentile delays within the fronthaul segment to evaluate the impact of variable end-to-end latency on user throughputs.

The fronthaul segment in beyond-5G mobile networks is envisioned to be equipped with FlexE. For efficient bandwidth utilisation in the fronthaul, traffic prediction for slot allocation in FlexE calendars is required. If CoMP is used, the allocation of users to remote units (RUs) with a fronthaul path of lower latency to central/distributed unit (CU/DU) will increase the achievable user bit rate. In the thesis research, the use of multi-agent deep reinforcement (DRL) learning for optimal slot allocations in a FlexE-enabled fronthaul, for traffic generated through CoMP, and for offloading users among different RUs is explored. It is shown via simulations, that the DRL agent can learn to predict input traffic patterns and allocate slots with the granularity of 5 Gbps in the FlexE calendar, resulting in bandwidth enhancement of up to 11.6 % in comparison to ARIMA-based predictions. A separate DRL agent can also offload UEs from paths of higher latency to lower latency paths, resulting in user throughput enhancement by up to 7.3%.

A model for end-to-end latency for a fronthaul network is also developed in the thesis research and used as an input parameter in a pre-developed model for CoMP-based beamforming in the thesis research. The model is adaptable to different network configurations, traffic parameters and can be extended to predict percentile delays higher than 99.9%. A comparison with simulation results indicates that the model tends to over-predict 99.9 percentile delays by varying degrees (8% to around 150%) as network load fluctuates from 0.1 to 0.9, thus serving as an upper bound on percentile delays, without any under-predictions.

The increasing bit rate demands placed on the fronthaul from high and variable user rates will make the consideration of its power consumption an important issue. DRL-based traffic predictions and slot allocations in a FlexE-enabled fronthaul link can reduce the bit rate, leading to lower transmit signal powers and reduced power consumption in the transmitter of a low-cost, short DMT-type fibre optic fronthaul link. Using DRL predictions and simulations of a DMT modulation EAM-based optical fibre link, it is shown that decreasing bit rates and modulation levels, the transmit signal power can be reduced ranging from 22.3% to 34.6% within a fixed bandwidth of 34 GHz and 18 GHz. Such a transmitter could function as a bandwidth variable transponder in a Flexible Ethernet fronthaul.

List of publications

1. S. Noor, I. Laurinavicius, M.N. Chughtai, P. Assimakopoulos, H. Zhu, N. J. Gomes and J. Wang, ‘Optimisation of transport infrastructure resources with MU-MIMO beam allocation requirements’, CEMS, University of KENT, Huawei-Kent collaboration [OPTIMO], 2021. [Confidential report].

2. M. N. Chughtai, S. Noor, I. Laurinavicius, P. Assimakopoulos, H. Zhu, N. J. Gomes and J. Wang, ‘User and resource allocation in latency constrained Xhaul via reinforcement learning’, Journal of Optical Communications and Networking, vol. 15, no. 4, pp. 219–228, Apr. 2023, doi: 10.1364/JOCN.485029.

3. M. N. Chughtai, P. Assimakopoulos, and N. J. Gomes, ‘Reinforcement Learning-Assisted Transmit Signal Power Savings in Variable Bit-rate Fronthaul’, IEEE Communications Letters, pp. 1–1, 2024, doi: 10.1109/LCOMM.2024.3386848.

4. M. N. Chughtai, P. Assimakopoulos, and N. J. Gomes, ‘A Model for end-to-end percentile delay in 5G and beyond Xhaul packet networks’, [Under preparation for submission]

Table of contents

List of figures

List of tables

List of Abbreviations

Acronyms / Abbreviations

3GPP	3rd generation partnership project
A/D	Analogue to digital
A2C/A3C	Actor-critic methods
AGC	Automatic gain control
AGI	Artificial general intelligence
ARIMA	Auto-regressive integrated moving average
ARQ	Automatic repeat request
ASE	Amplified spontaneous emission noise
BBU	Baseband unit
BVT	Bandwidth variable transponder
C-RAN	Cloud radio network access
CO	Central office
CoMP	Coordinated multi point beamforming
COTS	Commercial Off-The-Shelf
CPRI	Common protocol radio interface
CSI	Channel state information
CUA	Cumulative probability of under-allocation

D-RAN	Distributed radio access network
D/A	Digital to analogue
DC	Direct current
DDPG	Deep deterministic policy gradient
DFB	Distributed feedback laser
DML	Directly modulated laser
DMT	Discrete multitone transmission
DNN	Deep neural networks
DRL	Deep reinforcement learning
DSP	Digital signal processing
DU/CU	Distributed/central unit
EAM	Electro absorption modulator
EDFA	Erbium-doped power amplifier
eMBB	Enhanced mobile broadband
EOM	Electro-optical modulator
EVM	Error vector magnitude
FEC	Forward error correction
FFT	Fast fourier transform
FIFO	First-in-first-out
FlexE	Flexible ethernet
gNB	gnodeB
GPD	Generalised pareto distribution
H-CRAN	Heterogeneous cloud radio network architecture
HARQ	Hybrid Automatic Repeat Request

HLS	High layer split
IDFT	Inverse discrete fourier transform
IEEE	Institute of electrical and electronics Engineers
IFFT	Inverse fast fourier transform
IF	Intremediate frequency
ILP	Integer linear programming
IoT	Internet of Things
ITU-T	International telecommunication union standardisation sector
LLS	Low layer split
LO	Local oscillator
MAC	Media access control
MATLAB	Matrix laboratory
MAWI	Measurement and analysis on the WIDE internet
MDP	Markov decision process
MILP	Mixed integer linear programming
MIMO	Multiple-input multiple-output
mMTC	Massive machine type communication
MOA	Mean over-allocations
MZM	Mach-Zhender modulator
NF	Noise figure
NGFI	Next-generation fronthaul interface
NIC	Network interface controller
NR	New radio
ODU	Optical distribution units

OFDM	Orthogonal frequency division
OTN	Optical transport networks
PCA	Principal component analysis
PCN	Packet core network
PCS	Physical coding sublayer
PMD	Physical medium dependent
PDCP	Packet data convergence protocol
PDF	Probability density function
PDU	Protocol data unit
PHY	Physical layer
PCS	Physical medium attachment
PM	Phase modulator
PNC	Physical network controller
PPO	Proximal policy optimisation methods
PRB	Physical resource block
PSD	Power spectral density
QAM	Quadrature amplitude modulation
QoS	Quality of service
RAN	Radio Access Network
RF	Radio frequency
RIN	Relative intensity noise
RLC	Radio link control
RL	Reinforcement learning
RMS	Root mean square

RRC	Radio Resource Control
RU	Remote unit
SDAP	Service Data Adaptation Protocol
SDR	Software defined radio
SDU	Service data unit
SNR	Signal to noise ratio
SSMF	Standard single mode fibre
SVM	Support vector machines
UE	User equipment
uRLLC	Ultra-reliable and low-latency communication
VPI	Virtual Phonotics Incorporated
WIDE	Widely integrated distributed environment
Xhaul	Cross-haul
ZF	Zero forcing

Chapter 1

Introduction

1.1 Motivation and Aims of the Thesis

Functional splits in 5G and beyond mobile networks are proposed for the division of network functions of the physical, data link and network layer between the centralised unit (CU), the distributed unit (DU) and the remote unit (RU). These splits are crucial for optimising network performance and efficiency. They allow for tasks such as radio resource management and baseband processing to be distributed closer to the edge of the network, enabling lower latency and improved capacity [1]. In total eight splits ($1\sim8$) have been proposed by the 3$^{\text{rd}}$ generation partnership project (3GPP) [2]. The details and advantages of different functional splits will be discussed in Chapter 2. The segment between the RU and CU is termed as 'Fronthaul', and the segment between the CU and DU is termed as 'Midhaul'. These segments are jointly termed as 'Xhaul' [3].

In 5G networks, an RU serves as a key component responsible for wireless signal transmission and reception at the network edge. It is typically deployed at cell sites or base station locations and plays a crucial role in providing coverage and capacity to users. RUs are equipped with antennas and radio frequency (RF) equipment to communicate with user devices and other network elements [4]. Functionally, RUs are responsible for several tasks, including radio signal processing, modulation/demodulation, encoding/decoding, and beamforming. Additionally, RUs may perform protocol termination, packet processing, and scheduling functions, depending on the specific functional split deployed [1].

Beamforming at RUs of current and beyond-5G mobile networks will enhance wireless communication efficiency by directing radio frequency (RF) energy toward specific user locations rather than broadcasting uniformly. This targeted approach improves signal quality, capacity, and coverage, and reduces interference [5]. Centralised Coordinated Beamforming (CoMP) further optimises beamforming by coordinating multiple RUs under a central controller such as CU. RUs estimate Channel State Information (CSI) which comprises encompassing factors such as signal strength, interference levels, and propagation characteristics, and share it with the CU, which then optimises beamforming weights for each RU and user [6], [7]. This coordinated approach ensures efficient resource utilisation, enhancing network performance and user experience. It's worth noting that while CoMP is discussed, the thesis research utilises a pre-developed to model user throughput in beyond-5G mobile networks.

To achieve high bit rates of up to 10 Gbps per user, proposed technologies in current 5G and beyond-5G mobile networks include coordinated beamforming, massive multiple-input multiple-output (MIMO), and cell-free MIMO [8], [9]. These technologies heavily depend on the existence of a fronthaul network, connecting remote unit (RU) antennas to the packet core network (PCN) [10]. Transportation of data between the ends of RUs and PCN is constrained by factors such as available resources, end-to-end latency, and power consumption of fibre optic links in the fronthaul network [11], [12].

One of the functional splits option 7.2, shifts the functions of the Fast Fourier transform (FFT) and resource element mapper into the DU. For the functional split below 7.2, the fronthaul bit rate scales with user bit rates, resulting in overall bit rate variations in fronthaul [1]. User bit rates depend on factors of beamforming, user mobility, allocation to RUs, and resource allocation in RAN and fronthaul [13], [14], [15], [16], [17]. The variation in fronthaul bit rates results in a corresponding variation in the end-to-end latency of fronthaul paths. If users are managed by coordinated multipoint beamforming (CoMP), throughputs are impacted by the variable latency of the fronthaul path [18]. This variable latency affects the delay in feeding back channel state information (CSI) via uplink to the (central/distributed unit) CU/DU where beamforming vectors are computed, adding more variation in the overall bit rate in the fronthaul [18], [19]. In summary, fronthaul bit rate variation leads to end-to-end latency variation in fronthaul paths, causing user throughput variation, further contributing to overall fronthaul bit rate variation. The blue circles connected by blue arrows in Fig. 1.1 illustrate the cycle of the interdependence of fronthaul bit rate variation, end-to-end latency, and user throughput variation

Given the impact of end-to-end latency on fronthaul bit rates, one of the aspects of research in the thesis focuses on comprehensive end-to-end latency modelling shown by the green box on the lower left-hand side in Fig. 1.1. Similar to packet networks, aggregating various fronthaul flows results in packet queuing within RUs and aggregation nodes. Packets exceeding a certain end-to-end latency threshold will be dropped, necessitating modelling end-to-end latency in terms of percentile delay [3]. A unique aspect of the thesis research is modelling percentile delays for a packet network with multiple levels of aggregation, random

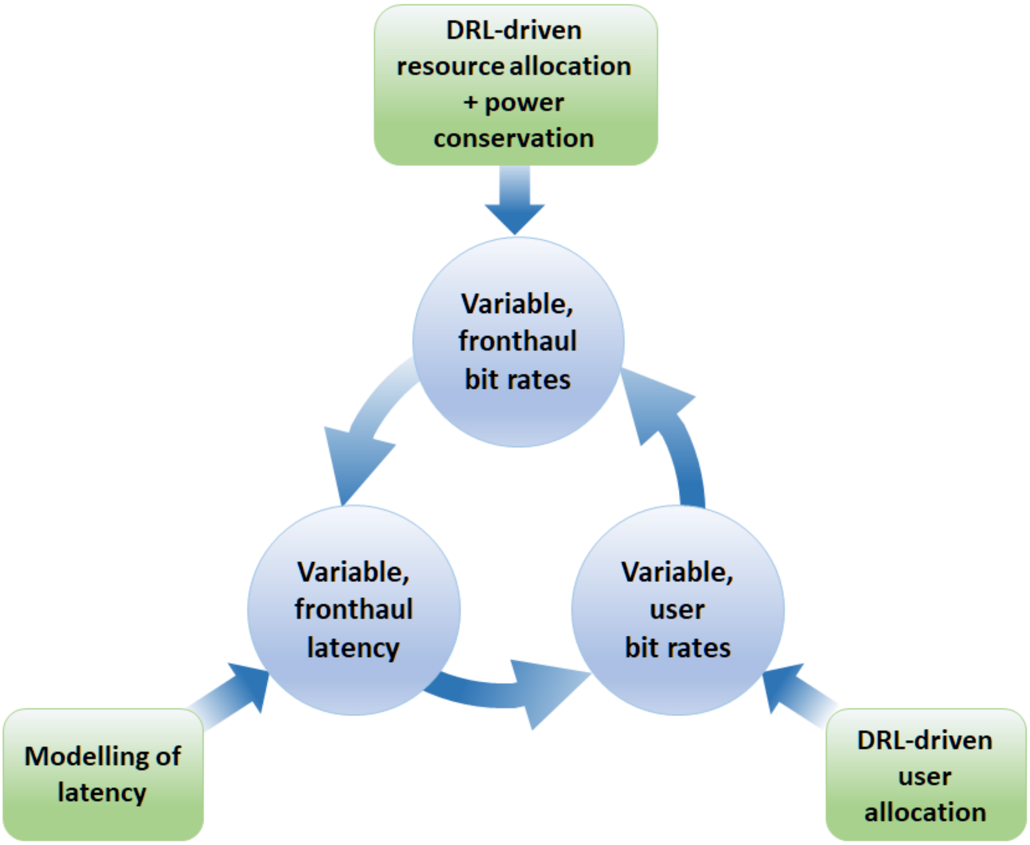

DRL-driven resource allocation + power conservation
Variable, fronthaul bit rates
Variable, fronthaul latency
Variable, user bit rates
Modelling of latency
DRL-driven user allocation

Fig. 1.1: Fronthaul bit rates, latency and user bit rates cycle.

inter-arrival times of packets, and random packet lengths. The modelling of end-to-end latency is carried out using different queuing models based on G/G/1 and Lindley's recursion. The delays are modelled in terms of 99-percentile delays for a generalised packet network similar to a fronthaul network.

The variation in bit rates in fronthaul will initiate the requirement of dynamic resource allocation in the fronthaul links which will lead to a reduction in bit rates from the maximum bit rate (e.g., 100 Gbps). The reduction will benefit in terms of efficient bandwidth utilisation in the physical layer (PHY) of fronthaul links. Studies in [13] and [20] have examined varying bit-rates between standard optical transport networks (OTN) or Ethernet PHY bit rates (e.g., at multiples of 100 Gbps), but a finer granularity of bit-rate variation for the fronthaul is considered in the thesis research. A finer granularity can respond to smaller changes in input traffic bit-rates resulting in a greater reduction in bit-rate, and the potential for a greater bandwidth efficiency in fronthaul links. A 5 Gbps granularity was chosen because it is supported by the Flexible Ethernet (FlexE) calendar mechanism (the FlexE shim) [23]. At the physical layer (PHY), the transmission of 5 Gbps slots can be facilitated by employing bandwidth variable transponders (BVTs). These BVTs can adjust overall bandwidth and quadrature amplitude modulation (QAM) levels to accommodate different bit rates, as elaborated in [24].

In the studies in [23], the PHY layer bit rates are reduced from the fixed maximum bit rate, in increments of 5 Gbps, following the input traffic bit rates, predicted by using auto-regressive integrated moving average (ARIMA). The prediction of traffic patterns through ARIMA requires fine-tuning of its parameters such as the length of filter taps and their coefficients, which depend on the statistical properties of the input traffic pattern. If there is a variation in the statistical traffic patterns, then it will require re-tuning of the parameters for ARIMA. On the other hand, deep reinforcement learning (DRL) can also be used to predict traffic patterns and subsequent slot allocations. The advantage of predicting traffic patterns via a DRL-based agent is that it can be trained on different traffic patterns with different statistical properties of the mean and standard deviations. After training, the DRL agent will not require re-training for different input traffic patterns. Therefore the thesis research utilises DRL for traffic prediction and resource allocation in a simulated environment, where mobile user traffic is generated using CoMP using delays from the model for end-to-end latency in the first aspect of the research. This aspect of the research in the thesis is indicated by the green box on the top in Fig. 1.1.

In CoMP, UEs provide feedback about the channel conditions to the serving RUs. This feedback includes information about channel gains, interference levels, and other relevant parameters. If this feedback is delayed due to network latency or processing time, the CUs may make suboptimal decisions based on outdated information. These errors can lead to suboptimal beamforming decisions, affecting user throughput as delayed CSI may not accurately reflect the current path loss. In a fronthaul network where users are managed by CoMP, the end-to-end latency can degrade the user throughputs [25], [26]. Considering the degradation of user throughputs, a second DRL agent was also trained for offloading users from a path of higher latency to a path of lower latency for the enhancement of the overall throughput of the user data rates, thus converting the system into a multiagent DRL system. This aspect of the research is shown in the lower green box on the right-hand side in Fig. 1.1.

In the physical layer of fronthaul links, electro-absorption modulators (EAMs) and Mach-Zehnder modulators (MZM) may be used to modulate the optical carrier for data transmission, requiring drive signal amplification using a drive amplifier. The power consumption from the modulator and the drive amplifier in the transmitter constitutes a significant portion of the overall transmitter power in a fibre optic transmission link [27],[28]. Based on the power consumption models for an EAM in [28] and [29], the power consumption of the modulator and the drive amplifiers can be minimized, by reducing drive signal powers, which in turn can be reduced via a reduction in fronthaul bit-rates. This is so as reduced bit rates will require lower QAM levels to maintain receiver signal-to-noise ratios (SNRs) required for the standard EVMs in the 3GPP standard for 5G mobile networks [30].

In FlexE-enabled fronthaul links, the transmission of varying numbers of 5 Gbps aided by DRL-based predictions can accommodate the varying fronthaul traffic from a fixed maximum bit rate (e.g., 100 Gbps). The reduction in fronthaul bit rates can then, lead to a reduction in power consumption of the modulator and drive amplifier in the transmitter, thus contributing to the overall power consumption of the link. The thesis research, therefore, aims to reduce transmit signal power indicated by the green box at the top in Fig 1.1.

Considering the discussion above for DRL-assisted slot/resource in FlexE-enabled fronthaul and user offloading, the overall vision of Xhaul for beyond-5G mobile networks as the basis for research for the thesis is given in Fig. 1.2 [31]. The fronthaul segment data from different RUs is aggregated by aggregation nodes enabled by FlexE. The data from different FlexE aggregation nodes is transported through an Xhual connected through discrete multitone modulation (DMT)–based fibre optic transmission links. The DU and CU are shown separately and connected to different FlexE-enabled aggregations. In the lower part of Fig. 1.2, the internal structure of the FlexE aggregation node is shown where the input traffic from fronthaul is measured in terms of bit rate and then a trained DRL agent carries out the prediction of the bit rates for slot allocations in the FlexE shim. At the output of the FlexE shim, a BVT transports the FlexE slots over different sub-carrier groups of the DMT multiplex bandwidth as shown in the spectrum in the upper right corner of Fig. 1.2. It was also assumed that the DRL agent responsible for handling the user among different RUs via CoMP is located in the CU in Fig. 1.2.

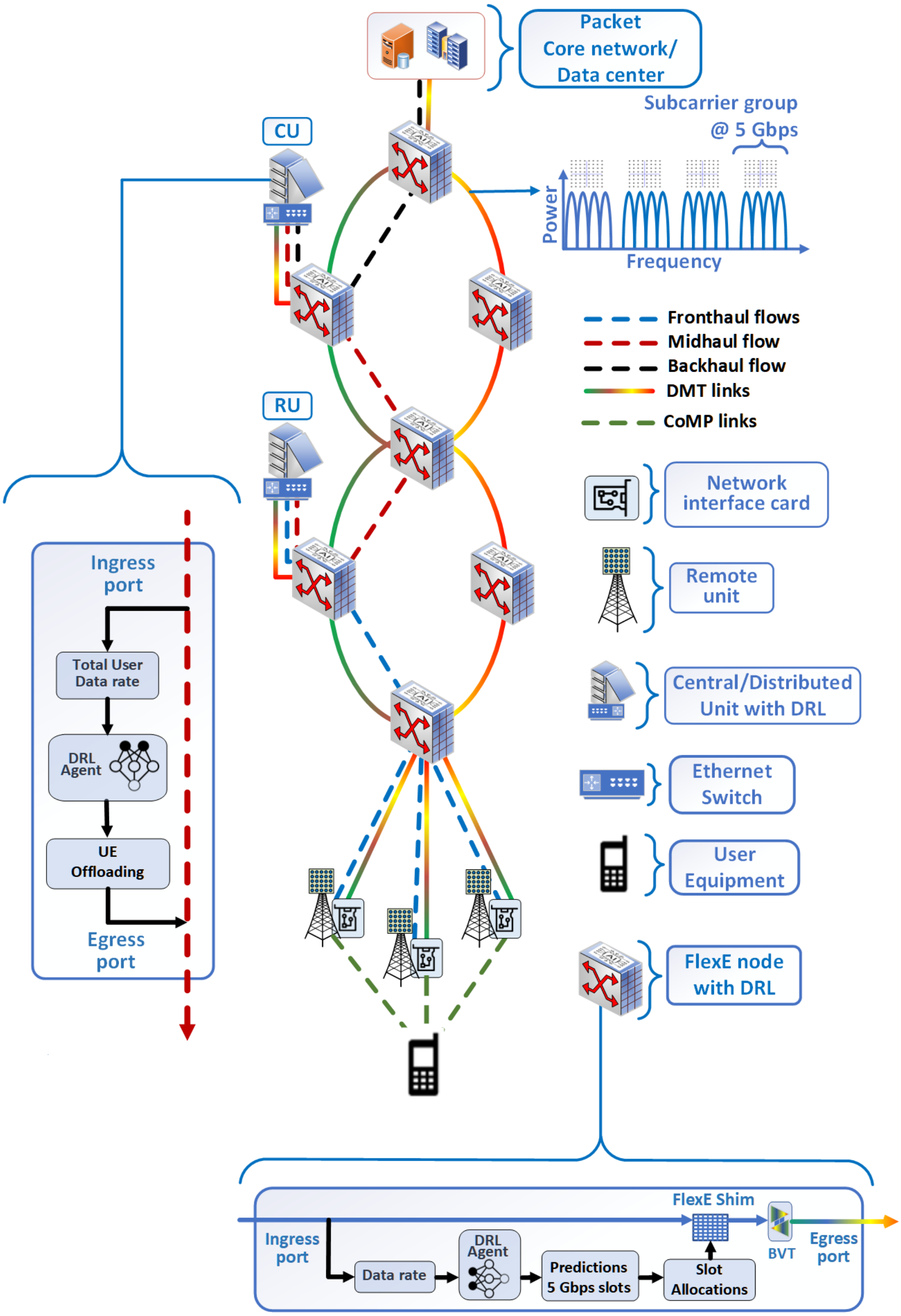


Fig. 1.2: Xhaul vision for 5G and beyond mobile networks.

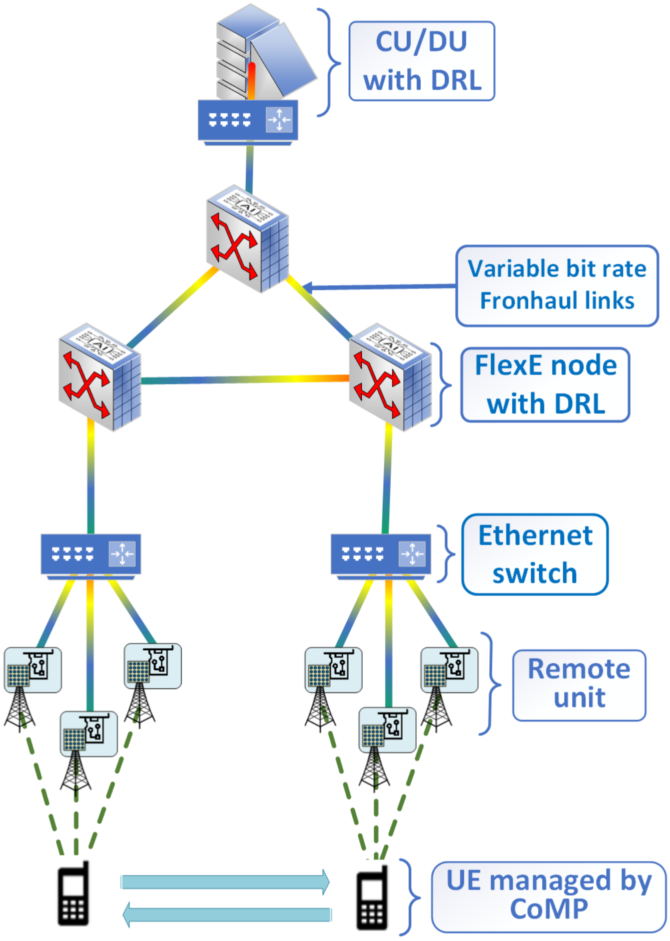


Fig. 1.3: Fronthaul network for 5G and beyond mobile networks.

To accommodate the research in resource allocation in fronthaul and user offloading, within reasonable limits of computation time and resources, a scaled-down architecture similar to the architecture in Fig. 1.2, was simulated and shown in Fig 1.3. The simplification was done by considering fewer numbers and levels of aggregation nodes than in Fig. 1.2. The analysis assumed a 5G network with only fronthaul links, consisting of two RU clusters and the CU/DU, similar to fronthaul for 5G network architectures considered for research in [32], [33], [34]. For the network in Fig 1.3. it was also assumed that the uplink traffic, from RUs was aggregated by Ethernet-based aggregation nodes and then further aggregated by FlexE-based aggregation nodes up to the CU/DU. Similar to Fig. 1.2, the FlexE aggregation nodes are equipped with DRL for traffic prediction and slot allocations for enhancement of bandwidth efficiency in the fronthaul network. The users in Fig 1.3 are managed by the CoMP in the CU/DU. The CU/DU is also equipped with DRL for user offloading from a path of higher latency to a path of lower latency for the enhancement of user bit rates. The results based on the simplified network in Fig. 1.3 are presented in Chapter 4.

1.2 Summary of Main Contribution of the Thesis

There are three aspects of research in the thesis given as follows,

1. Development of an estimation model for end-to-end percentile delay in fronthaul networks.

   In this part of the thesis research, the main contribution is in the form of an analytical model for end-to-end latency in terms of 99 and 99.9 percentile delays in fronthaul network paths. Although the results were developed for a packet network similar to a fronthaul network the model can also be applied to any generalised packet network.

   For the development of the model, the statistical distributions, the mean, and the standard deviation of inter-arrival times of packets at different levels of aggregation were estimated. Using these properties of inter-arrival times of packets, the mean waiting times of packets at different levels of aggregation were modelled using Lindley's recursion and G/G/1 queuing model. To model the end-to-end percentile delays the statistical distribution of the end-to-end delays was also modelled for different loads on aggregation nodes. The output of the model was in the form of upper bounds on 99 and 99.9 percentile delays. The performance of the model was also benchmarked with percentile delays extracted from packet traces from simulation in NetSim®. The predicted percentile delays were used as one of the input parameters in the model for beamforming based on CoMP in the first aspect of the research.

   The model for percentile delays developed in this part of the research was used as input for the model of RAN in Chapter 4. The model has also been discussed in the report for the OpTIMO project [Publication Num. 1]. Based on the simulation results in this part of the research, a journal is in progress to be submitted for publication [Publication Num. 4].

2. Reinforcement learning assisted user and resource allocation in latency-constrained fronthaul.

   In this part of the thesis research, a DRL system was implemented using Simulink MATLAB®, to allocate resources and manage users within a pre-existing model of a Radio Access Network (RAN). This model incorporated Coordinated Multipoint (CoMP) and FlexE-enabled aggregation nodes within the fronthaul framework. To demonstrate the impact of latency in fronthaul, a model based on the G/G/1 queuing model was also incorporated into the model. Three DRL agents were then concurrently trained, two for slot allocation in FlexE shim based on the observation of input traffic from RAN and background sources in the network and one agent was trained to offload user equipment (UEs) among two RUs connected to the CU/DU via the fronthaul model.

The main contribution from this part of research is the bandwidth efficiency enhancement in fronthual and user throughput, through the use of DRL in a simulated combined RAN/Fronthaul model. In the contribution, it is shown using simulation results that DRL agents can track fronthaul traffic and allocate resources as well as offload users from a path of higher latency to lower latency for throughput enhancement.

In the contribution, the impact of different hyper-parameters on the training of the DRL agents was analysed via simulations. The impact of latency on the overall throughput of the system and the enhancement of throughput via user offloading controlled by a DRL agent was also shown in the results as a proof of concept.

The results obtained from this part of the research have been published in [Publication Num. 2].

3. Physical layer transport of FlexE using the discrete multitone transmission in short, low-cost fibre optic links.

   In this part of the thesis research, the transport of FlexE slots over a short-link low-cost fibre optic transmission link was demonstrated via simulations in VPItransmission Maker v11.2®. The transmission of FlexE slots was simulated using different modulation formats. The contribution and novelty were in terms of reducing the average transmit DMT signal power required to transmit varying numbers of 5 Gbps slots for the traffic predicted by the DRL agent within fixed bandwidth limits. The motivation for the reduction of the transmit DMT single-power originates from the fact that the power consumption of a modulator unit comprising of an MZM, EAM and the drive amplifier depends on the power of the drive signal which in the current research is the transmit DMT signal power. [27]. The main objective was to demonstrate relative power savings in trasmit DMT signal power rather than absolute performance when compared to fixed bit rate transmission.

   The second novelty in this part of the thesis research is the development of preamble averaging-based equalisation for improved performance in terms of percentage EVM of the DMT transmission link under test in comparison to zero-forcing (ZF) equalisation. Preamble averaging-based equalisation eliminated the need for interpolation operation required in ZF equalisation and the dependence of residual post-ZF equalisation noise on the symbol amplitudes of the QAM constellation.

The results for the transmit DMT signal power savings aided by DRL-based traffic prediction have been published in the [Publication Num: 3].

1.3 Structure of the Thesis

1. In the Chapter 1 of the thesis, the motivation and novelty of the research thesis are discussed.

2. In the Chapter 2 of the thesis, the background theory of the research aspects is discussed and given as follows;

   - An introduction to the basic 5G network architecture, 5g network services, 5G network fronthaul architecture, and the advantages/disadvantages of the different functional splits are discussed.
   - An introduction to the Flexible Ethernet standard, its operational modes, and frame structure is given.
   - An introduction to the concepts of the G/G/1 queuing model and Lindley's recursion is given in detail. The model for prediction of latency in terms of percentile delays using the G/G/1 queuing model and Lindley's recursion is also discussed in detail.
   - An introduction to the concepts of reinforcement learning and deep-Q learning is given.
   - An introduction to the concepts of discrete multitone modulation, digital and analogue fronthaul architectures, Nyquist zone demapping and preamble averaging-based equalisation are given in detail.

3. In Chapter 3 the results related to the modelling of end-to-end delays via simulations are given. A comparison of simulation results with the theoretical models presented in Chapter 2 for different network configurations is also given in detail.

4. In Chapter 4, the simulation results related to user and resource allocation using DRL are discussed in detail. The impact of different hyper-parameters on the training of the DRL agents is also discussed. The impact of latency in fronthaul on the overall user throughput in RAN is shown. It is also shown that the user offloading among different RUs via the DRL agent can enhance the overall user throughput.

5. In Chapter 5, the simulation results for the transmission of 5 Gbps slots via DMT transmission over a short fibre optic communication link are given in detail. The transmission results in terms of transmit DMT single power for different bit rates, advanced modulation formats and fixed bandwidth limits are given in detail. The achievable spectral efficiency of the link is also modelled and compared with the Shannon's capacity.

   The performance of the link in terms of EVM using pilot-based equalisation and preamble-based equalisation is also discussed. The results for the reduction in the transmit DMT signal power within fixed bandwidth limits are also given in detail.

6. In Chapter 6 the conclusions of research and future directions for research are discussed.

References

[1] L. M. P. Larsen, A. Checko, and H. L. Christiansen, ‘A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks’, IEEE Communications Surveys Tutorials, vol. 21, no. 1, pp. 146–172, First quarter 2019, doi: 10.1109/COMST.2018.2868805.

[2] ‘Specification 38.801’. Accessed: Mar. 29, 2024. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3056

[3] P. Iovanna, F. Cavaliere, S. Stracca, L. Giorgi, and F. Ubaldi, ‘5G Xhaul and Service Convergence: Transmission, Switching and Automation Enabling Technologies’, Journal of Lightwave Technology, vol. 38, no. 10, pp. 2799–2806, May 2020, doi: 10.1109/JLT.2020.2983469.

[4] L. Peterson, O. Sunay, and B. Davie, ‘Radio Access Network’, in Private 5G: A Systems Approach, Systems Approach LLC, 2023. [Online]. Available: https://5g.systemsapproach.org/index.html

[5] I. A. Haroun, ‘Introduction to MIMO and Beamforming Technology’, in Essentials of RF Front-end Design and Testing: A Practical Guide for Wireless Systems, IEEE, 2024, pp. 207–222. doi: 10.1002/9781394210640.ch8.

[6] F. Qamar, K. B. Dimyati, M. N. Hindia, K. A. B. Noordin, and A. M. Al-Samman, ‘A comprehensive review on coordinated multi-point operation for LTE-A’, Computer Networks, vol. 123, pp. 19–37, Aug. 2017, doi: 10.1016/j.comnet.2017.05.003.

[7] V. Jungnickel et al., ‘The role of small cells, coordinated multipoint, and massive MIMO in 5G’, IEEE Communications Magazine, vol. 52, no. 5, pp. 44–51, May 2014, doi: 10.1109/MCOM.2014.6815892.

[8] S. Chen, Y.-C. Liang, S. Sun, S. Kang, W. Cheng, and M. Peng, ‘Vision, Requirements, and Technology Trend of 6G: How to Tackle the Challenges of System Coverage,

Capacity, User Data-Rate and Movement Speed', IEEE Wireless Communications, vol. 27, no. 2, pp. 218–228, Apr. 2020, doi: 10.1109/MWC.001.1900333.

[9] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, 'User-Centric Cell-Free Massive MIMO Networks: A Survey of Opportunities, Challenges and Solutions', IEEE Communications Surveys & Tutorials, vol. 24, no. 1, pp. 611–652, 2022, doi: 10.1109/COMST.2021.3135119.

[10] A. D. La Oliva et al., 'Xhaul: toward an integrated fronthaul/backhaul architecture in 5G networks', IEEE Wireless Communications, vol. 22, no. 5, pp. 32–40, Oct. 2015, doi: 10.1109/MWC.2015.7306535.

[11] N. Molner, A. de la Oliva, I. Stavrakakis, and A. Azcorra, 'Optimisation of an integrated fronthaul/backhaul network under path and delay constraints', Ad Hoc Networks, vol. 83, pp. 41–54, Feb. 2019, doi: 10.1016/j.adhoc.2018.08.025.

[12] A. Younis, T. X. Tran, and D. Pompili, 'Fronthaul-Aware Resource Allocation for Energy Efficiency Maximization in C-RANs', in 2018 IEEE International Conference on Autonomic Computing (ICAC), Sep. 2018, pp. 91–100. doi: 10.1109/ICAC.2018.00019.

[13] N. Namburi, N. D. Rani, and D. Bera, 'Multi-User Transmit Beamforming for Achieving Higher Capacity and Reliability in 5G Standards', Wireless Pers Commun, vol. 124, no. 3, pp. 2211–2227, Jun. 2022, doi: 10.1007/s11277-021-09452-6.

[14] 5g Networks, 'Massive MIMO and Beamforming in 5G', 5G Networks. Accessed: Dec. 31, 2023. [Online]. Available: https://www.5g-networks.net/5g-technology/massive-mimo-and-beamforming-in-5g/

[15] K. M. O Hara, 'mmWave Massive MIMO Beamforming in Dynamic Urban Environments', Remcom. Accessed: Dec. 31, 2023. [Online]. Available: https://www.remcom.com/articles-and-papers/mmwave-massive-mimo-beamforming-in-dynamic-urban

[16] R. G. Stephen and R. Zhang, 'Joint Millimeter-Wave Fronthaul and OFDMA Resource Allocation in Ultra-Dense CRAN', IEEE Transactions on Communications, vol. 65, no. 3, pp. 1411–1423, Mar. 2017, doi: 10.1109/TCOMM.2017.2649519.

[17] J. Zhang, Y. Xiao, D. Song, L. Bai, and Y. Ji, 'Joint Wavelength, Antenna, and Radio Resource Block Allocation for Massive MIMO Enabled Beamforming in a TWDM-PON Based Fronthaul', Journal of Lightwave Technology, vol. 37, no. 4, pp. 1396–1407, Feb. 2019, doi: 10.1109/JLT.2019.2894152.

[18] B. E. Godana and D. Gesbert, “Coordinated beamforming in multicell networks with Channel State Information exchange delays,” in 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Sep. 2013, pp. 713–718. doi: 10.1109/PIMRC.2013.6666229.

[19] F. Zhou, G. Luo, Y. Liu, Y. Wang, and L. Fan, “Coordinated beamforming for heterogeneous small-cell networks with a non-ideal backhaul,” IET Communications, vol. 12, no. 5, pp. 595–602, 2018, doi: 10.1049/iet-com.2016.1451.

[20] G. O. Pérez, J. A. Hernández, and D. Larrabeiti, ‘Fronthaul Network Modeling and Dimensioning Meeting Ultra-Low Latency Requirements for 5G’, J. Opt. Commun. Netw., vol. 10, no. 6, p. 573, Jun. 2018, doi: 10.1364/JOCN.10.000573.

[21] R1. M. Jinno, ‘Elastic Optical Networking: Roles and Benefits in Beyond 100-Gb/s Era’, Journal of Lightwave Technology, vol. 35, no. 5, pp. 1116–1124, Mar. 2017, doi: 10.1109/JLT.2016.2642480.

[22] J. A. Hernández, G. Otero, D. Larrabeiti, and Ó. G. de Dios, ‘Dimensioning Flex Ethernet Groups for the transport of 5G NR fronthaul traffic in C-RAN scenarios’, in 2021 International Conference on Optical Network Design and Modeling (ONDM), Jun. 2021, pp. 1–3. doi: 10.23919/ONDM51796.2021.9492417.

[23] Y. Liao, S. A. Hashemi, H. ElBakoury, J. Cioffi, and A. Goldsmith, ‘Calendar Allocation Based on Client Traffic in the Flexible Ethernet Standard’, in ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Jun. 2020, pp. 1–6. doi: 10.1109/ICC40277.2020.9149293.

[24] W. Lu, J. Kong, L. Liang, S. Liu, and Z. Zhu, ‘How Much Can Flexible Ethernet and Elastic Optical Networking Benefit Mutually?’, in ICC 2019 - 2019 IEEE International Conference on Communications (ICC), May 2019, pp. 1–6. doi: 10.1109/ICC.2019.8761102.

[25] W. Xu and L. Liang, ‘On Coordinated Multi-point Transmission with Partial Channel State Information Via Delayed Feedback’, Wireless Pers Commun, vol. 75, no. 4, pp. 2103–2119, Apr. 2014, doi: 10.1007/s11277-013-1456-6.

[26] N.-S. Kim and Y. H. Lee, ‘Effect of channel estimation errors and feedback delay on the performance of closed-loop transmit diversity system’, in 2003 4th IEEE Workshop on Signal Processing Advances in Wireless Communications - SPAWC 2003 (IEEE Cat. No.03EX689), Jun. 2003, pp. 542–545. doi: 10.1109/SPAWC.2003.1319019.

[27] B. S. G. Pillai et al., 'End-to-End Energy Modeling and Analysis of Long-Haul Coherent Transmission Systems', Journal of Lightwave Technology, vol. 32, no. 18, pp. 3093–3111, Sep. 2014, doi: 10.1109/JLT.2014.2331086.

[28] R. S. Tucker, 'Green Optical Communications—Part I: Energy Limitations in Transport', IEEE Journal of Selected Topics in Quantum Electronics, vol. 17, no. 2, pp. 245–260, Mar. 2011, doi: 10.1109/JSTQE.2010.2051216.

[29] S. Wolf et al., 'DAC-Less Amplifier-Less Generation and Transmission of QAM Signals Using Sub-Volt Silicon-Organic Hybrid Modulators', Journal of Lightwave Technology, vol. 33, no. 7, pp. 1425–1432, Apr. 2015, doi: 10.1109/JLT.2015.2394511.

[30] J. M. Meredith, 'NR; Base Station (BS) radio transmission and reception.', 3GPP Specification 38.104, Jan. 2018. Accessed: May 10, 2024. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3202

[31] M. Klinkowski, 'Latency-Aware DU/CU Placement in Convergent Packet-Based 5G Fronthaul Transport Networks', Applied Sciences, vol. 10, no. 21, Art. no. 21, Jan. 2020, doi: 10.3390/app10217429.

[32] A. Fayad, M. Jha, T. Cinkler, and J. Rak, Planning a Cost-Effective Delay-Constrained Passive Optical Network for 5G Fronthaul. 2022.

[33] I. Sousa, N. Sousa, M. P. Queluz, and A. Rodrigues, "Fronthaul Design for Wireless Networks," Applied Sciences, vol. 10, no. 14, Art. no. 14, Jan. 2020, doi: 10.3390/app10144754.

[34] N. Gomes, P. Chanclou, P. Turnbull, A. Magee, and V. Jungnickel, "Fronthaul evolution: From CPRI to Ethernet," Optical Fiber Technology, vol. 26, Aug. 2015, doi: 10.1016/j.yofte.2015.07.009.

Chapter 2

5G and Beyond Fronthaul Networks, Functional Splits and Optimisation

2.1 Introduction

In this Chapter, an introduction to the foundation of 5G and beyond mobile networks is given, covering the topics of, fronthaul networks, functional splits, Flexible Ethernet, models for delays in fronthaul networks, physical layer transport in fronthaul networks, reinforcement learning and its applications in beyond-5G mobile networks for optimisations.

In Section 2.2 an overview of 5G networks, covering the scope of services, prospective frequency spectra, Massive MIMO (Multiple Input, Multiple Output), 5G core networks, and the Cloud radio network architecture for 5G networks is presented.

In Section 2.3 the fronthaul network architectures for 5G and beyond are discussed. The Section delves into topics such as the Common Public Radio Interface, Next Generation fronthaul Interface, functional splits in 5G fronthaul networks, and the advantages and disadvantages of functional splits. The Section also touches upon fronthaul data rates for different functional layer splits.

In Section 2.4 an introduction to Flexible Ethernet is given, covering the FlexE architecture, modes of operation, FlexE use cases, FlexE Multi-Frame Structure, and how FlexE is applied to functional splits and network slicing in 5G networks.

In Section 2.5 an introduction to different types of delays in packet networks is given. The introduction for the G/G/1 queuing model and Lindley's recursion are also given in the chapter. The delay variations at different levels of aggregation in a packet network are also discussed in the section.

In Section 2.6 the background theory of data transmission via discrete multitone modulation is given specifically for short link fibre optic fronthaul links. An introduction to different techniques of equalisation of DMT signals is also given in the chapter. In this Section, an introduction to Analogue and Digital fronthaul in the context of 5G networks is also given.

In Section 2.7 the introduction to reinforcement learning, starting with Bellman's equation is given. The Section further discusses Q-learning and Deep-Q learning as well as the algorithms for their implementation. The section also provides insights into the utilisation of reinforcement learning for improvement to optimise 5G networks.

2.2 Introduction to 5G Networks

2.2.1 Scope of Services of 5G Wireless Networks

The scope of 5G services has been categorized by ITU-T into three categories: enhanced mobile broadband (eMBB), ultra-reliable and low-latency communication (uRLLC) and massive machine type communication (mMTC) [1]. Enhanced mobile broadband speeds promise users speed up to 10Gbps on uplinks. These services in eMBB include multimedia media, virtual reality and augmented reality. Ultra-reliable and low latency communications are aimed at remotely driven semi-autonomous vehicles, performance of medical surgeries via remotely controlled robots and control of industrial manufacturing plants. Massive machine-type communication focuses on the massive number of Internet of Things (IoT) devices communicating in a network. As per targets laid down by the International Telecommunication Union telecommunication standardisation sector (ITU-T), the uRLLC services should not offer delays of more than 1 milli-second and mMTC should offer connectivity of one million devices per square kilometre [1] [2]. The targeted services for 5G wireless networks are also shown in Fig 2.1.

2.2.2 Massive MIMO

MIMO transmission systems have multiple antennas both at the transmitter and receiver. Transmission of data through multiple antennas creates space diversity in addition to frequency and time diversity. At the receiver, data is received through multiple antennas which can be combined by different techniques that can boost the signal-to-noise ratio (SNR). This greatly increases the spectral efficiency and in turn the throughput of the data transmission system. For long-term evolution (LTE)-Advance 8x8 MIMO is supported [4]. However, the proposed MIMO order for 5G is far greater than LTE. The number of transmit antennas at the base station ranges from 256 to 1024 [3]. The benefit of this large number of transmit

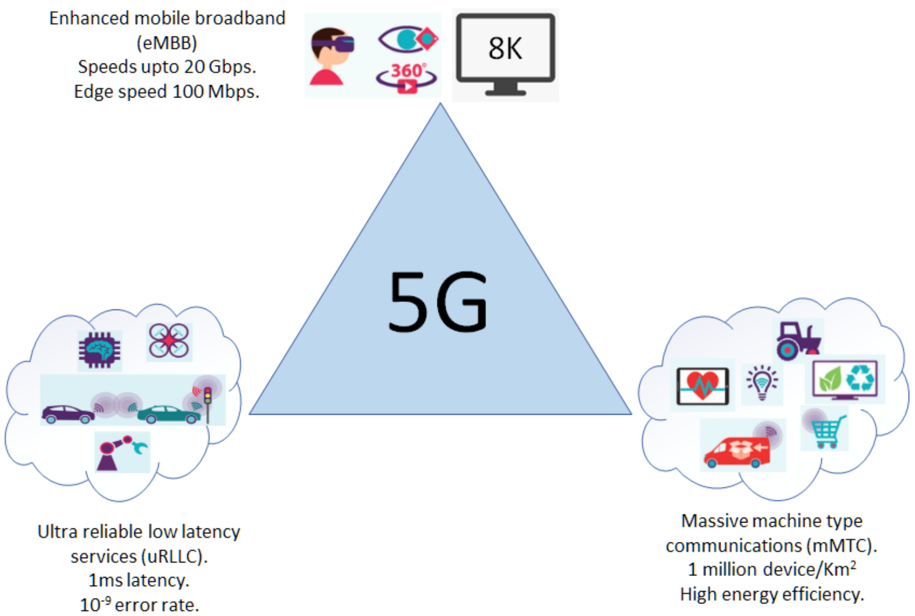


Fig. 2.1: 5G services triangle [1].

antenna array elements is that a base station can use beamforming to direct radio waves to multiple users simultaneously.

2.2.3 Cloud Radio Network Architecture for 5G Networks

Cloud radio network access (C-RAN) architecture is the basis for 5G networks [5]. It has its origin in distributed radio access network (D-RAN) architecture. In D-RAN architecture a baseband unit (BBU) is connected to many remote radio heads (RRH). A BBU dynamically allocates resources to remotely connected RRHs [5]. The function of RRHs is only related to radio frequency functions. In a D-RAN the interface between the BBUs and RRHs is through a common protocol radio interface (CRPI). The physical medium of the interface is through both microwave and fibre optic communication. The interface is termed "fronthaul". The D-RAN architecture is shown in Fig. 2.2. Manny BBUs are connected to the core via a fibre optic link which is called "backhaul". The traffic related to the user data, control and management and handover data flows on this link.

The architecture for D-RAN is not scalable and efficient enough to deliver the diverse traffic requirements of 5G networks. The alternative to D-RAN is cloud radio access network architecture (C-RAN) depicted in Fig. 2.3. In C-RAN all the BBUs are merged in the form of a centralised cloud. All the BBUs are then connected to a backhaul link to the core network. As CRAN centralizes the BBU in data centres, by consolidating processing

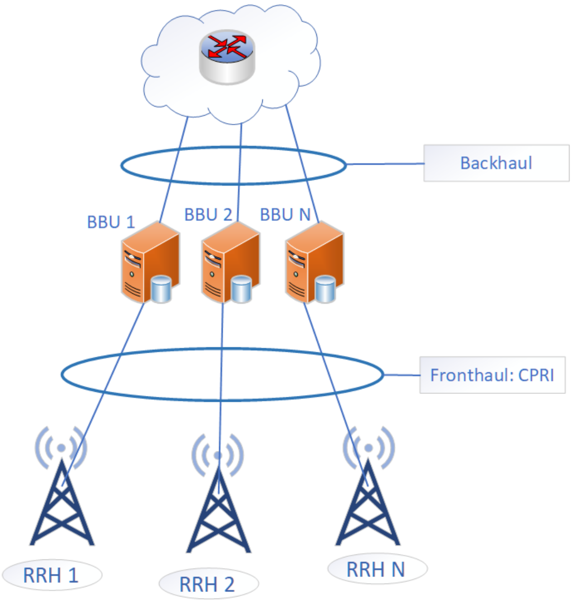


Fig. 2.2: D-RAN architecture [5].

resources, it reduces the overall energy consumption compared to the distributed approach of DRAN. Fewer physical sites mean less power required for cooling, lighting, and other infrastructure components. CRAN allows dynamic allocation of resources based on demand. As traffic increases, additional BBUs can be provisioned in the data centre. In contrast, DRAN has fixed resources at each site, limiting scalability. CRAN streamlines management by centralizing control. Configuration changes, software updates, and maintenance tasks are performed at the data centre. DRAN requires separate management for each site, leading to complexity and potential inconsistencies. CRAN optimises resource allocation, ensuring efficient utilisation of available spectrum. Coordinated scheduling and interference management enhance spectral efficiency. DRAN, with its distributed architecture, may suffer from interference and suboptimal resource allocation. CRAN can dynamically balance traffic across BBUs. During peak hours, the load can be shifted to underutilised BBUs. DRAN lacks this flexibility, potentially leading to congestion at specific sites [5].

In summary, for 5G networks, C-RAN brings easy expansion of network coverage, expansion of network capacity, network resource sharing and multi-cell collaborative signal processing. Fully centralised C-RANs can also facilitate the implementation of software-defined radios (SDRs) for dynamically adjusting the transmission capacity of the links [5].

The network architecture of 5G cellular networks should be agile and scalable. It should be able to accommodate new services on the pre-existing network. For this, new cells with different sizes have to be created within the pre-existing cellular infrastructure to cater for

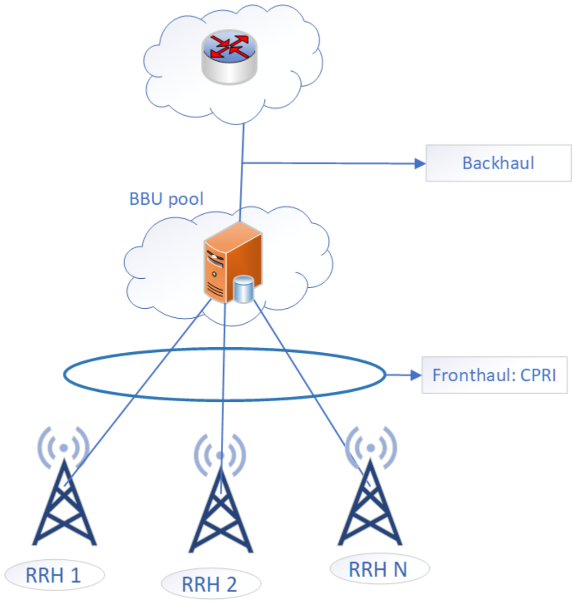


Fig. 2.3: C-RAN architecture. [5]

the changing demands of services in terms of data rate, coverage and latency. This results in network architecture that has a non-uniform network layout and is called heterogeneous cloud radio network architecture (H-CRAN) [5].

2.3 Fronthaul Network Architectures and Functional Splits

At a high level of abstraction, the conceptual diagram of a 5G network is shown in Fig. 2.4. A 5G network comprises user equipment (UEs), gnodeB (gNB), a backhaul network, aggregation nodes, and a mobile core data centre.

The functions of gNB are disaggregated or split into three different entities. These are termed central unit (CU), distributed unit (DU), and remote unit (RU). These entities can be either colocated or separated. The link between RU and DU is termed 'Fronthaul', DC and CU is termed 'Midhaul', and the link between DU and mobile core is termed 'Backhaul'. Overall these links are collectively termed 'Xhaul'.

The RU serves as the interface between the wireless antennas and the digital network. Its primary functions include signal transmission and reception. The RU converts analogue radio signals received from the antennas into digital signals suitable for transmission over packet-based networks. It handles the initial processing of the received signals, including filtering, amplification, and analog-to-digital conversion. The RU plays a crucial role in beamforming, adjusting the phase and amplitude of transmitted signals to optimise coverage

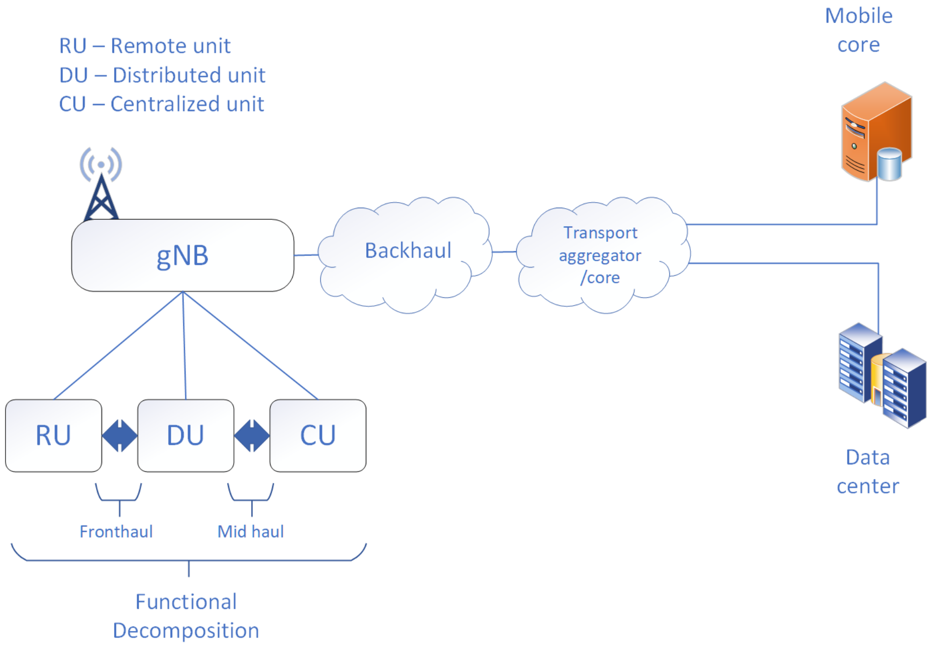


Fig. 2.4: 5G network abstraction. [6]

and capacity. RUs are deployed at cell sites, close to the antennas. RUs are designed to be compact, lightweight, and energy-efficient.

The DU is responsible for real-time processing and coordination between the RU and the core network. The DU runs on a Commercial Off-The-Shelf (COTS) server deployed at the cell site. It handles functions of layer-1 which involve physical layer processing, including modulation, coding, and channel estimation. The DU manages data link layer functions, such as scheduling, HARQ (Hybrid Automatic Repeat Request), and error correction. The DU is controlled by the CU and communicates with it via the F1 interface. It ensures efficient communication between the RU and the core network. The DU allocates radio resources to connected devices. DUs can be added or removed based on network demand, allowing scalability without significant physical changes.

The CU resides in the cloud or a centralised data centre. The CU handles higher-layer functions, including, RRC (Radio Resource Control), SDAP (Service Data Adaptation Protocol), and PDCP (Packet Data Convergence Protocol). The CU manages mobility, session management, and network control. The CU orchestrates network-wide functions, such as load balancing and edge computing. By coordinating multiple DUs, the CU optimises resource utilisation and ensures efficient network operation. The CU can be geographically separated from the cell sites, enabling centralised management and resource pooling.

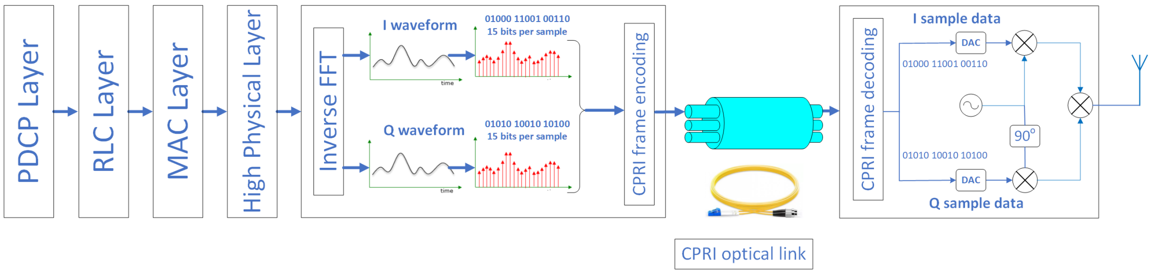


Fig. 2.5: Common Public Radio Interface (CPRI) based Fronthaul. [8]

Along with the main functions of routing and aggregation of data streams flowing in the fronthaul network, the aggregation nodes in Fig. 2.4, also perform the function of isolation of the slices of the 5G network. The isolation can be implemented via software, through SDNs, as discussed in the previous section and hardware via Flexible Ethernet (FlexE) enabled aggregation nodes [7]. The research in the thesis will focus on resource allocations in a FlexE calendar impacted by the latency in fronthaul networks.

2.3.1 Common Public Radio Interface

A common public radio interface (CPRI) was defined for the transportation of radio signals from baseband band units (BBUs) to remote radio heads (RRH) [8]. In CPRI, time-domain signals are first sampled by a high-speed analogue to digital converter (A/D) converter. Each sample is quantized using 15 bits. These bits are encapsulated in a CPRI frame at the BBU and then transported over a fibre optic link. At the RRH, the bits are then converted back to the analogue domain using a digital-to-analogue converter (D/A), to drive an RF upconverter for transmission over the wireless channel. The architecture of a CPRI-based fronthaul is depicted in Fig. 2.5.

CPRI is designed for high-speed serial communication between the Baseband Unit (BBU) and the Remote Radio Unit (RRU) over dedicated fibre links. However, the protocol itself introduces significant overhead due to framing, synchronisation, and control information. This overhead reduces the effective data rate available for actual user data transmission. CPRI specifies logical connections between the BBU and RRU, but these connections are relatively fixed. In dynamic network scenarios, where re-configurations are frequent, more flexible logical connections would be advantageous. CPRI defines specific line rates (e.g., CPRI Line Rates 1, 2, 3, etc.) for different use cases. However, these fixed rates may not align perfectly with the varying requirements of different network scenarios. To cope with the increasing user traffic, mm-wave spectra of up to 100 MHz and subcarrier spacing of 240 kHz will be used for 5G wireless networks. However, sampling of these large bandwidth spectra will put an impractical high load on the CPRI interface.

2.3.2 Next Generation Fronthaul Interface

The fronthaul for future 5G networks will have to accommodate different types of services at extremely high data rates and low latency from different 5G network slices. These requirements can only be met if the networking architecture is based on a flexible Ethernet networking topology. The key benefits of adopting Ethernet-based data transport in 5G fronthaul would be gains from statistical multiplexing, packet-switched operation and pooling [9]. Institute of Electrical and Electronics Engineers (IEEE) 1914 working group established the next-generation fronthaul interface (NGFI) body in 2016 to develop next-generation fronthaul for 5G. One of the projects of NGFI is P1914. The key tasks for this project were to develop an ethernet-based architecture for handling the control plane and user plane data rates, timing and synchronisation requirements and quality of service (QoS). Therefore the aggregation nodes in the fronthaul shown in Fig. 2.4, can be based on Ethernet switching or even routing.

The main difference between NGFI fronthaul and CPRI lies in their transport mechanisms and architectures. While CPRI uses a dedicated point-to-point link for each BBU-RRH connection and relies on fixed-rate transport, NGFI fronthaul leverages packet-based transport networks, enabling the sharing of a common infrastructure and providing greater flexibility and scalability. NGFI fronthaul can support multiple RRH connections over a single transport link, allowing for efficient utilisation of network resources.

2.3.3 Functional Splits in 5G Fronthaul Networks

As mentioned in the previous Section with the increasing data rate requirements of 5G networks, the sampling of user traffic data will result in impractically high bit rates for the CPRI interface. To overcome this constraint, different functional splits of the layers 1, 2 and 3 between the CU, DU and CU in 5G RAN are under consideration by the 3$^{\text{rd}}$ generation partnership project (3GPP) [10]. The functional splits are illustrated in the following Fig. 2.6. The factors that govern functional splits are data rate, number of layers in MIMO, latency, jitter, scalability to support advanced features, error correction and support for packet-optical networks [12].

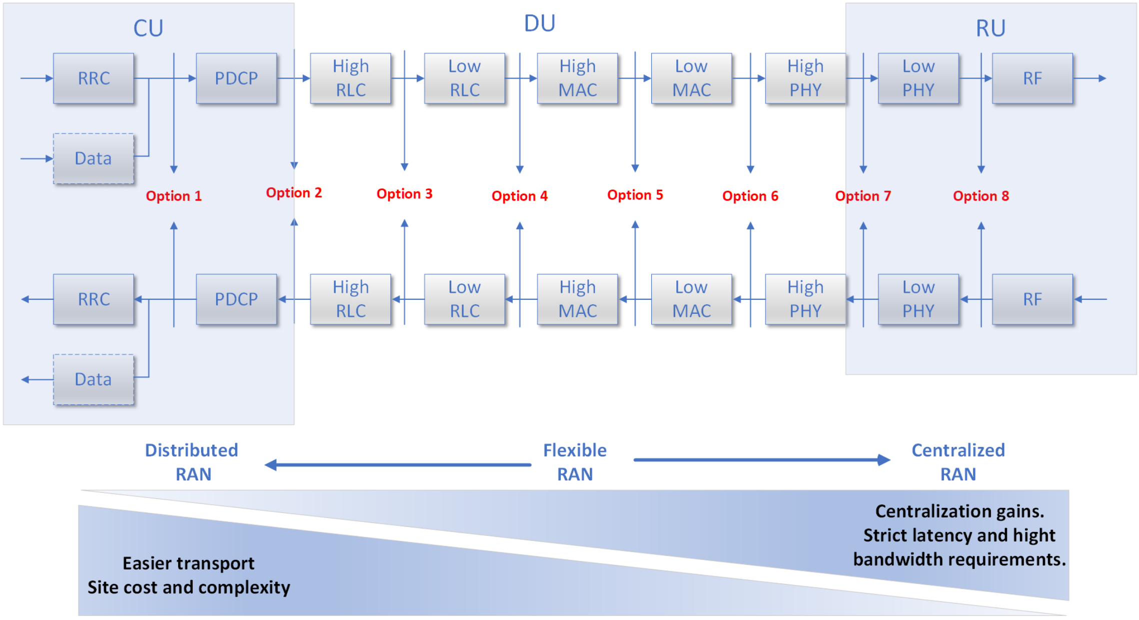


Fig. 2.6: Functional splits in 5G networks [11].

2.3.4 Advantages and Disadvantages of Functional Splits

As shown in Fig. 2.6, the split option 8 when shifted in the RU puts a high load on the fronthaul links between the RU and DU. This is so since the data rates in the fronthaul are dependent on the length of fast Fourier transform (FFT) length, the subcarrier spacing, and the number of antenna layers and bits per sample.

However, the main benefit of the PHY layer functions being performed in the RU results in gains of user data rate from CoMP, a key feature in current and 5G networks. Shifting the functions from the higher layers into the RU results in a significant reduction in fronthaul link load, since the data rates are dependent on user data, resulting in relaxed requirements of latency. These advantages come with the increase in complexity of the RU units, their cost and the diminished benefits of CoMP [12]. The benefits and shortcomings of different functional splits are discussed in further detail below [12].

Split option 8

Split option 8 is based on the standard CPRI. The key advantage of the option 8 split is that RU is very cheap and small. Only the analogue to digital, digital to analogue converter, and up/down converters are located in the RU. High levels of centralisation of the entire protocol stack can be done in CU. The separation of RF and physical layer enables operators to share

RF components. Cooperative multipoint communication (CoMP) in uplink and downlink can be supported to reduce interference in the user equipment (UEs).

With many advantages of physical layer spit option 8, there are many disadvantages. The key disadvantage of split option 8 is the high and constant bit rate for fronthaul. This scales with the number of antenna ports. It requires a high amount of processing for CPRI data compression at the DU and CU.

Split option 7-1

In split option 7-1, FFT is carried out in the RU. This enables the removal of the cyclic prefix and the guard subcarriers. This reduces the required bit rate. Further, uplink and downlink CoMP can be supported. Centralised scheduling can also be implemented. However, constant fronthaul bitrate is again required in the uplink and downlink.

Split option 7-2

In split option 7-2, the resource element mapper and precoding are implemented in the RU, which reduces the fronthaul link data rates. So, from this split downwards the fronthaul has a moderated and variable bit rate requirement. In this split a separate physical resource block (PRB) allocation protocol is necessary. Uplink and downlink CoMP can be supported in CU/DU. In this split again, centralised scheduling is possible. However, the disadvantage is the requirement of an in-band protocol for PRB allocation for users.

Split option 7-3

In split option 7-3, scrambling, modulation and layer mapper are implemented in the RU. This reduces the fronthaul bit rate requirement since each symbol transmits several bits depending on the modulation order. Forward error correction (FEC) is included in the CU-pool which establishes a close relation between FEC and the media access control (MAC) layer. Centralised scheduling is possible. Also, uplink and downlink CoMP are possible.

However, with these advantages, RU becomes more complex as modulation is included in RU. An in-band protocol is necessary for modulation, MIMO and PRB allocation.

Split option 6

In split option 6, the baseband functionality is split at the boundary of the MAC (Medium Access Control) and PHY (Physical Layer) layers. This means that all PHY functions remain in the RU. As a result, the fronthaul link carries only the transport blocks transmitted between the MAC and PHY layers. This reduction in payload leads to a significant decrease in the required fronthaul bit rate.

Since the RU handles both MAC and PHY functions, there is a concentration of resources at the RU. If the RU encounters issues or fails, it affects a larger portion of the network. The RU must accommodate both MAC and PHY processing, which can lead to complex hardware design and potentially higher costs. Since most PHY processing stays in the RU, the potential for centralised control and optimisation of baseband functions is reduced compared to other split options. The split between FEC and MAC introduces a need for low-latency communication for optimal performance.

Split option 5

Split option 5, also known as the Intra-MAC split, separates the functionalities within the MAC layer. This split real-time function of hybrid automatic repeat request (ARQ) is performed locally in the DU/RU. In this split, fronthaul latency requirements depend on the scheduling functions in the CU and DU/RU. With more functions in the DU/RU, it becomes more complex. Scheduling algorithms are also harder to define between DU/RU and CU.

The additional layer split introduces complexity for managing CoMP coordination. The scheduler needs to consider not just user data but also the coordination overhead for CoMP signals within the split MAC layer. Split option 5 might introduce additional latency due to the extra processing involved in scheduling within the split MAC layer. This could be detrimental to CoMP performance, as timely coordination between RUs is crucial for successful signal combining.

Split option 4

Split option 4, also known as the RLC-MAC split, separates the Radio Link Control (RLC) protocol from the MAC layer within the DU. This split receives the radio link control (RLC) protocol data units (PDUs) in the downlink direction and transmits the MAC service data units (SDUs) in the uplink direction. This allows low user data-dependent bitrates on fronthaul links. Separating RLC allows for independent optimisation of both layers. The CU can focus on higher-level functionalities like scheduling and mobility management, while the DU/RU handles reliable data delivery on the air interface.

This split is impractical in the case of shorter subframe sizes in 5G. Shorter subframes translate to shorter transmission times. Offloading RLC functions to the CU adds an extra communication hop between the CU and DU/RU. This additional communication overhead can significantly impact latency, especially for short subframes where the data payload itself might be small. The latency introduced by split option 4 might negate the benefits of shorter subframes designed for low-latency applications.

Split option 3

In split option 3, the RLC is separated into high RLC and low RLC. Low RLC consists of segmentation and high RLC consists of ARQ and other RLC functions. In this split multiple MAC entities are associated with one RLC thus scheduling is performed locally in the DU/RU, thus reducing the overall latency requirements in the fronthaul.

Split option 2

In this option, all functions except packet data convergence protocol (PDCP) and radio resource control (RRC) are centralised whereas all other functions are performed locally in the DU/RU. Since all functions except PDCP and RRC are performed locally, the data rate requirements for fronthaul are low.

Split option 1

In split option 1, all the functions except RRC are in DU/RU i.e., all the layer 2 and 3 functions are located in the DU/RU. It has low and cell load-dependent fronthaul data rate requirements. Also, the user is closer to the transmission point. With these advantages, the DU/RU becomes more complex.

2.3.5 Fronthaul Data Rates for Different Functional Layer Splits

Based on the literature review in [12] and [13] calculations were performed to calculate the required data rate for different splits for the 20 MHz spectrum in 5G. The split options considered here are for CPRI, PHY1 (option 7-1), PHY2 (option 7-2), MAC-PHY (option 6) and PDCP-RLC (option 2) splits. The formulations for bit rate calculations for the splits mentioned above are given as follows,

$$\text{CPRI bit rate} = 2\, N_{ant}\, N_{bits}\, \Delta f\, N_{fft} \tag{2.1}$$

$$\text{PHY1 bit rate} = 2\, N_{ant}\, N_{bits}\, \Delta f\, N_{sub} \tag{2.2}$$

$$\text{PHY2 bit rate} = \text{PHY1 bit rate} \times \text{load from user traffic} \tag{2.3}$$

$$\text{MAC-Phy bit rate} = 2\, N_{ant}\, N_{bits}\, \Delta f\, N_{sub}\, N_{bps}\, R_{code} \tag{2.4}$$

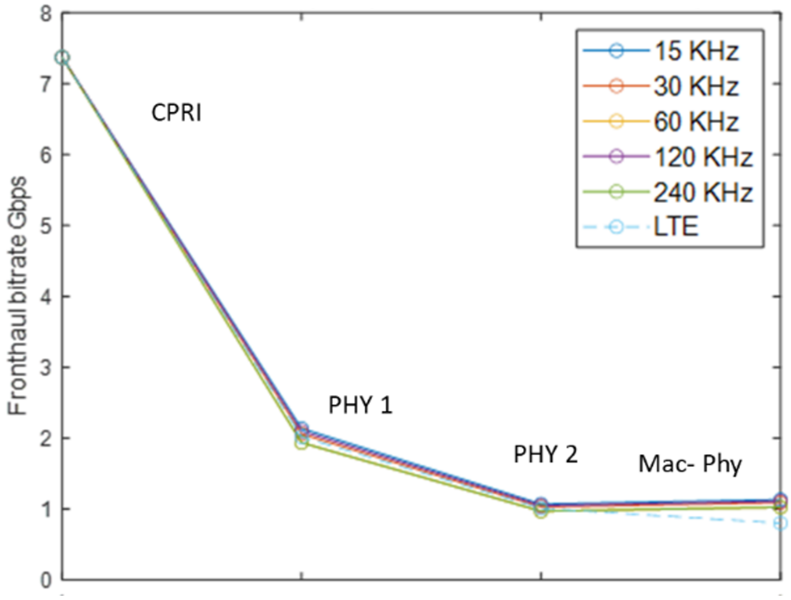


Fig. 2.7: Bit rate estimates for split options in 5G NR within 20 MHz spectrum.

where N_{ant} is the number of antenna layer, N_{bits} is the bits per sample, Δf is the sub carrier spacing, N_{fft} is the number of FFT points within 20MHz spectrum, N_{bps} is the number of bits per symbol, R_{code} is the code rate. Fig. 2.7 shows the bit rate required for the parameters of, N_{ant} = 8, N_{bits} = 15 and 7, N_{bps} = 6 (For 64 QAM), R_{code} = 948/1024, user traffic load = 70%, for different subcarrier spacing in 5G new radio (NR). Note that for LTE, R_{code} = 0.9258.

From the analysis of Fig. 2.7, it can be observed that for CPRI the approximate data rate was 7.5 Gbps and by implementing MAC-PHY split the data rate was reduced to approximately 1.5 Gbps for the different spectra for 5G NR. Thus it can be concluded that functional split significantly reduces the load on fronthual and improves bandwidth efficiency by up to 400%.

2.4 Introduction to Flexible Ethernet

Flexible Ethernet (FlexE) is proposed to be incorporated in the aggregation nodes in the fronthaul of 5G and beyond networks [14]. FlexE will enable bandwidth or slot allocation to different clients from different slices of the 5G network. The research in the thesis proposes the allocation of bandwidth with a finer granularity of 5Gbps in comparison to standard Ethernet rates of (10 Gbps, 25 Gbps, 40 Gbps, 50 Gbps, 100 Gbps, 200 Gbps, 400 Gbps). This results in improved optical bandwidth and energy efficiency of the transport network

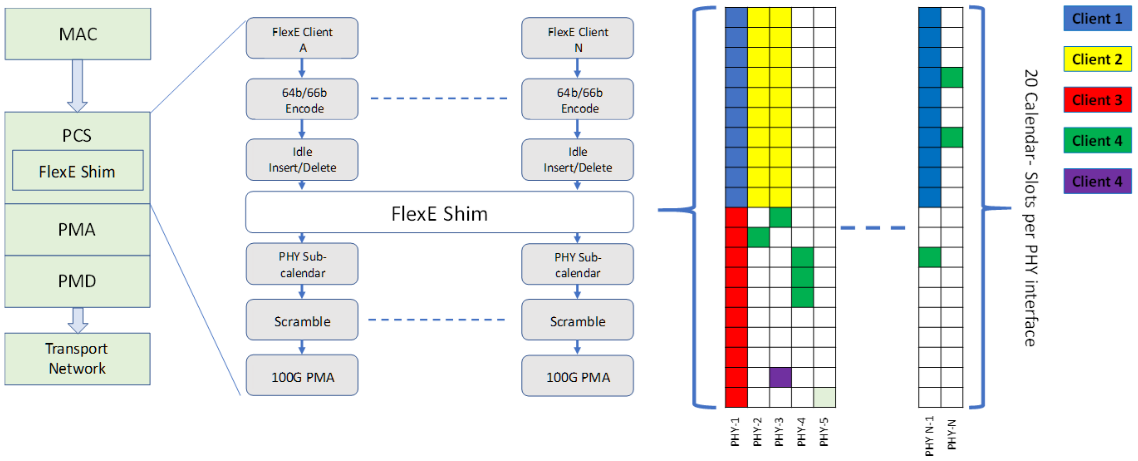


Fig. 2.8: Flexible Ethernet architecture [16].

segment of fronthaul. FlexE, being a TDM-based allocation scheme, dedicates separate bandwidth to each MAC client on the physical layer. This enables isolation among clients on the hardware level, which reduces the interference among them [15].

With FlexE incorporated into the fronthaul, the achievable user bit rate will depend on the over and under-allocation of slots in the FlexE calendar, as this will affect serialisation and buffering delays and, in turn, the fronthaul's total latency. The analysis of the impact of latency in fronthaul enabled by FlexE is one of the main focuses of the thesis. The architecture of FlexE and its different operating modes are discussed below,

2.4.1 FlexE Architecture

FlexE introduces a new layer called FlexE Shim within the Physical layer (PHY). The FlexE Shim is essentially a map or a calendar that distributes the incoming traffic from different clients onto a 100 Gb/s Ethernet (100GE) link. The calendar splits each 100GE link into 20 slots, each with a granularity of 5 Gb/s. An illustration of FlexE is shown in Fig. 2.8. The function of each sublayer is also discussed as follows [16],

Physical Coding Sublayer

The Physical Coding Sublayer (PCS) in Fig. 2.8 performs scrambling, frame alignment and 64B/66B coding. The 66B blocks consist of two header synchronisation bits. The blocks can be either data or control blocks. Control and data blocks are differentiated by two header bits: '01' (data) and '10' (control). The rest of the block consists of the payload. The payload is scrambled by the scrambler and the header bits are bypassed by the scrambler. After coding, the 66B code blocks are distributed among virtual lanes via round-robin switching.

Physical Medium Attachment

The Physical Medium Attachment (PMA) layer in Fig. 2.8 serialises and de-serialises the codes of the 66B block transmitted and received from the physical channel.

Physical Medium Dependent

The Physical Medium Dependent (PMD) layer defines the physical hardware such as connectors, used for interfacing of the Ethernet layer to the PHY link.

FlexE Shim/calendar

The FlexE Shim/calendar is situated within the PCS layer and is a resource allocation table used to allocate time division multiplexing (TDM) resources to clients. The slot positions for any client can be either contiguous or non-contiguous in the calendar.

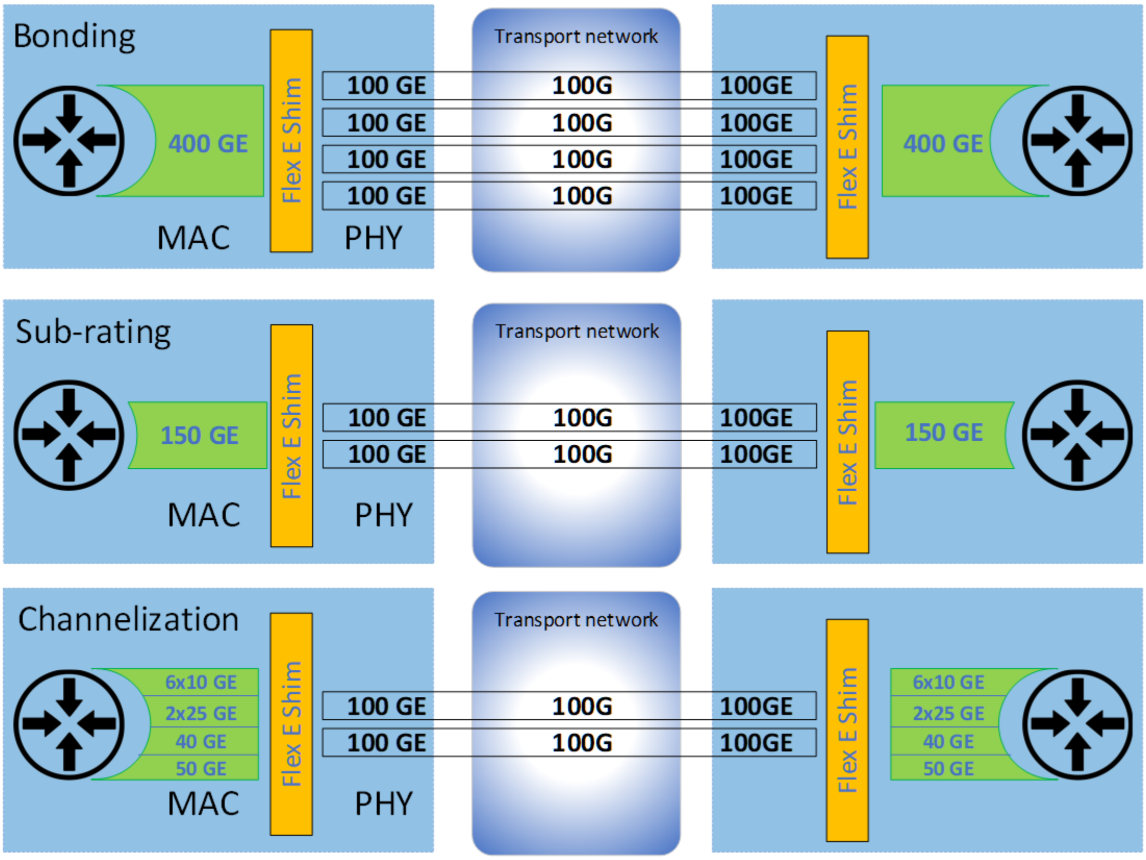


Fig. 2.9: Modes of operation of FlexE [18].

2.4.2 Modes of Operation of FlexE

FlexE has three modes of operation: Bonding, Channelisation and Sub-rating [17]. These operating modes are illustrated in Fig. 2.9. The bonding mode, shown in Fig .2.9 creates high-capacity data links by grouping or bonding several 100GE PHYs to transport MAC rates greater than 100 Gb/s. The channelisation mode in Fig. 2.9 is the provisioning of different client data rates with a granularity of 5 Gb/s, on the same or different 100GE PHY links. The sub-rating mode in Fig. 2.9 is the adjustment of low data rate clients onto a group of 100GE PHYs.

A frame of FlexE consists of 1023 repetitions of 20 calendar slots shown in Fig. 2.10. Each frame includes an overhead spanning one calendar slot (indicated by the black colour shaded slots in Fig. 2.10). As an example of channelisation, in a 400 Gb/s FlexE Shim, there will be 80 total calendar slots (4x20). The FlexE clients can then be distributed across the 80 total calendar slots. Fig. 2.11 shows the distribution of 10 Gb/s and 25 Gb/s clients (indicated by dark blue colour shaded slots). The 10 Gb/s client occupies two calendar slots in the first

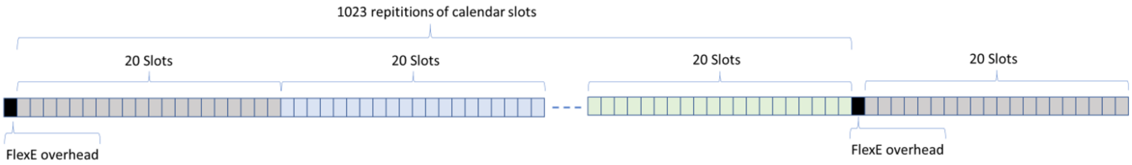


Fig. 2.10: FlexE frame structure [17].

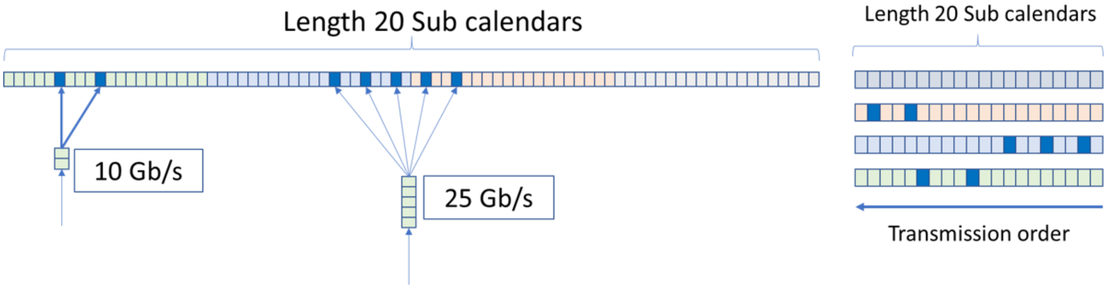


Fig. 2.11: Multiplexing of clients in FlexE calendar [17].

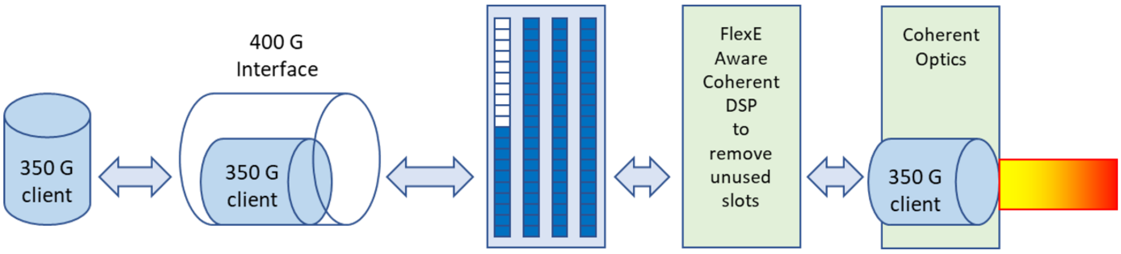


Fig. 2.12: Example of sub rating in FlexE [17].

set of 20 calendar slots and the 25 Gb/s client occupies 5 calendar slots that span the first and second sub-calendars [17].

The calendaring mechanism can also be used for sub-rating. An example is shown in Fig. 2.12). A 350 Gb/s client is mapped to a 400GE interface. Out of a total of 80 calendar slots, 70 calendar slots have been used for the transmission. The remaining 10 Calendar slots can be used to cater to other clients. If there are no other clients, a smart, coherent Digital Signal Processing (DSP) block can be used to discard the unused slots to get the desired rate.

2.4.3 FlexE Transport Modes

There are three modes of transportation of FlexE slots over the transport network, which are FlexE Unaware, FlexE Aware and FlexE Terminated modes [19]. The modes of FlexE transport are illustrated in Fig. 2.13 and are discussed below.

FlexE Unaware Transport

In FlexE-unaware transport mode, the transport network equipment is not aware that it carries the PHYs of the FlexE group as indicated by the Standard Ethernet interface block in Fig. 2.13(a). The disadvantage of this use case is that although the data rates in the transport network are adjusted according to the client bandwidth, the bandwidth of the transport network is not fully utilised. This is because IDLE markers are inserted in the calendar slots to match the data rate of the clients to the capacity of the different wavelengths of the transport network.

FlexE Aware Transport

In FlexE-aware transport mode, the transport network equipment is aware that it is carrying PHYs of a FlexE group as indicated by a Partial FlexE Shim in Fig. 2.13(b). The FlexE Shim can communicate with the transport network equipment through the optional FlexE Section Management. The key benefit of the FlexE Aware use case is that the transport network is aware that it is transporting FlexE calendar slots. As a result, all the calendar slots that have IDLE markers are dropped by the transport network. This results in more efficient utilisation of the capacity of the transport network. This is illustrated in Fig. 2.13(b), where the utilised bandwidth in the transport network is 150 Gb/s, compared to the FlexE unaware mode shown in Fig. 2.13(a), where the utilised bandwidth is 200 Gb/s.

FlexE Terminated Transport

In FlexE terminated transport mode, the transport network equipment terminates the FlexE Shim before the transport network link as shown in Fig. 2.13(c). The FlexE clients of the FlexE group are carried across the transport network. An additional FlexE Shim is inserted at the transport network egress ports. This makes the transport network aware of the client's traffic bit rate. This will enable the transport network to dynamically map the client's traffic onto the bandwidth available in the fibre optic links to enhance bandwidth efficiency. This eliminates the need for pre-provisioning fixed bandwidth capacities, reducing wasted bandwidth during periods of low traffic. The bandwidth allocation can be varied

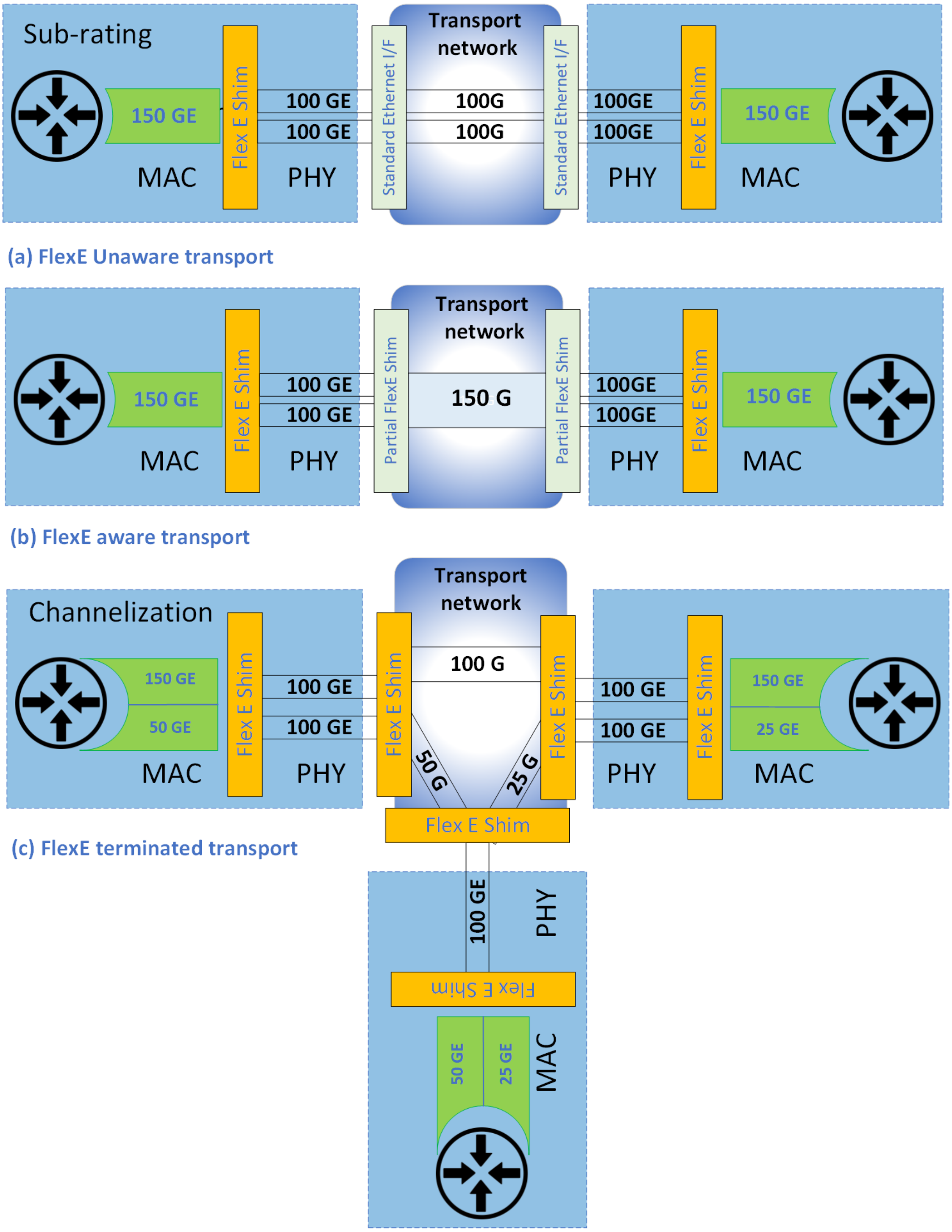


Fig. 2.13: FlexE transport modes, FlexE unaware transport (a), FlexE aware transport (b), FlexE terminated transport (c), [19].

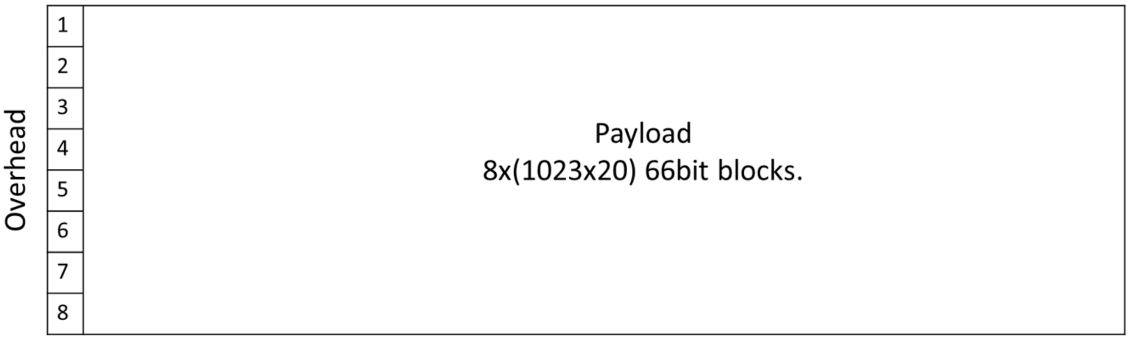


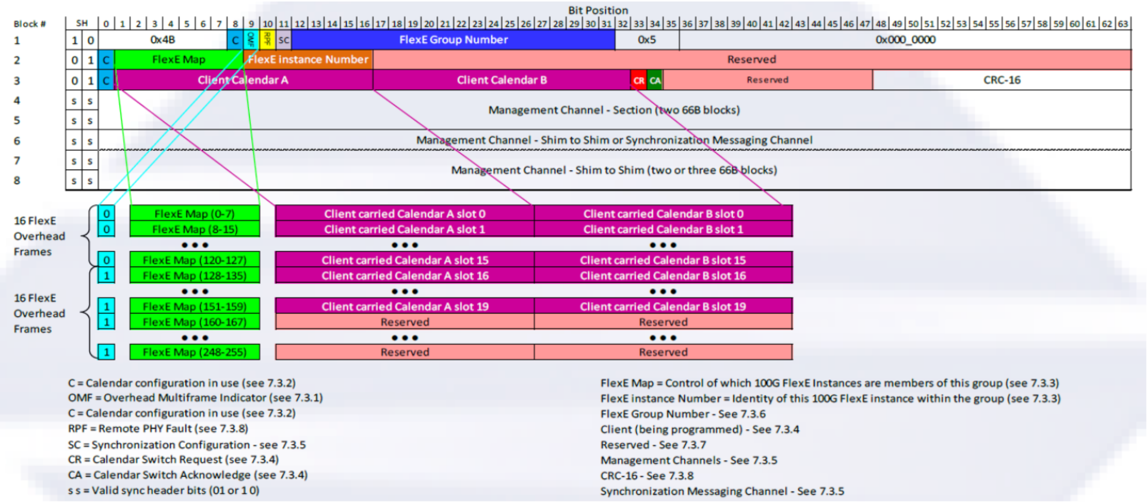
Fig. 2.14: FlexE multi-frame structure [19].

via bandwidth variable transponders which can vary the number of subcarriers in the DMT Multiplex or by varying the modulation levels for different FlexE bit rate transmissions with a fixed bandwidth. This also results in reduced power consumption of the network components, such as the transmitter modulator unit. The research in the thesis, therefore, focuses on the reduction of transmitted FlexE bit rates from fixed maximum data rates for enhancing bandwidth and power consumption efficiency, for fronthaul links with FlexE-enabled aggregation nodes operating in terminated mode.

2.4.4 FlexE Multi-Frame Structure

The FlexE multi-frame structure is shown in Fig. 2.14. The multi-frame of FlexE consists of 8 frames. As a result, the total payload consists of $8 \times 1023 \times 20 \times 66$ bits. The overhead consists of 8x66 bits. The transmission time of 1023, 66B blocks is 13.1 μs and thus the total transmission time of a multi-frame is 104.77 μs (13.1 μs $\times$ 8). The detailed functions of the overhead are illustrated in Fig. 2.15. The FlexE Group Number shown in the first row (indicated by the blue colour shaded block) points to a group of bonded 100GE PHYs. The FlexE Map in the second row consists of 8 bits (indicated by the green colour shaded block). It identifies 100GE PHYs within a group indicated by the FlexE Group Number. The complete map is constructed by a set of 32 FlexE Maps. So, in total, it consists of 256 bits.

The FlexE Instance Number in the second row (indicated by the red colour shaded block) is used for identifying the sequence of 100GE PHYs within the group being transmitted. The sub-calendar configuration within a 100GE PHY is given in the Calendar A and Calendar B bits in the third row of Fig. 2.15 (indicated by the magenta colour shaded blocks). The traffic dedicated to each client is given by the number of slots containing the Client ID. The total time required for the transmission of one sub-calendar is that of 32 multi-frames. So the total time for 32 multi frames of FlexE is 104.77 μs $\times$ 32 $=$ 3352.64 μs which is quite high


Bit Position
Block #
SH
1
2
3
4
5
6
7
8
0x4B
C
OMF
RPF
SC
FlexE Group Number
0x5
0x000_0000
FlexE Map
FlexE Instance Number
Reserved
Client Calendar A
Client Calendar B
CR
CA
Reserved
CRC-16
Management Channel - Section (two 66B blocks)
Management Channel - Shim to Shim or Synchronization Messaging Channel
Management Channel - Shim to Shim (two or three 66B blocks)
16 FlexE Overhead Frames
16 FlexE Overhead Frames
FlexE Map (0-7)
FlexE Map (8-15)
FlexE Map (120-127)
FlexE Map (128-135)
FlexE Map (151-159)
FlexE Map (160-167)
FlexE Map (248-255)
Client carried Calendar A slot 0
Client carried Calendar A slot 1
Client carried Calendar A slot 15
Client carried Calendar A slot 16
Client carried Calendar A slot 19
Client carried Calendar B slot 0
Client carried Calendar B slot 1
Client carried Calendar B slot 15
Client carried Calendar B slot 16
Client carried Calendar B slot 19
Reserved

C = Calendar configuration in use (see 7.3.2)
OMF = Overhead Multiframe Indicator (see 7.3.1)
C = Calendar configuration in use (see 7.3.2)
RPF = Remote PHY Fault (see 7.3.8)
SC = Synchronization Configuration - see 7.3.5
CR = Calendar Switch Request (see 7.3.4)
CA = Calendar Switch Acknowledge (see 7.3.4)
s s = Valid sync header bits (01 or 1 0)

FlexE Map = Control of which 100G FlexE Instances are members of this group (see 7.3.3)
FlexE instance Number = Identity of this 100G FlexE instance within the group (see 7.3.3)
FlexE Group Number - See 7.3.6
Client (being programmed) - See 7.3.4
Reserved - See 7.3.7
Management Channels - See 7.3.5
CRC-16 - See 7.3.8
Synchronization Messaging Channel - See 7.3.5

Fig. 2.15: FlexE multi-frame structure with overheads. (Snapshot from [21])

if dynamic traffic requirement changes are considered at the RU. Calendar reconfiguration is implemented by sending a request through the Calendar switch Request (CR) bit and transmission of the new calendar takes place after the Calendar switch Acknowledge (CA) bit is received. The CR and CA bits are indicated in the third row in Fig. 2.15 using red and green colour shaded blocks. The communication between different FlexE Shims is carried out using the section management channels in rows 4 to 8 of the overhead frame.

2.4.5 FlexE for Functional Splits and Network Slicing in 5G Networks

The choice of split options in 5G networks can lead to a reduction in the data rate requirements on the fronthaul. The choices of the functional splits are limited by latency constraints. For high-layer splits (HLS), the efficiency of the conventional Ethernet-based mid-haul network (without FlexE) can be improved if FlexE operational modes of grouping and channelisation are deployed. The data from the clients in HLS can be multiplexed using statistical multiplexing. The delays caused by queuing in the Ethernet mid-haul can be tolerated due to relaxed requirements for latency. In low layer splits (LLS) there can be an aggregation of several RUs resulting in high data rates and due to stringent latency requirements, statistical multiplexing may not be feasible as it may lead to queuing delays. Therefore, peak data rates may be assigned to the FlexE tunnels to avoid queuing delays. This flexibility of using a varying degree of statistical multiplexing in the Ethernet fronthaul can be very effectively provisioned in hardware by FlexE.

To cater to different traffic types, the slicing of 5G networks is envisioned. Different slices will be transported through different functional splits in fronthaul. This will result in different fronthaul latency requirements. A single Ethernet fronthaul will not cater to every slice's latency and bandwidth requirements. As a result, several Ethernet channels will be merged or split using FlexE to accommodate different splits. For practical implementation, a unified fronthaul control plane will also be required. This has been recently demonstrated using FlexE via experimentation and orchestration using SDN [22]. In this demonstration, a separate Physical Network Controller (PNC) is used to configure the underlying optical network resources in terms of Optical Distribution Units (ODUs). Another PNC controller is used to configure the grouping of the physical Ethernet interfaces. The clients are notified about the configuration of the FlexE path via the SDN controller. The entire experiment is performed on a hardware testbed called ADRENALINE.

In another experiment, FlexE has been demonstrated by Huawei in the 5G-PICTURE project, Deliverable 4.2 [23]. In this experiment, FlexE clients of 75 Gb/s and 25 Gb/s are configured. On average, the latency is 60 µs. However, the latency of each slot of the FlexE calendar is observed to be 0.5 µs, which is independent of the packet length of the incoming data. The advantage of using FlexE is that non-standardised client data rates can be transmitted over the standardised 100 Gb/s Ethernet. Before the client's data is transmitted on the FlexE, the PCS layer performs the 64/66B coding. In this process, if no traffic is encountered by the PCS layer it inserts an IDLE marker in the octets to fill up the bytes. This marker can then be utilised to release the bandwidth for provisioning of traffic to other clients, thus improving the bandwidth efficiency of the network. A bandwidth efficiency of up to 95 % with a latency of 12.8 ns has been reported in [24].

In the above-mentioned theoretical and experimental demonstrations of FlexE, there has been no mention of the total time consumed in establishing the E2E provisioning of the FlexE tunnel. Nor is there any mention of which part of the fronthaul the FlexE is dedicated to. However, the latencies demonstrated are very low, compared to the HLS and LLS split requirements.

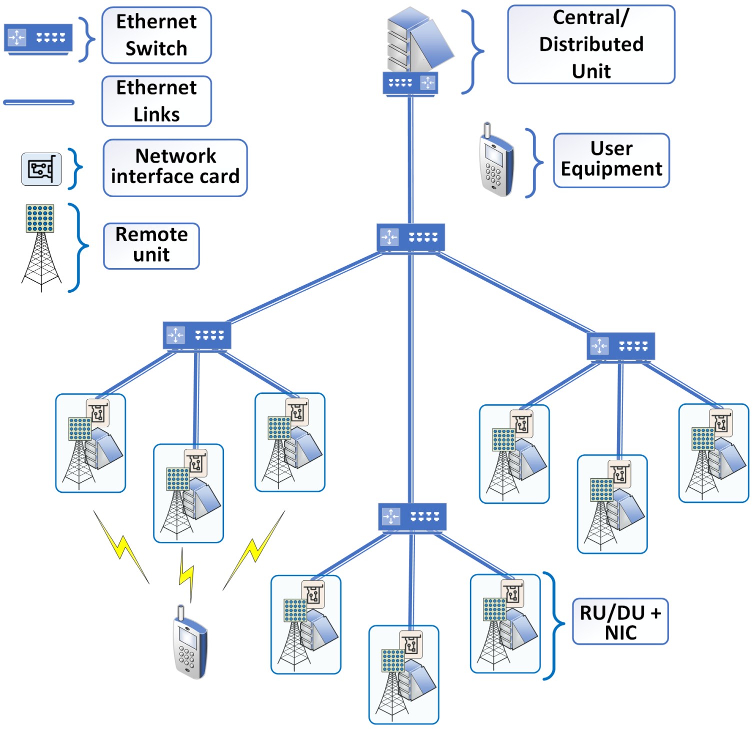


Fig. 2.16: Packet network with Ethernet-based aggregation.

2.5 End-to-End Delays in Fronthaul Networks and Queuing Models

To model, the end-to-end delay of different paths in a fronthaul network, a generalised network topology of a 5G fronthaul with Ethernet-based aggregation is shown in Fig. 2.16 [25]. This network architecture was used as a reference for simulations to test the model for the latency in fronthaul, with the results discussed in Chapter 3. Note that the Ethernet-based aggregation nodes may also be equipped with FlexE technology.

In the uplink in Fig. 2.16, the mobile user traffic is first aggregated and packetized by RUs or combined remote/distributed RU/DUs. Ethernet-based aggregation nodes further aggregate the fronthaul traffic from multiple (RU/DUs). The traffic from different Ethernet-based aggregation nodes is finally aggregated in the destination node which acts as a CU or a combined CU/DU unit. The RU/DUs, Ethernet aggregation nodes and CU/DUs are linked via fibre optic links. In the downlink, the traffic from CU/DU is disaggregated by Ethernet-based aggregation nodes and RU/DUs.

With the Ethernet-based aggregation in the fronthaul network, different delays will be added to the packetized user data when it propagates from a source to a destination node. The

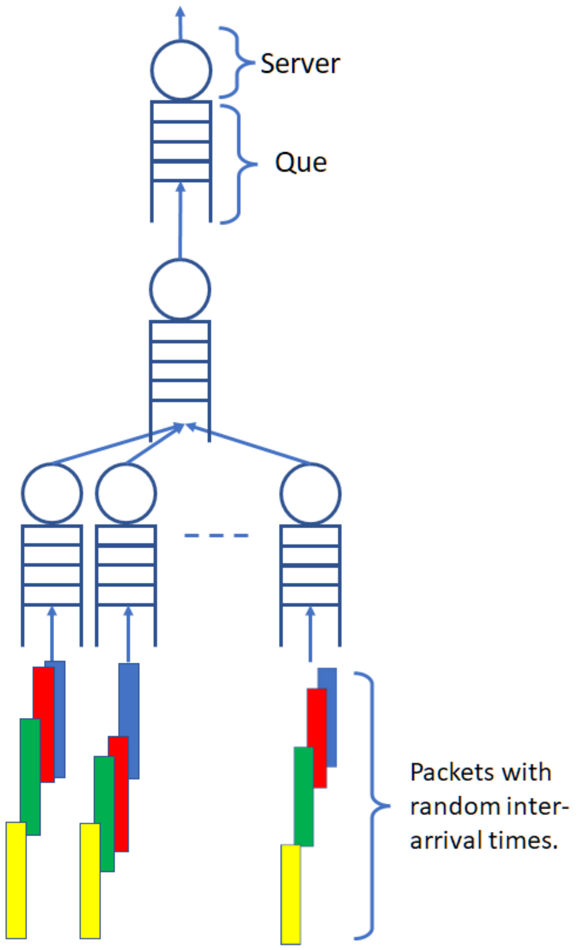


Fig. 2.17: Tandem queuing system for Fig. 2.15.

total delay between a source and a destination node is termed end-to-end delay. The end-to-end delay in any packet-based network includes both deterministic and random components. Deterministic components arise primarily from packet serialisation for fixed packet lengths, propagation delay, and overhead processing [26]. The random component arises from serialisation delays for random packet lengths, queuing delays in Ethernet interfaces in RUs and queuing delays in Ethernet switches in the network.

In essence, modelling a network's end-to-end delay (such as the system in Fig. 2.16) involves understanding the total delay experienced by packets, in a system with multiple queues and servers connected in tandem. This concept is illustrated in Fig. 2.17. Here, each combination of a queue and server represents a node from Fig. 2.16, which could be an RU or an Ethernet switch. At the bottom of Fig. 2.17 different entities or packets arrive at different queues at the bottom with different arrival rates and random inter-arrival times. The packets are then serviced by a server at the top of each queue in Fig. 2.17.

The time packets spend waiting in a queue, as shown in Fig. 2.17, to be served or processed by the server is referred to as 'queuing delay' [26]. In this thesis, it is also termed 'waiting time'. As the packets propagate in the network in the uplink direction in Fig. 2.17, they pass through different queues in the network. Each queue adds its queuing delay to the packets propagating through the network, adding to the total end-to-end delay of a path in the fronthaul network. The queuing delays are random and highly depend on the statistics of the inter-arrival times packets.

A 'server' in Fig. 2.17 represents an output port of a network node, such as the one shown in Fig. 2.16, operating at the speed of the output link. The time taken for packets to be processed or transmitted out of a node by the server is called 'serialisation delay' in this thesis [26]. The higher the speed of the output port of a node, the lower the serialisation delay of the packets with a fixed length. As the packets propagate in the network in the uplink direction in Fig. 2.17, they pass through different servers in the network. Each server adds its serialisation delay to the packets propagating through the network, adding to the total end-to-end delay of a path in the fronthaul network. The serialisation delay can be random if the lengths of the packets in the traffic are random.

Propagation delay refers to the time required for packets to traverse an optical fibre channel and is determined by the channel's length [26]. Fabric delay in nodes, such as those in Fig. 2.16, refers to the time required for a packet to move from the input port to the desired output port of the node via the switching fabric [26]. Therefore, the total end-to-end delay of a path in the fronthaul network can be considered as the sum of queuing delays, serialisation delays, propagation delays, and fabric delays [26]. However, fabric and propagation delays will be excluded from the results discussed in Chapter 3.

Note that one of the key challenges in modelling the end-to-end latency of a path is modelling the standard deviation of inter-arrival times of packets at the egress ports of Ethernet switches at different levels of aggregation in a network. The standard deviation of inter-arrival times of packets at the egress port of an aggregation node changes significantly from the standard deviation of inter-arrival times of packets at the ingress ports. The reason is that the randomness in inter-arrival times of packets smooths out after the packets leave the queues and are serialised at the egress ports of the Ethernet switches. This results in less randomness in inter-arrival times of packets reaching the ingress ports of Ethernet switches at a higher level of aggregation. The randomness also reduces with the increase in the load as at higher loads the bit rate increases resulting in smaller gaps between consecutive packets.

The process is also depicted in the following Fig. 2.18. The packets with different colour codes at the input of RUs in Fig. 2.18 have different and large inter-arrival times labelled as, Δt. When packets exit the output port of the NIC in the RU, the inter-arrival time

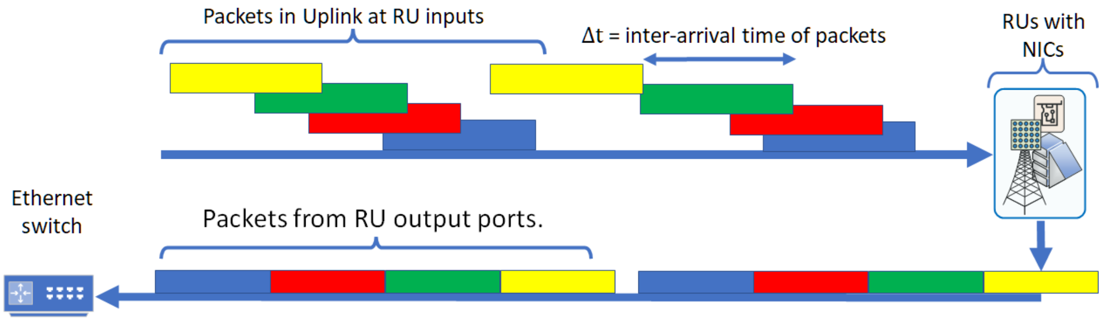

Packets in Uplink at RU inputs
Δt = inter-arrival time of packets
RUs with NICs
Ethernet switch
Packets from RU output ports.

Fig. 2.18: Packets at input and output ports of RU or Ethernet switch in Fig. 2.15.

of packets reduces significantly and reaches the input ports of the Ethernet switch in Fig. 2.16. The alteration of the inter-arrival times of packets at the Ethernet switches initiates the requirement for the modelling of the statistics of the inter-arrival times of packets. The reduction in inter-arrival times of packets will increase when the packets traverse through a network, from Ethernet switches at a lower level of aggregation to Ethernet switches at a higher level of aggregation in a fronthaul network such as in Fig. 2.16.

2.5.1 G/G/1 Queuing Model

A queuing system is a generalised mathematical model to represent congestion in a system. In the case of networks, a queuing model aids in understanding or predicting the delays associated with packets that traverse through different aggregation nodes. The delays are predicted as a function of the load on the aggregation nodes. The classical queuing models such as M/M/1 assume that the inter-arrival time of packets has a Poisson distribution, and the service time of packets also has a Poisson distribution [25]. The M/G/1 queuing model assumes that the arrival time of packets has a Poisson distribution while the service times of packets have a generalised distribution if served by a single server [25]. However, in networks, the statistics of the inter-arrival time of packets may not have a Poisson distribution.

Specifically in 5G networks and beyond, different service/slice traffic may show bursty or continuous behavior. Aggregating such traffic patterns will result in traffic having a distribution different from the Poisson distribution of the inter-arrival time of packets. In 5G networks, the overall traffic will comprise of a mix different traffic from different slices of 5G networks such as eMBB, uRLLC and massive machine-type communications (mMTC). In uRLLC packets can range from 32 bytes to 256 bytes, while eMBB packets can range from 128 bytes to 1500 bytes [20]. The concurrent flow of different traffic flows in fronthaul will result in packet lengths having a random distribution different from the Poisson distribution.

So the G/G/1 model was used to model the queuing delays in the aggregation nodes in the research discussed in the thesis since the G/G/1 queuing model assumes a generalised distribution of inter-arrival times and service times.

According to the G/G/1 queuing model, the mean waiting time $T_{w,i}$ of packets in a queue in node i in a path from RU to CU/DU in Fig. 2.16 can be expressed as [28],[29]:

$$T_{w,i} \approx \frac{1}{\mu_{r,i}} \frac{\rho_i}{(1-\rho_i)} \frac{{C_{a,i}}^2 + {C_{r,i}}^2}{2}, \quad \forall\, i \in \{1,2,3\} \tag{2.5}$$

where ρ_i is the total load on an RU or an Ethernet switch,

$$\rho_i = \frac{\lambda_{t,i}}{\mu_{r,i}}, \tag{2.6}$$

where, $\lambda_{t,i}$ is the arrival rate of packets (packet/s), and $\mu_{r,i}$ is the packet service rate (packet/s), at the node i defined as follows,

$$\mu_{r,i} = \frac{R_{o,i}}{\overline{L_p}}, \tag{2.7}$$

and,

$$C_r = \frac{\sigma_{r,i}}{1/\mu_{r,i}}, C_a = \frac{\sigma_{a,i}}{1/\lambda_{t,i}}, \tag{2.8}$$

where C_a is the coefficient of variation of the inter-arrival time of packets, C_r is the coefficient of variation of service rate of packets, $\sigma_{r,i}$ is the standard deviation of service time of packets and $\sigma_{a,i}$ is the standard deviation of the inter-arrival time of packets.The standard deviation of service times of packets $\sigma_{r,i}$ is defined as follows,

$$\sigma_{r,i} = \frac{\sigma_{\overline{L_p}}}{R_{o,i}}, \tag{2.9}$$

where $\sigma_{\overline{L_p}}$ is the standard deviation of packet length and $R_{o,i}$ is the service rate of the packets at the egress ports of the source or aggregation nodes. The total packet arrival rate at aggregation nodes is equal to the sum of packet arrival rates at the lower level of aggregation nodes. The packet arrival rate at Ethernet switches in Fig. 2.16 is equal to the sum of packet arrival rates at the lower-level Ethernet switches represented as follows,

$$\lambda_{t,i} = \sum_{k=1}^{J} \lambda_{t,i-1}(k) \;\forall\, i \in \{2,3\}, \tag{2.10}$$

where J is the number of lower-level Ethernet switches linked to a switch at level i.

2.5.2 Lindley's Recursion-Based Queuing Model

The G/G/1 queuing model assumes exponential distribution of the waiting times in a queue for packets [29]. This, however, may not be the case in real-world packet networks, as the waiting time of a packet in a queue depends on the statistics of the inter-arrival times of packets and the statistics of the distribution of the lengths of the packets [30]. This assumption can also lead to inaccuracy in the estimates of the mean waiting times and percentile delays of packets traversing through the fronthaul. So considering this limitation in the G/G/1 queuing model, the modelling of waiting of packets in queues in aggregation nodes via Lindley's recursions [31] is also explored in the research.

Lindley's recursion model utilises the exact statistics or probability density function (PDF) of the inter-arrival time of packets and the service times of packets, for modelling the waiting times of the packets in aggregation nodes. The PDF of service times of packets depends on the statistics of the packet lengths. In this method, the waiting time for a current packet, $T_{w,n+1}$, is given as follows,

$$T_{w,n+1} = \max\{0, B_n + T_{w,n} - A_n\}, \tag{2.11}$$

where, A_n is the inter-arrival time of a packet preceding the current packet at a queue, W_n is the waiting of the packet preceding the current packet and B_n is the service time of the packet preceding the current packet. The PDF of the waiting times of packets at an aggregation node can be modelled by convolving the PDFs of waiting times and service times iteratively for many packets until the PDF converges after multiple iterations:

$$f_{T_{w,n+1}}(t) \approx f_{B_n}(t) \otimes f_{A_n}(-t) \otimes f_{T_{w,n}}(t), \tag{2.12}$$

where, $f_{T_{w,n+1}}(t)$, is the PDF of the waiting time of the current packet, $f_{B_n}(t)$, is the PDF of the service time of the current packet, $f_{A_n}(-t)$, is the PDF of the interarrival times of packets flipped in time to accommodate subtraction operation in Eq 2.11 and, $f_{T_{w,n}}(t)$, is the waiting time for the preceding packet. The pdf, $f_{B_n}(t)$, depends on the statistical distribution of the packet lengths configured in the simulation setup which was assumed to have a Normal distribution since the packet lengths in the simulation setup also had a Normal distribution. The pdf, $f_{A_n}(-t)$, for RUs is the same as configured in the simulation setup. For aggregation nodes, the PDFs of inter-arrival times of packets were modelled using simulation results discussed in Chapter 3. After modelling the PDF of waiting times, $f_{W_{n+1}}(t)$, the mean waiting times in queues, $T'_{w,i}$, were calculated using the following relation,

$$T'_{w,i} = \int f_{T_{w,n+1}}(t) * t \, dt, \tag{2.13}$$

where i is the index of aggregation nodes in the fronthaul and t is the period over which the PDF is modelled.

As discussed above, Lindley's recursion model depends on the iteratively convolving the PDFs of inter-arrival times of packets and their service times, so it is more computationally exhaustive than the G/G/1 queueing model. The process of computation of PDF of waiting times of packets using Lindley's recursion is also described in the Algorithm 1, [32]. In step 1, the PDFs of inter-arrival times of packets, $f_{A_n}(-t)$, and service times of packets, $f_{B_n}(t)$, are given as input to the algorithm. The iterative convolution of PDFs of inter-arrival times of packets, service times of packets, and PDF of waiting times of packets in the previous iteration, $f_{T_{w,n}}(t)$, is performed in step 8. The iterations continue till the PDF, $f_{T_{w,n+1}}(t)$, does not converge to a specific shape. The convergence is evaluated using the difference or norm, $\Delta^{(l)}$, between the PDFs from consecutive iterations, $f_{T_{w,n+1}}(t)$ and $f_{T_{w,n}}(t)$. When the norm reduces below a certain threshold or tolerance Tol, the recursions stop.

Algorithm 1 Lindley's recursion for modelling of latency

```
Input: $f_{A_n}(t)$,$f_{B_n}(t)$,$Tol$
Output: $f_{T_{w+1,n}}(t)$
Initialize: $f_{W_1}(t) = \delta$
Initialize: $\Delta^{(1)} = 0$
$f_{A_n}(-t) = fliplr(f_{A_n}(t))$
$l \leftarrow 1$
while $\Delta^{(l)} > Tol$ do
    Convole the PDFs
        $f_{T_{w,n+1}}(t) \approx f_{B_n}(t) \otimes f_{A_n}(-t) \otimes f_{T_{w,n}}(t)$
    Integrate the negative part of PDF $f_{T_{w,n+1}}(t)$ and assign it to PDF at $t = 0$
        $f_{T_{w,n+1}}(0) \approx \int_{-\infty}^{0} f_{T_{w,n+1}}(t)$
    Set the negative part of the PDF $f_{T_{w,n+1}}(t)$ to zero
        $f_{T_{w,n+1}}(t) \approx 0 \quad \forall \quad t \leqslant 0$
    Calculate the Norm $\Delta^{(l)}$ of the PDFs
        $\Delta^{(l)} = \left[ \int_{0}^{\infty} \left\{ f_{T_{w,n+1}}(t) - f_{T_W}(t) \right\}^2 \right]^{\frac{1}{2}}$
    $l \leftarrow l + 1$
end while
```

2.6 Discrete Multitone Modulation for 5G Fronthaul Networks

Discrete multitone modulation (DMT) is based on orthogonal frequency division multiplexing (OFDM) for the transmission of data through a channel. In DMT modulation only the real part of the signal is transmitted through the channel, thus eliminating the need for coherent detection of the received signal. Only direct detection of the received signal is required if deployed in fibre optic communication links, thus reducing the cost of the receiver [33]. The transmitted signal also retains immunity against frequency selective fading [34]. Fading at certain frequencies in fibre-optic communication can occur, due to the linear and nonlinear impairments of the fibre optics, nonlinear frequency responses of the modulator, laser source and the photodetector [35].

The use of DMT signal transmission has widely been used in data centre interconnections (DCI) [36]. DMT has also been proposed for fronthaul for 5G wireless communications [37] since the distances between the RU and DU are not expected to be very long.

If the transport segment of the fronthaul networks is equipped with BVTs enabled by DMT transmission [38], then the slots of FlexE can be transmitted following the input traffic demands from different MAC clients. The transmission of 5Gbps slots of FlexE can be carried by either varying the number of data sub-carriers or the bandwidth of each sub-carrier. The dynamic allocation of slots will result in enhanced bandwidth and power efficiency.

2.6.1 Background Theory of DMT Transmission

DMT signal transmission is based on orthogonal frequency division (OFDM). Using the property of Hermitian symmetry, in which the frequency-domain signal is flipped and concatenated with the original frequency-domain signal before the inverse Fourier transform (IFFT) operation, only the real part of the signal is transmitted. The mathematical principle of discrete multitone modulation is given in detail in [33] as well as given below.

For a frequency domain signal $a[k]$, the complex inverse discrete Fourier transform (IDFT), $c[n]$, is defined as follows,

$$c[n] = \sum_{k=0}^{k=2N-1} a[k]\, e^{\frac{2\pi nk}{2N}} \equiv 2N \times IDFT a[k], \tag{2.14}$$

where N is the IFFT length. The real part of the output signal, $s[n]$, can be obtained as follows,

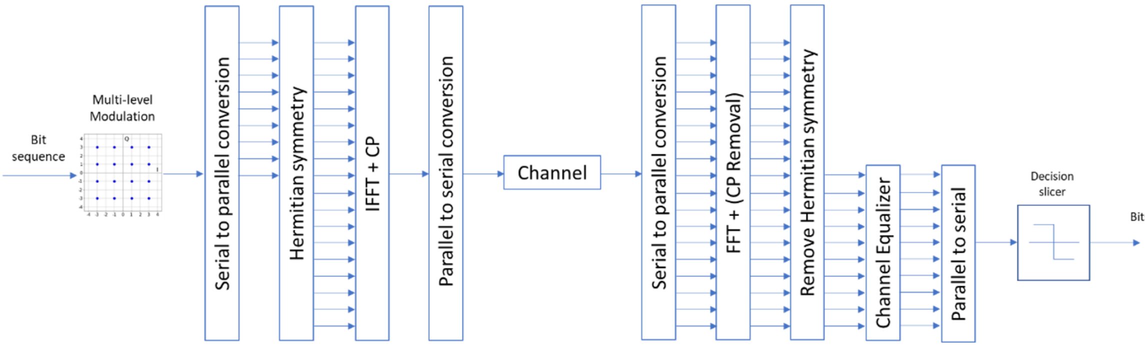


Fig. 2.19: DMT modulated and de-modulator architecture.

$$s\left[n\right] = \frac{\left(c\left[n\right] + c\left[n\right]^{*}\right)}{2}, \tag{2.15}$$

where $c\left[n\right]^{*}$ is the complex conjugate of IDFT defined in Equation 2.14. From the Hermitian symmetry property, the complex conjugate of IDFT is obtained by flipping the conjugate of the signal in the frequency domain given as follows,

$$\text{IDFT } \{a[0], a[1], \ldots, a[N-1]\}^{*} = \text{IDFT } \left\{[a[0], a[N-1], \ldots, a[1]]^{*}\right\}, \tag{2.16}$$

$$\text{IDFT } \{a[k]\}^{*} = \text{IDFT } \left\{[a[-k]]^{*}\right\}. \tag{2.17}$$

Substituting Eq 2.17 into Eq 4.17 the result $s[n]$ is given as follows,

$$s\left[n\right] = \frac{2N \times \text{IDFT}\left\{a\left[k\right] + \ \left[a\left[-k\right]\right]^{*}\ \right\}}{2}, \tag{2.18}$$

and

$$s\left[n\right] = \ N \times \text{IDFT}\left\{a\left[k\right] + \ \left[a\left[-k\right]\right]^{*}\ \right\}. \tag{2.19}$$

Considering the above derivations, the architecture of a DMT-type transmission scheme is shown in Fig. 2.19. At the receiver, the conjugate part of the signal in the frequency domain is discarded. Afterwards, channel equalisation is performed to estimate the transmitted data symbols.

2.6.2 Equalisation of DMT Signal

The equalisation of the received DMT signal can be performed both in the time domain and frequency domain. The thesis will emphasize the frequency domain equalisation techniques. One of the basic techniques for equalisation in the frequency domain is zero-forcing (ZF) equalisation. In this technique, the transmitted symbols are estimated as a product of the received signal and the inverse frequency domain response of the channel. But for the estimation of frequency domain response, training or pilot symbols are transmitted which are compared with the pilots stored in the receiver.

If it is assumed that the transmitted signal in the frequency domain is $X[k]$ and $H[k]$ is the channel response in the frequency domain, then the received symbol $Y[k]$ is given below [39] [40],

$$Y[k] = X[k] H[k] + N[k]. \tag{2.20}$$

where k is the subcarrier index and $N[k]$ is the noise term at every subcarrier. Channel estimation at the receiver can be performed using reference pilot sequences $X[k_p]$ using the following relation,

$$\hat{H}[k_p] = \frac{Y[k_p]}{X[k_p]}, \tag{2.21}$$

where k_p is the index of the pilot subcarriers. The pilots are only transmitted over a few subcarriers which is less than the total number of subcarriers in the DMT multiplex. The subcarriers transmitting pilot symbols are also equally spaced over the time-frequency grid. The channel estimation is thus carried out only for the pilot subcarriers. So for the subcarriers transmitting only data, the channel estimation is carried out via interpolation of the pre-calculated channel estimates. Finally, the transmitted sequence $\hat{X}[k]$ is estimated by the following relation,

$$\hat{X}[k] = \frac{Y[k]}{\hat{H}[k]}, \tag{2.22}$$

where $\hat{H}[k]$ is the overall estimated channel frequency response after interpolation of the pilot symbols frequency response $\hat{H}[k_p]$. After equalisation, the variance of an estimated symbol in Equation 2.22 is calculated using the properties of variance for a random variable multiplied by a constant [41] and given as follows,

$$Var\left(\hat{X}[k]\right) = Y[k]^2 \ \mathrm{Var}\left(\frac{1}{\hat{H}[k]}\right). \tag{2.23}$$

Substituting Equation 2.20 into Equation 2.21, results in the following channel estimate at pilot symbol subcarrier frequencies,

$$\hat{H}[k_p] = \frac{X[k_p]H[k_p] + N[k_p]}{X[k_p]}, \tag{2.24}$$

resulting in

$$\hat{H}[k_p] = H[k_p] + \frac{N[k_p]}{X[k_p]}. \tag{2.25}$$

Let us consider two cases of channel estimation, in the first case the channel is estimated by transmitting only symbol 'A' as pilots in Fig. 2.20, and in the second case, the channel is estimated by transmitting only the symbol 'B' as pilots in Fig. 2.20. The pilot symbol 'A' has the least amplitude in the QAM constellation in Fig. 2.20, whereas the pilot symbol 'B' has the highest symbol in Fig. 2.20.

The utilisation of pilot symbol 'A' results in the higher variance in the channel estimate $\hat{H}[k_p]$ in Equation 2.25, as the noise term ${}^{N[k_p]}/_{X[k_p]}$ in Equation 2.25, will become significant in comparison to the term $H[k_p]$. A higher variance in the channel estimate, $\hat{H}[k]$, will also result in a higher variance of the term, $({}^{1}/_{\hat{H}[k]})$, in Equation 2.23. Due to the enhanced variance of the term $({}^{1}/_{\hat{H}[k]})$, the variance of the estimated symbol, $\hat{X}[k]$, in Equation 2.25, will become significantly dependent on the variance of the power of the received symbol, $Y[k]^2$. Due to this dependence, the variance of noise in the equalised quadrature amplitude modulation (QAM) constellation becomes significantly dependent on the amplitude of the transmitted QAM symbols as shown in Fig. 2.21.

On the other hand, if the channel estimation is carried out using pilot symbols with higher amplitudes, such as symbol 'B' in Fig. 2.20, the noise term, ${}^{N[k_p]}/_{X[k_p]}$, in Equation 2.25 is not significant, resulting in less variance in the channel estimate terms, $\hat{H}[k_p]$ and $\hat{H}[k]$, in comparison to the case when symbols 'A' is used as a pilot symbol. This will further result in a comparatively less dependence of the variance of the noise, in the equalised QAM constellation, on the transmitted QAM symbols as shown in Fig 2.22. However there is the dependence of the variance of noise on the amplitude of the transmitted symbols is still pronounced in Fig 2.22, which can result in high error vector magnitudes (EVMs). Note that the constellations Fig. 2.21 and Fig. 2.22 are generated after propagating a DMT signal through an arbitrary channel with a signal-to-noise ratio (SNR) of 30 dB and subsequent equalisation.

For the research in the thesis, it is assumed that the aggregation and de-aggregation in fronthaul is digital and the channel frequency response consists of the fibre optic channel, the electro-optical modulator, and optical and electrical amplifiers. It was assumed that

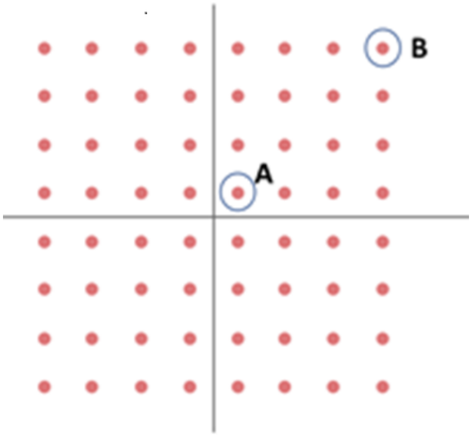


Fig. 2.20: 64 QAM constellation.

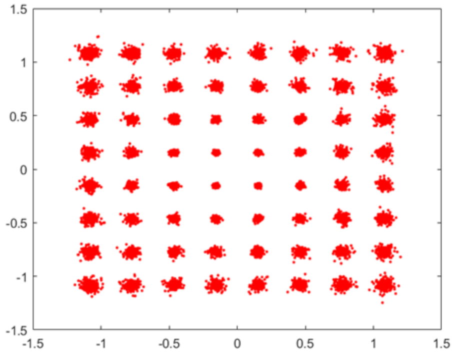


Fig. 2.21: 64 QAM constellation after ZF equalisation using symbol A pilot.

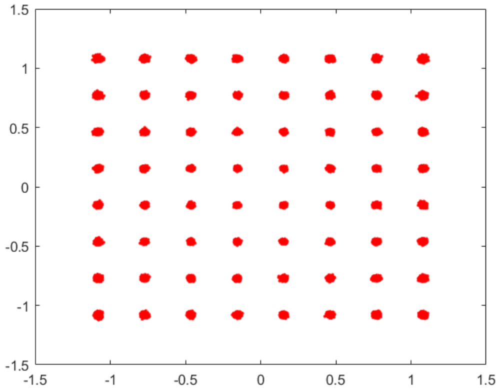


Fig. 2.22: 64 QAM constellation after ZF equalisation using symbol B pilot.

the frequency response of these components remains almost constant over time. Channel equalisation was performed by sending a preamble sequence comprising, random symbols from the QAM constellation symbol set. The preamble sequence was contiguous in time and frequency domain in the DMT multiplex. Channel estimation was then carried out for every DMT symbol using Equation 2.21 in the preamble. The final channel estimate was calculated by averaging the estimates from every symbol in the preamble and mathematically expressed as follows,

$$\overline{H[k]} = \frac{1}{N}\sum_{i=1}^{N}\hat{H}_i[k], \tag{2.26}$$

where $\overline{H[k]}$ is the averaged estimate of the channel over the subcarrier k and $\hat{H}_i[k]$ is the channel estimate at subcarrier k and i^{th} DMT symbol.

Equalisation via preambles will eliminate the requirement for continuous ZF forcing equalisation operation for all the transmitted symbols since preambles are transmitted over a finite number of symbols. This will improve the bandwidth efficiency and power consumption efficiency of the receiver digital signal processing (DSP) operation. Another benefit of using the preamble-averaging equalisation is the elimination of enhanced noise variance after equalisation, dependent on the transmitted symbol amplitude, among constellation points as shown in Fig. 2.21 and Fig. 2.22. A thorough comparison of the performance of ZF and the preamble-averaging equalisation is given in Chapter 5.

2.6.3 Analogue and Digital Fronthaul

There are two mechanisms for the transmission of data in 5G fronthaul i.e., via analogue, and digital aggregation and de-aggregation of data streams. The architecture of a fronthaul with analogue aggregation is shown in Fig. 2.23, [42]. For transmission of data streams via analogue aggregation in fronthaul, the data streams are first modulated via DMT modulation, then upconverted to different intermediate frequencies (IF). The DMT signals are combined and the multiplexed signal is used to modulate the carrier from a laser source using an electro-optical modulator such as a Mach-Zhender modulator (MZM) or Electro-absorption modulator (EAM). The optical signal propagates through the fibre optic channel. At the receiver, the optical signal is detected via a PIN diode. The composite detected signal is demultiplexed by bandpass filtering and then upconverted to radio frequency (RF) for transmission over the wireless channel.

In the case of digital aggregation in fronthaul shown in Fig. 2.24, different data streams are first aggregated in the digital domain and then modulated onto different subcarriers via DMT modulation. As in the case of analogue aggregation the DMT signal is then used to modulate a carrier signal from a laser source via MZM or EAM. At the receiver, the aggregate DMT signal is detected via a PIN diode. The detected signal is demodulated, and different

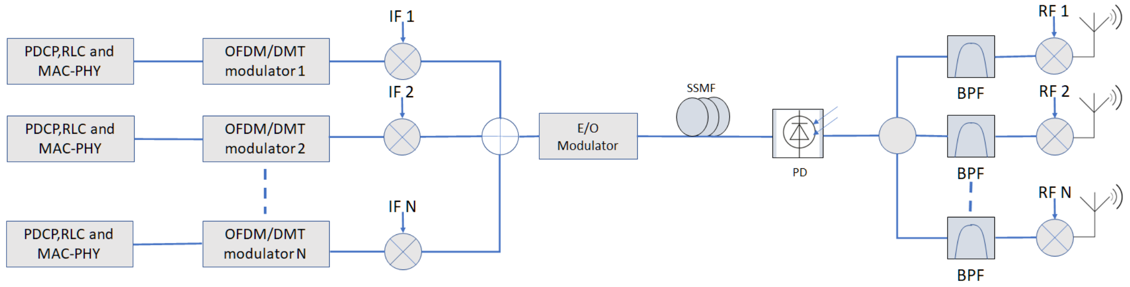


Fig. 2.23: Fronthaul with analogue aggregation/de-aggregation.

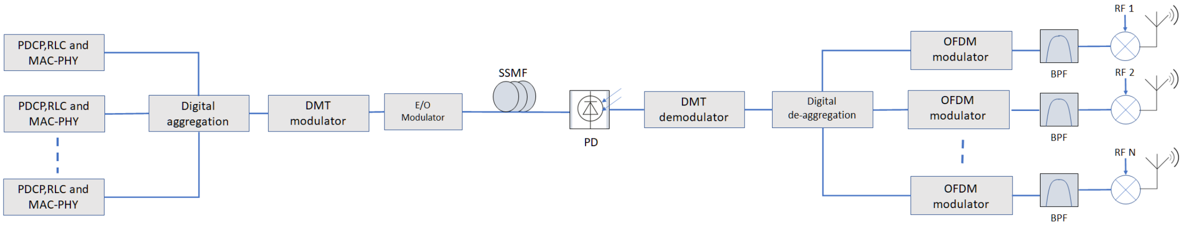
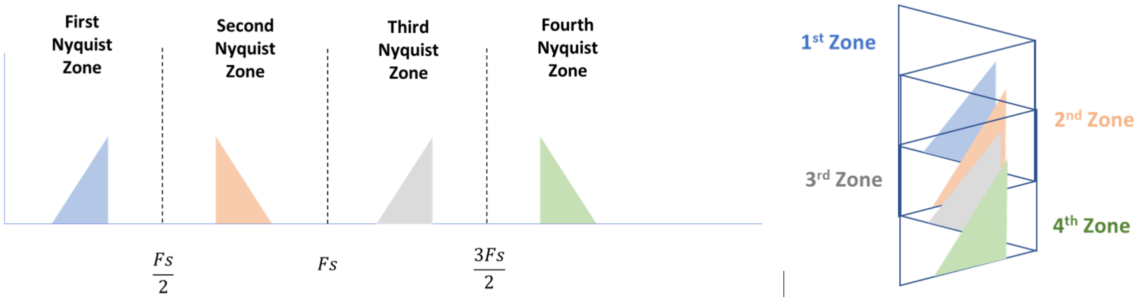


Fig. 2.24: Fronthaul with digital aggregation/de-aggregation.



(a) (b)

Fig. 2.25: Nyquist zone mapping (a) and down-sampling (b).

streams are de-aggregated in the digital domain. The streams are again modulated via OFDM for transmission over the wireless channel.

The cost of a fronthaul receiver with digital de-aggregation can also be reduced by using the Nyquist zone mapping and de-mapping technique. Nyquist zone de-mapping removes the requirement of high-speed A/D for a sampling of the entire spectrum. The received signal is sampled at a rate less than twice the highest frequency component in the signal. Higher frequency components then fold back onto baseband frequency between 0 Hz and $Fs/2$ Hz shown in Fig. 2.25(b). The spectra can be separated by bandpass filtering before Nyquist zone de-mapping. The technique for Nyquist zone de-mapping has been demonstrated in [43], [44] through experimentation for fronthaul in LTE. Each Nyquist zone can be modulated by a different channel or a MIMO layer in the downlink. The Nyquist zones can also have a mix of DMT and OFDM modulation demonstrated through simulation and experimentation in [45].

2.7 Introduction to Reinforcement Learning

Machine learning has introduced autonomous intelligence to different spheres of science and technology. The major impacts have appeared in the field of medical diagnostics, robotics, recommendation-based information technology systems such as social media, autonomous self-driving vehicles, drug discovery and telecommunication systems optimisations [46]. Overall machine learning techniques are categorized into three main categories [47]. These are supervised learning, unsupervised learning, and reinforcement learning.

In supervised learning a model is trained using data, that is assigned labels. The model learns to make predictions or classifications based on the patterns it learns from the labelled data. Examples of unsupervised learning are linear regression, support vector machines (SVM) and decision trees.

In unsupervised learning, a model is trained on unlabelled data. The model learns to label or classify data on its own, based on existing patterns or correlations among data samples. Examples of unsupervised learning algorithms include K-means clustering, hierarchical clustering and principal component analysis (PCA).

The third main category of machine learning is reinforcement learning (RL), which will be the focus of the thesis and its application in fronthaul for 5G networks. The mathematical foundations of RL are based on optimal control theory and dynamic programming. In optimal control theory and dynamic programming, a control problem is solved by breaking it down into simpler problems and then finding the best action at every stage [48],[49].

In comparison to previously mentioned machine learning algorithms, RL can emphasize or target the problem as a whole picture. It can generate its own sampling data set for training by interacting with the environment rather than relying on the data set required separately [50]. RL can adjust to dynamic environments, where the basic statistical properties of the data being generated change with time. Generally, RL is closer to artificial general intelligence (AGI) in comparison to other categories of machine learning as it can improve a long-term goal or reward while exploring various possibilities autonomously [51]. The main building blocks of reinforcement learning are given in Fig. 2.26 [52].

In RL an ‘agent’ observes the environment parameters and then performs actions to increase its reward. A reward is a user-defined function that depends on the observations, actions taken by the agent or both. The actions taken by the agent are with regular time

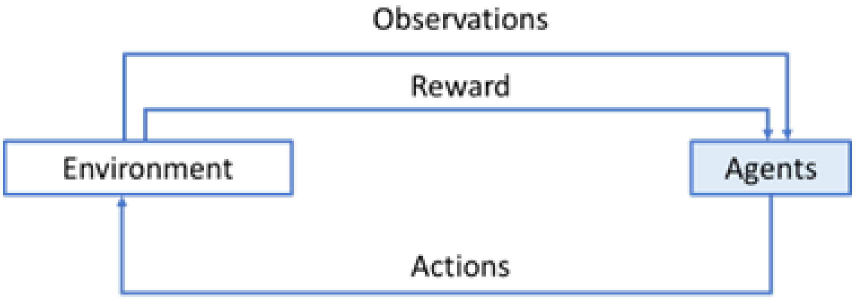


Fig. 2.26: Block diagram of reinforcement learning flow.

intervals. The collection of actions taken over a fixed length of time is termed an ‘episode’. The actions taken by the agent aim to increase the reward collected over multiple episodes.

The agent can perform actions either by using a model as an input or by learning the environment variables by interacting with it. The former RL is called model-based RL and the latter is termed model-free RL. The thesis will emphasize model-free RL as it is autonomous and can learn dynamic environments. Model-free RL is categorized into two classes. The first is policy optimisation-based RL and the second is Q-learning [47], [50].

The policy is the probability density of the actions that can be performed by an agent when in a particular state. In policy-based optimisation-based RL, the agent takes actions based on its policy, after observing the resulting states and rewards, and it updates its policy. The goal is to iteratively improve the policy so that the agent can make better decisions and achieve higher rewards [53]. One common approach in policy optimisation is to use gradient ascent optimisation methods. In this method, the parameters are updated based on the gradient of an objective, such as the expected cumulative reward. Examples of policy-based methods are actor-critic methods (A2C/A3C), proximal policy optimisation methods (PPO) and deep deterministic policy gradient (DDPG) [53].

In Q-learning-based methods, the agent attempts to optimise the value of the actions taken when in a state so that the overall reward accumulated increases. Q-learning is termed as off-policy-based learning since each update can use data collected at any point during the past training process, regardless of how the agent chose to explore the environment. The details of the Q-learning and its mathematical foundation will be discussed in the coming Section [53].

From the perspective of optimisation, conventional optimisation techniques such as integer linear programming (ILP) and mixed integer linear programming (MILP) require prior knowledge of the environment variables and constraints, whereas RL agent or agents

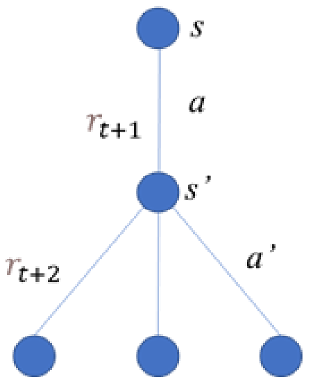


Fig. 2.27: MDP for RL.

observe the environment's variable values and then take actions to achieve a reward defined by the user. Therefore, agents do not require any prior knowledge of environment variable values.

2.7.1 Introduction to Bellman's Equation

If it is assumed that the RL paradigm is a Markov decision process (MDP), then out of many possible actions from the action set, the agent takes action (a) from a given state of the (s) of the agent within the environment. The action taken by the agent is such that it maximizes its cumulative reward over the subsequent actions taken from subsequent states. This is shown in the Markov chain in Fig. 2.27, where an agent takes an action (a) from a state (s) and reaches another state (s'). In transitioning from the state (s) to state (s') it earns a reward (r).

As mentioned, the goal of the agent is to maximize the reward over the actions taken by it from different states. The accumulated rewards collected by an agent are defined as follows,

$$G_t = r_{t+1} + \gamma\, r_{t+2} + \gamma^2 r_{t+3} + \dots. \tag{2.27}$$

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{2.28}$$

where, t is the time of the current state, and, γ^k is the discount factor that emphasises rewards from immediate actions rather than future actions. The value of the action is termed as Q-value, of an agent from a given state (s) and takes an action (a) is defined as follows,

$$q_\pi(s,a) = E_\pi\left(G_t | S_t = s, A_t = a\right), \tag{2.29}$$

Algorithm 2 Q-table learning algorithm

```
Initialize $Q$ matrix
Set the number of episodes $J$
Set the maximum number of steps per episode $S$
for $j = 1,2,\ldots,J$ do
    $k \leftarrow 1$
    Select a random initial state $s$
    while The goal state is not reached and $k \leqslant S$ do
        Select a valid action $a_k$ at random
        Record the resulting state $s_{k+1}$ and corresponding reward
        Update the Q-values of the state-action pair
            $q_*(s_k,a_k) \leftarrow r_{t+1} + \gamma \max_{a_k} q_*(s_{k+1},a_{k+1})$
        $k \leftarrow k+1$
    end while
end for
```

where, E_π is the expectation operator, $q_\pi(s,a)$ is defined as the probability density/mass function defining the probability of the action (a) taken by the agent from the state (s), S_t is the state set and A_t is the action set. The optimal or the maximum Q-value of the state action pair is defined as follows from the bellman's optimality equation [54], [55],

$$q_*(s,a) = E_\pi \left(r_{t+1} + \gamma \max_{a\prime} q_*(s\prime,a\prime) \right). \tag{2.30}$$

The expectation operator is performed over the next state (s') probabilities from the state(s).

2.7.2 Q-Learning and Deep-Q Learning

In Q-learning an agent is trained so that it can learn to take actions having the highest Q-values from a state. This process continues until the agent reaches the desired state. This process of learning can be carried out iteratively where the agent learns and updates the Q-values of the state-action pair in each iteration. The algorithm for learning the Q-values is given in Algorithm, 2 [56], [57].

After the learning process, the agent is deployed. The deployed agent takes the action from a state having the highest Q-values to reach its destination state. From a computation standpoint, the Q-values are stored in a table termed a 'Q-table'. The input to the table is the state and action pair, and the output is the corresponding Q-values as shown in Fig. 2.28.

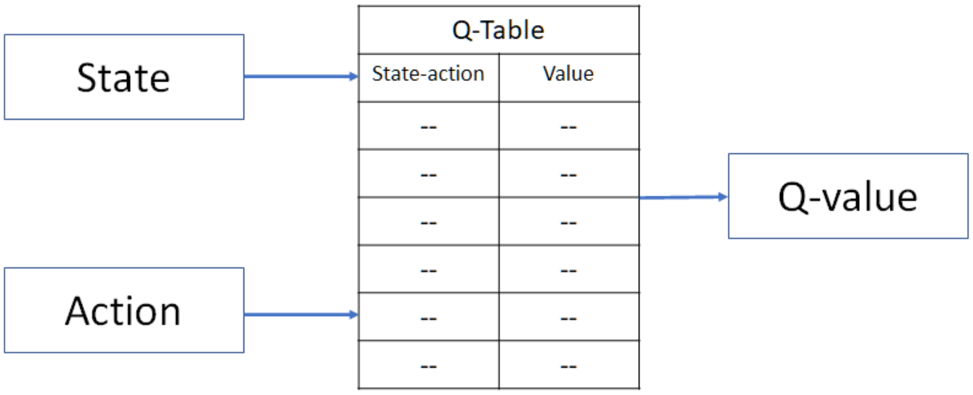


Fig. 2.28: Illustration of Q-table based RL.

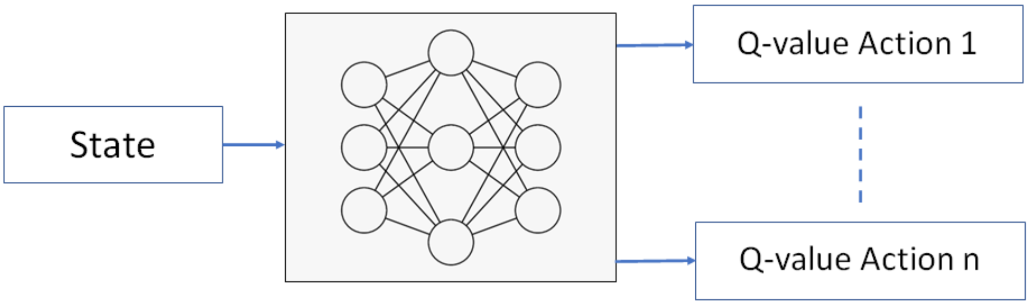


Fig. 2.29: Illustration of Q-table based RL

If the state and action space is large and stored in a look-up table (LUT), the storage will require large memory resources and will result in large delays in producing the Q-values.

To overcome the constraints of memory resources, deep-Q learning was proposed [58]. In deep-Q learning the states and actions are related to Q-values via deep neural networks (DNNs), illustrated in Fig. 2.29. The input to the DNN in Fig. 2.29 is the state, where the output of the DNN is the Q-values for the actions related to the state. Similar to any DNN, the labels in this case are the optimal Q-values calculated from the bellman's equation. The process of training the deep-Q network boils down to minimizing the temporal difference or loss function between the optimal Q values from Bellman's equation and current Q-values. The loss function is given as follows,

$$L_i\left(\theta_i\right) = \mathbb{E}_{s,a,r,s\prime}\left[\left(y_i - Q(s,a;\theta_i)\right)^2\right], \tag{2.31}$$

where the optimal Q-values from Bellman's equation are given as follows,

$$y_i = r_{t+1} + \gamma \max_{a\prime} q_*(s', a'; \theta_{i-1}). \tag{2.32}$$

For the calculation of the loss function, $L_i(\theta_i)$, the current state (s) is passed to the deep-Q network to calculate the current Q-value, $Q(s,a;\theta_i)$ as shown in Fig. 2.29, and in the second step, the next state, $s\prime$ is passed to calculate the optimal Q-values $y_{,i}$. This will result in the calculation of the optimal Q-values being calculated using the same weights as that of the current Q-values. This will cause an unstable calculation of loss function and training of the deep-Q network. To overcome this shortcoming, the weights of the deep-Q network are passed to a second network called the target network to calculate the target Q-values, y_i. However, the weights of the target network are updated fewer times than the Q-network and the update frequency parameter is the number of steps between updates. The steps for deep-Q learning are given in Algorithm 3 [58], [59], [60].

The Algorithm 3, in the first step, requires initialisation of replay memory for storing the state, action and rewards in an experience buffer. The Algorimth 3 also requires the initialisation of the weights of the Deep Q-network and the target-Deep Q-network. The Algorithm 3 computes the loss function given in Equation 2.32 iteratively, all while storing the state, the action, the rewards and the next states within an experience buffer. The experience buffer is used to compute the loss function averaged over the sampled data from the experience buffer. The Algorithm 3 continues till the pre-defined number of steps of training are not reached.

Algorithm 3 Deep Q-learning algorithm

```
Initialize replay memory $D$ capacity to $N$
Initialize the Q-network with random weights $\theta$
Clone the Q-network with random weights $\theta'$ for the target network.
Set the number of episodes $J$
Set the maximum number of steps per episode $S$
for $j = 1,2,\ldots,J$ do
    $k \leftarrow 1$
    Select a random initial State $s$.
    while The goal state is not reached and $k \leqslant S$ do
        For the current observation $s$, select a random action $A$ with probability $\varepsilon$
        Otherwise, select the action for which the critic value function is the greatest
            $$A = \max_{a'} q_*(s', a').$$
        Observe the reward $r$ and next observation $s'$.
        Store the experience $(s, a, r, s\prime)$ in the experience buffer $D$
        Sample a random mini-batch of $M < N$ experiences $(s_i, a_i, r_i, s_i\prime)$ from the experience buffer
        Gradient descent updates weights in the policy network to minimize loss over samples from the minibatch given as follows,
            $$L_i(\theta_i) = \frac{1}{2M} \sum_{i=1}^{M} \left[ (y_i - Q(s, a; \theta_i))^2 \right]$$
        where,
            $$y_i = r_{t+1} + \gamma \max_{a'} q_*(s', a'; \theta_{i-1}).$$
        After $x < S$ time steps, weights in the target network are updated to the weights in the policy network
        $k \leftarrow k + 1$
    end while
end for
```

The overall process of training the deep Q-network is also given in Fig. 2.30. The replay memory contains the tuples for states, corresponding actions, rewards, and next states. In the first step, the deep Q-network extracts the state action pair from the tuple in the experience buffer for the calculation of the current Q-values. The target Q-network extracts the next state and rewards for the calculation of the optimal Q-values. Using the computed current Q-values and target Q-values the loss function is computed by the loss function block in Fig. 2.30, based on which the weights of the Q-network are computed. The Q-network in

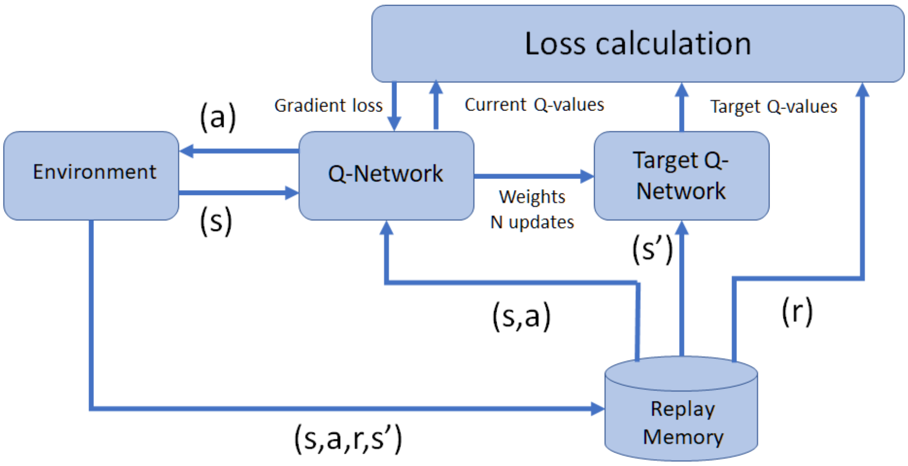


Fig. 2.30: Deep-Q network training process.

Fig. 2.30 after computations of the Q values, outputs the actions (a) which are fed into the environment block in Fig. 2.30.

2.7.3 Reinforcement Learning for 5G Networks

There have been several studies that have researched the utilisation of RL for the optimisation of fronthaul in 5G networks. RL has been demonstrated for power control for interference management among users in ultra-dense networks where the network controller can optimally decide to find a balance between the transmit powers of users and the user throughputs. Deep reinforcement learning (DRL) has been applied to learn the best joint computing, caching, and communication design problems to enhance the performance of vehicular networks. DRL has also been used for the optimisation of slice allocation to different services based on the queue lengths and available resources in different slices [61].

In [62] optimal joint route selection in fronthaul and placement of BBU within the central office (CO) has been demonstrated using DRL. In this study, the latency of a path in fronthaul is only modelled in terms of the propagation delay of a path in fronthaul.

In [63], resource allocation is performed by using DRL within the constraint of transmission delay of packets in an ultra-reliable low latency (uRLLC) network. Although the transmission delays are calculated based on the available capacity of the channel between the IoT nodes, gNode-B and the packet lengths, the dependence of delays on queuing or buffering delays in aggregation nodes in fronthaul is not considered.

In [64], optimal route selection in SDNs has been demonstrated using integer linear programming (ILP) and DRL, under the constraints of latency. However, latency is only modelled based on link utilisation or load on a link.

In [65] and [66] resource allocation in Fog RAN and RAN in 5G networks is performed using DRL considering the latency constraints of different service requests. Based on the latency constraints of the service requests the caching and signal processing resources are allocated on the edge or in the cloud. The latency of a path in fronthaul is calculated based on the load on a path using the available Shannon's capacity of a path and the desired throughput of a service. Yet there is no consideration of the queuing delays of a path in fronthaul connecting different IoT nodes to the cloud. Within the studies, no mechanism for handling the attachment of UEs with RUs having a low latency path in fronthaul is considered or discussed.

The aforementioned studies omitted any mechanism for handling the allocation of users to RUs under the constraints of latency of their paths in the fronthaul, as a means of maximisation of their achievable bit rate. The models for latency in these studies were based either on propagation delay, serialisation delay, mean queuing delay or a priori end-to-end delay measurements, and thus did not consider the combined effect of all possible sources of delays in their models. They further omitted any consideration of percentile delays due to the buffering of data at the destination nodes of fronthaul paths. None of the aforementioned studies considered a resource allocation scheme for fronthaul, that is enabled by FlexE for efficient bandwidth utilisation or its influence on the latency constraint.

Considering the shortcoming in the studies the objective of the research in this thesis is to optimise the allocation of resources/slots in a fronthaul network using FlexE technology. This optimisation also involves managing user allocations based on the latency of the path in the fronthaul network, where the latency of the path in fronthaul was benchmarked in terms of percentile delays, incorporating serialisation and queuing delays in aggregation nodes.

References

[1] ‘White Paper: 5G Network Architecture - A High-Level Perspective - Industry insight in Huawei’, huawei. Accessed: May 26, 2023. [Online]. Available: https://www.huawei.com/en/huaweitech/industry-insights/outlook/mobile-broadband/insights-reports/5g-network-architecture

[2] M. Shafi et al., ‘5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice’, IEEE Journal on Selected Areas in Communications, vol. 35, no. 6, pp. 1201–1221, Jun. 2017, doi: 10.1109/JSAC.2017.2692307.

[3] C.-X. Wang et al., ‘Cellular architecture and key technologies for 5G wireless communication networks’, IEEE Communications Magazine, vol. 52, no. 2, pp. 122–130, Feb. 2014, doi: 10.1109/MCOM.2014.6736752.

[4] J. Duplicy et al., ‘MU-MIMO in LTE Systems’, J Wireless Com Network, vol. 2011, no. 1, Art. no. 1, Dec. 2011, doi: 10.1155/2011/496763.

[5] M. A. Habibi, M. Nasimi, B. Han, and H. D. Schotten, ‘A Comprehensive Survey of RAN Architectures Toward 5G Mobile Communication System’, IEEE Access, vol. 7, pp. 70371–70421, 2019, doi: 10.1109/ACCESS.2019.2919657.

[6] Y. Stein, ‘Everything you need to know about 5G xHaul’. Nov. 02, 2020. Accessed: Aug. 07, 2023. [Online]. Available: https://www.rad.com/blog/everything-you-need-know-about-5g-xhaul.

[7] OIF, ‘Flex Ethernet 2.2 Implementation Agreement’, p. 51, Oct. 2021.

[8] A. de la Oliva, J. A. Hernandez, D. Larrabeiti, and A. Azcorra, ‘An overview of the CPRI specification and its application to C-RAN-based LTE scenarios’, IEEE Communications Magazine, vol. 54, no. 2, pp. 152–159, Feb. 2016, doi: 10.1109/MCOM.2016.7402275.

[9] N. J. Gomes et al., “Boosting 5G Through Ethernet: How Evolved Fronthaul Can Take Next-Generation Mobile to the Next Level,” IEEE Vehicular Technology Magazine, vol. 13, no. 1, pp. 74–84, Mar. 2018

[10] 3GPP, “3GPP TR 38.801 V14.0.0 (2017-03): Study on new radio access technology: Radio access architecture and interfaces.” 3GPP, 2017.

[11] Richard MacKenzie, ‘NGMN Overview on 5G RAN Functional Decomposition’, NGMN. Accessed: Oct. 30, 2019. [Online]. Available: https://www.ngmn.org/publications/ngmn-overview-on-5g-ran-functional-decomposition.html

[12] L. M. P. Larsen, A. Checko, and H. L. Christiansen, ‘A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks’, IEEE Communications Surveys Tutorials, vol. 21, no. 1, pp. 146–172, First quarter 2019, doi: 10.1109/COMST.2018.2868805.

[13] U. Dötsch, M. Doll, H.-P. Mayer, F. Schaich, J. Segel, and P. Sehier, ‘Quantitative analysis of split base station processing and determination of advantageous architectures for LTE’, Bell Labs Technical Journal, vol. 18, no. 1, pp. 105–128, Jun. 2013, doi: 10.1002/bltj.21595.

[14] J. A. Hernández, G. Otero, D. Larrabeiti and Ó. G. de Dios, "Dimensioning Flex Ethernet Groups for the transport of 5G NR fronthaul traffic in C-RAN scenarios," 2021 International Conference on Optical Network Design and Modeling (ONDM), Gothenburg, Sweden, 2021, pp. 1-3, doi: 10.23919/ONDM51796.2021.9492417.

[15] N. Huin et al., "Hard-isolation for Network Slicing," IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 2019, pp. 955-956, doi: 10.1109/INFCOMW.2019.8845282.

[16] ‘What is FlexEthernet and why is it so important? - Ciena’. http://www.ciena.com/insights/articles/What-is-FlexEthernet-and-Why-is-it-so-Important.html(accessed Dec. 16, 2019).

[17] Microchip Technology Inc., ‘Benefits of Flex Ethernet in Cloud and Telecom Service Provider Networks’. Accessed: Jun. 03, 2023. [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/DS-00003413B.pdf

[18] ‘Flex Ethernet: breaking the chains of physical bandwidth’, Optical Connections News, Oct. 02, 2018. https://opticalconnectionsnews.com/2018/10/flex-ethernet-breaking-the-chains-of-physical-bandwidth/ (accessed Jun. 03, 2023).

[19] Huawei Ltd., ‘FlexE Technical White Paper (2018)’. Accessed: Jun. 03, 2023. [Online]. Available: https://www.ycict.net/id/wp-content/uploads/sites/5/2020/01/FlexE-Technical-White-Paper-1.pdf

[20] International Telecommunication Union-Telecommunication Standardisation Sector (ITU-T), "Characteristics of equipment functional blocks supporting Ethernet physical layer and Flex Ethernet interfaces," Recommendation ITU-T G.8023, pp. 1-56, June 2018.

[21] Optical Internetworking Forum (OIF), "Flex Ethernet Implementation Agreement 2.0," pp. 1-51, 22 June 2018.

[22] R. Vilalta, R. Martínez, R. Casellas, R. Muñoz, Y. Lee, L. Fei, P. Tang and V. López, "Network Slicing Using Dynamic Flex Ethernet over Transport Networks," in European Conference on Optical Communication (ECOC), Gothenburg, Sweden, 2017.

[23] ‘5G Programmable Infrastructure Converging disaggregated network and compute resources’, D 4.2, Nov. 2018. [Online]. Available: https://www.5g-picture-project.eu/download/5g-picture_D4.2.pdf

[24] S. Zhang, Q. Zhong, M. Zha and T. Zuo, "Hybrid Multiplexing over FlexE Group," in OptoElectronics and Communications Conference (OECC 2018) Technical Digest, Jeju, Korea, 2018.

[25] ‘Introductin Xhaul: Rewriting the playbook for transport networking in the RAN in 5G.’ Accessed: Jan. 09, 2023. [Online]. Available: https://www.acgcc.com/reports/introducing-xhaul-rewriting-the-playbook-for-trans/

[26] D. Bertsekas and R. Gallager, Data Networks: Second Edition. Athena Scientific, 2021.

[27] ‘3GPP Release 15 Overview - IEEE Spectrum’. Accessed: Apr. 01, 2024. [Online]. Available: https://spectrum.ieee.org/3gpp-release-15-overview

[28] G. O. Pérez, J. A. Hernández, and D. Larrabeiti, “Fronthaul Network Modeling and Dimensioning Meeting Ultra-Low Latency Requirements for 5G,” J. Opt. Commun. Netw., vol. 10, no. 6, p. 573, Jun. 2018, doi: 10.1364/JOCN.10.000573.

[29] J. F. C. Kingman, “The single server queue in heavy traffic,” Math. Proc. Cambridge Philos. Soc., vol. 57, no. 4, pp. 902–904, 1961.

[30] W. Dai and J.-Q. Hu, ‘Correlated queues with service times depending on inter-arrival times’, Queueing Syst, vol. 100, no. 1, pp. 41–60, Feb. 2022, doi: 10.1007/s11134-021-09718-7.

[31] Lindley, D.V.: ‘The theory of queues with a single server’ Mathematical Proceedings of the Cambridge Philosophical Society, 1952, 48, (2), pp. 277–289.

[32] P. Schulz, ‘Queueing-Theoretic End-to-End Latency Modeling of Future Wireless Networks’, PhD thesis, Technische Universität Dresden, Dresden, 2020. [Online]. Available: https://d-nb.info/1227833598/34

[33] S. C. J. Lee, ‘Discrete multitone modulation for short-range optical communications’, PhD Thesis, Technische Universiteit Eindhoven, Eindhoven, 2009. doi: 10.6100/IR656509.

[34] A. Goldsmith, Ed., ‘Multicarrier Modulation’, in Wireless Communications, Cambridge: Cambridge University Press, 2005, pp. 374–402. doi: 10.1017/CBO9780511841224.013.

[35] K. Zhang, Q. Zhuge, H. Xin, W. Hu, and D. V. Plant, ‘Performance comparison of DML, EML and MZM in dispersion-unmanaged short reach transmissions with digital signal processing’, Opt. Express, OE, vol. 26, no. 26, pp. 34288–34304, Dec. 2018, doi: 10.1364/OE.26.034288.

[36] H. Griesser, N. Eiselt, A. Dochhan, M. Eiselt, and J.-P. Elbers, “PAM-4 and DMT for Inter-Data Center Connections and 5G Fronthaul Applications,” in 2018 20th International Conference on Transparent Optical Networks (ICTON), Bucharest, Jul. 2018, pp. 1–1, doi: 10.1109/ICTON.2018.8473944.

[37] G. N. Liu, L. Zhang, T. Zuo, and Q. Zhang, “IM/DD Transmission Techniques for Emerging 5G Fronthaul, DCI, and Metro Applications,” Journal of Lightwave Technology, vol. 36, no. 2, pp. 560–567, Jan. 2018, doi: 10.1109/JLT.2018.2793313.

[38] M. Jinno, "Elastic Optical Networking: Roles and Benefits in Beyond 100-Gb/s Era," in Journal of Lightwave Technology, vol. 35, no. 5, pp. 1116-1124, 1 March1, 2017, doi: 10.1109/JLT.2016.2642480.

[39] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, ‘Channel Estimation’, in MIMO-OFDM Wireless Communications with MATLAB, 1st edition., Singapore; Hoboken, NJ: Wiley-IEEE Press, 2010.

[40] ‘Communication Technology | ShareTechnote’. https://www.sharetechnote.com/html/Communication_ChannelModel_MMSE.html (accessed Jun. 20, 2023).

[41] L. Wasserman, ‘Expectation’, in All of Statistics: A Concise Course in Statistical Inference, L. Wasserman, Ed., New York, NY: Springer, 2004, pp. 48–61. doi: 10.1007/978-0-387-21736-9-3.

[42] M. Hinrichs, L. F. del Rosal, C. Kottke, and V. Jungnickel, ‘Analog vs. next-generation digital fronthaul: How to minimize optical bandwidth utilisation’, in 2017 International Conference on Optical Network Design and Modeling (ONDM), May 2017, pp. 1–6. doi: 10.23919/ONDM.2017.7958539.

[43] L. Cheng, X. Liu, N. Chard, F. Effenberger, and G.-K. Chang, “Experimental Demonstration of Sub-Nyquist Sampling for Bandwidth- and Hardware-Efficient Mobile Fronthaul Supporting 128×128 MIMO with 100-MHz OFDM Signals,” in Optical Fiber Communication Conference, Anaheim, California, 2016, p. W3C.3, doi: 10.1364/OFC.2016.W3C.3.

[44] X. Liu, H. Zeng, N. Chand, and F. Effenberger, “Efficient Mobile Fronthaul via DSP-Based Channel Aggregation,” J. Lightwave Technol., vol. 34, no. 6, pp. 1556–1564, Mar. 2016, doi: 10.1109/JLT.2015.2508451.

[45] S. Noor, P. Assimakopoulos, and N. J. Gomes, “A Flexible Subcarrier Multiplexing System With Analog Transport and Digital Processing for 5G (and Beyond) Fronthaul,” Journal of Lightwave Technology, vol. 37, no. 14, pp. 3689–3700, Jul. 2019, doi: 10.1109/JLT.2019.2918215.

[46] P. Johri, J. K. Verma, and S. Paul, Applications of Machine Learning. Springer Nature, 2020.

[47] S. Ray, ‘A Quick Review of Machine Learning Algorithms’, in 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Feb. 2019, pp. 35–39. doi: 10.1109/COMITCon.2019.8862451.

[48] O. Hernández-Lerma, L. R. Laura-Guarachi, S. Mendoza-Palacios, and D. González-Sánchez, An Introduction to Optimal Control Theory: The Dynamic Programming Approach, vol. 76. in Texts in Applied Mathematics, vol. 76. Cham: Springer International Publishing, 2023. doi: 10.1007/978-3-031-21139-3.

[49] D. Bertsekas, Dynamic Programming and Optimal Control: Volume II; Approximate Dynamic Programming. Athena Scientific, 2012.

[50] 'What is Reinforcement Learning? – Overview of How it Works | Synopsys'. https://www.synopsys.com/ai/what-is-reinforcement-learning.html (accessed Apr. 19, 2023).

[51] I. H. Sarker, 'Machine Learning: Algorithms, Real-World Applications and Research Directions', SN COMPUT. SCI., vol. 2, no. 3, p. 160, Mar. 2021, doi: 10.1007/s42979-021-00592-x.

[52] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. in Adaptive Computation and Machine Learning series. Cambridge, MA, USA: A Bradford Book, 2018.

[53] A. Plaat, 'Policy-Based Reinforcement Learning', in Deep Reinforcement Learning, A. Plaat, Ed., Singapore: Springer Nature, 2022, pp. 101–133. doi: 10.1007/978-981-19-0638-1-4.

[54] S. J. Bradtke and M. O. Duff, 'Reinforcement learning methods for continuous-time Markov decision problems', in Proceedings of the 7th International Conference on Neural Information Processing Systems, in NIPS'94. Cambridge, MA, USA: MIT Press, Jan. 1994, pp. 393–400.

[55] H. Dave, 'Understanding the Bellman Optimality Equation in Reinforcement Learning', Analytics Vidhya, Feb. 13, 2021. https://www.analyticsvidhya.com/blog/2021/02/understanding-the-bellman-optimality-equation-in-reinforcement-learning/ (accessed Apr. 19, 2023).

[56] C. J. C. H. Watkins and P. Dayan, 'Technical Note: Q-Learning', Machine Learning, vol. 8, no. 3, pp. 279–292, May 1992, doi: 10.1023/A:1022676722315.

[57] 'Q-Learning algorithm' https://rezaborhani.github.io/mlr/blog_posts/Reinforcement_Learning/Q_learning.html (accessed Apr. 19, 2023).

[58] V. Mnih et al., 'Playing Atari with Deep Reinforcement Learning'. arXiv, Dec. 19, 2013. doi: 10.48550/arXiv.1312.5602.

[59] 'Deep Q-Network (DQN) Agents - MATLAB & Simulink'. https://www.mathworks.com/help/reinforcement-learning/ug/dqn-agents.html (accessed Apr. 19, 2023).

[60] 'Training a Deep Q-Network with Fixed Q-targets - Reinforcement Learning'. https://deeplizard.com/learn/video/xVkPh9E9GfE (accessed Apr. 19, 2023).

[61] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L.-C. Wang, ‘Deep Reinforcement Learning for Mobile 5G and Beyond: Fundamentals, Applications, and Challenges’, IEEE Vehicular Technology Magazine, vol. 14, no. 2, pp. 44–52, Jun. 2019, doi: 10.1109/MVT.2019.2903655.

[62] Z. Gao, J. Zhang, S. Yan, Y. Xiao, D. Simeonidou, and Y. Ji, “Deep Reinforcement Learning for BBU Placement and Routing in C-RAN,” in 2019 Optical Fiber Communications Conference and Exhibition (OFC), Mar. 2019, pp. 1–3.

[63] M. Elsayed and M. Erol-Kantarci, “Reinforcement Learning-Based Joint Power and Resource Allocation for URLLC in 5G,” in 2019 IEEE Global Communications Conference (GLOBECOM), Dec. 2019, pp. 1–6. doi: 10.1109/GLOBECOM38437.2019.9014032.

[64] E. H. Bouzidi, A. Outtagarts, and R. Langar, “Deep Reinforcement Learning Application for Network Latency Management in Software Defined Networks,” in 2019 IEEE Global Communications Conference (GLOBECOM), Dec. 2019, pp. 1–6. doi: 10.1109/GLOBECOM38437.2019.9013221.

[65] A. Nassar and Y. Yilmaz, “Reinforcement Learning for Adaptive Resource Allocation in Fog RAN for IoT with Heterogeneous Latency Requirements,” IEEE Access, vol. 7, pp. 128014–128025, 2019, doi: 10.1109/ACCESS.2019.2939735.

[66] S. Sritharan, H. Weligampola, and H. Gacanin, “A Study on Deep Learning for Latency Constraint Applications in Beyond 5G Wireless Systems,” IEEE Access, vol. 8, pp. 218037–218061, 2020, doi: 10.1109/ACCESS.2020.3040133.

Chapter 3

Modelling of End-to-End Latency in Fronthaul

3.1 Introduction

In fronthaul for beyond-5G mobile networks, different functional splits will impose either strict or relaxed latency requirements as mentioned in Section 2.3.4. Different services in 5G and beyond networks will also have requirements of end-to-end latency in the network. The services will also have different reliability standards in terms of the percentage of packets transported through the network or dropped. For example, according to 3GPP Release 15, URLLC targets a target latency of 1 millisecond and a reliability level of 99.999% [1], [2]. So, end-to-end delays in networks are modelled in terms of percentage delays [3].

Recently there have been several studies to model the end-to-end delays of 5G Xhaul networks., using M/G/1 [4] and G/G/1 [3] queuing models. Latency calculations from these models were used as a constraint for the optimal selection of functional split factors [4] and optimal route selection in Xhaul networks [3], [4]. Variations in packet inter-arrival times will tend to smooth out as packets propagate through higher-level Ethernet-based aggregation nodes in the Xhaul network, altering the statistics of the inter-arrival times [6]. In the current 5G and beyond-5G mobile networks, the data packet lengths may not follow a Poisson distribution as in classical queuing theory models. This is so since the overall traffic will comprise of a mix different traffic from different slices of 5G networks such as eMBB, uRLLC and mMTC. In uRLLC packets can range from 32 bytes to 256 bytes, while eMBB packets can range from 128 bytes to 1500 bytes [7].

A generalised queuing model based on G/G/1 was used for modelling up to 99th percentile delays in [3] which assumed a generalised distribution of inter-arrival times of packets, as well as their service times. The analysis was carried out for two levels of Ethernet-based aggregation in Xhaul and fixed packet lengths in eCPRI. To accurately model higher percentiles, authors in [6] have used the N*D/D/1 queuing model with the assumption of fixed lengths of packets and only two levels of Ethernet-based aggregation. The N*D/D/1 was able to model the extreme percentile delays up to 99.9999999%, but this was demonstrated for only two levels of Ethernet-based aggregation. In the thesis, the end-to-end delay is modelled for a packet network with three levels of Ethernet aggregation and different input link rates. While this is similar to what might be expected in a fronthaul, the results are generally applicable to Ethernet aggregation networks. The model predicts the end-to-end delays in terms of percentile delays with the assumption that the packets have random lengths. The distributions of inter-arrival times of packets at different levels of Ethernet-based aggregations were also modelled as they were one of the input parameters to the model.

In this Chapter, a model for modelling the end-to-end latency in fronthaul is presented using both G/G/1 and Lindley's recursion-based models. The modelling was aided by the simulation of a generalised packet network similar to a Fronthaul network in 5G networks, in a network simulator (NetSim®). The Chapter also highlights the statistical properties of waiting times in queues of packets in Ethernet switches, in terms of mean, standard deviation and probability density functions (PDFs). The model presented in this Chapter predicts the end-to-latency in fronthaul in terms of percentile delays. The predicted percentile delays are also benchmarked with measured percentile delays from simulations.

In Section 3.2, the reference packet network used for simulations, to model the end-to-end latency in fronthaul is discussed. In Section 3.3 and Section 3.4, the results for modelling of mean waiting times of a path in fronthaul network using the G/G/1 and Lindley's recursion-based models are discussed. In Section 3.5, the results for the statistical modelling of the end-to-end delays and percentile delays are given. The process for the modelling of end-to-end percentile delays is also summarized in the form of an algorithm and a flow chart in Section 3.5. In Section 3.6, the results for the deployment of percentile delays in different network configurations are presented via benchmarking with simulation results. In Section 3.7, the conclusions for the modelling of end-to-end percentile delays are given.

3.2 Simulation Parameters for the Modelling of Latency in Fronthaul

The packet network architecture, similar to a fronthaul network in Fig. 2.15, for the development and testing of the end-to-end delay model, is shown in Fig. 3.1. It was assumed that the uplink path comprises an RU and co-located CU/DU. The packetized user and mobile data in an RU in fronthaul was modelled by the source Ethernet nodes shown in Fig. 3.1, which generate packets with random lengths. The traffic from the four source Ethernet nodes in Fig. 3.1 is further aggregated by an Ethernet switch. This switch is termed as a level-1 Ethernet switch. The traffic from three level-1 Ethernet switches is further aggregated by a single leve1-2 Ethernet switch. The aggregated traffic is transported to a destination node in Fig. 3.1, which might be assumed to be the NIC of the CU/DU in the Ethernet fronthaul network shown in Fig. 2.16. It is assumed that the Ethernet switches operate in first-in-first-out (FIFO) and the Ethernet switches are connected by fibre-optic links, which add propagation delays to the end-to-end delay.

A path in the network is defined as the end-to-end link between a source Ethernet node and the destination node, as shown by the blue arrow in Fig. 3.1. The different components of the total end-to-end delay in a path are also shown in Fig. 3.1. These include propagation delay in the fibre-optic links, serialisation delay, fabric delay, overhead processing delay and variable queuing delays in source Ethernet nodes and Ethernet switches. The screenshot of the network architecture implemented in NetSim® is given in Fig. 3.2.

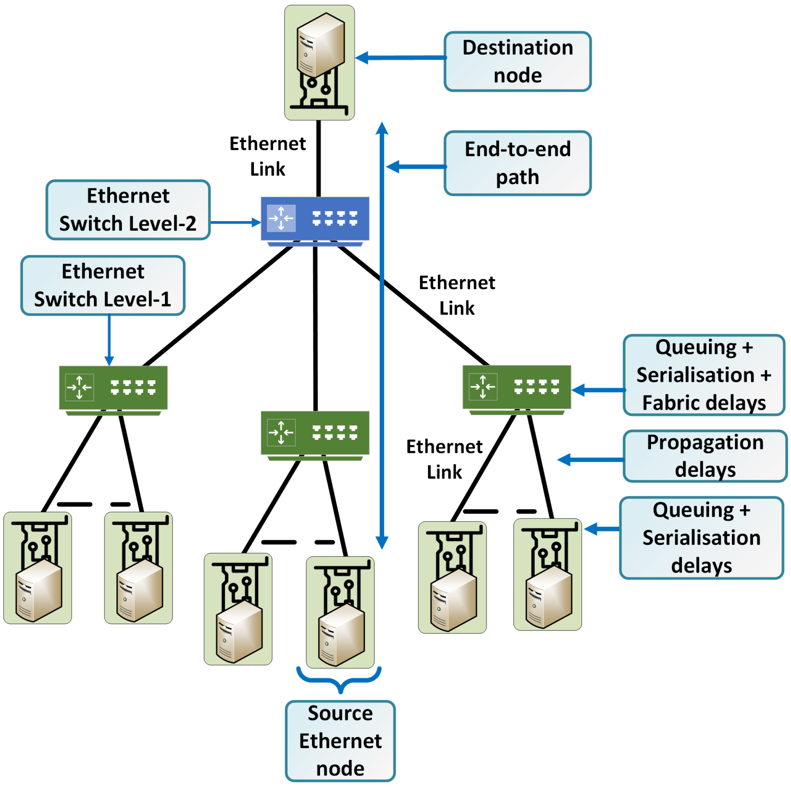


Fig. 3.1: Packet network with Ethernet-based aggregation.

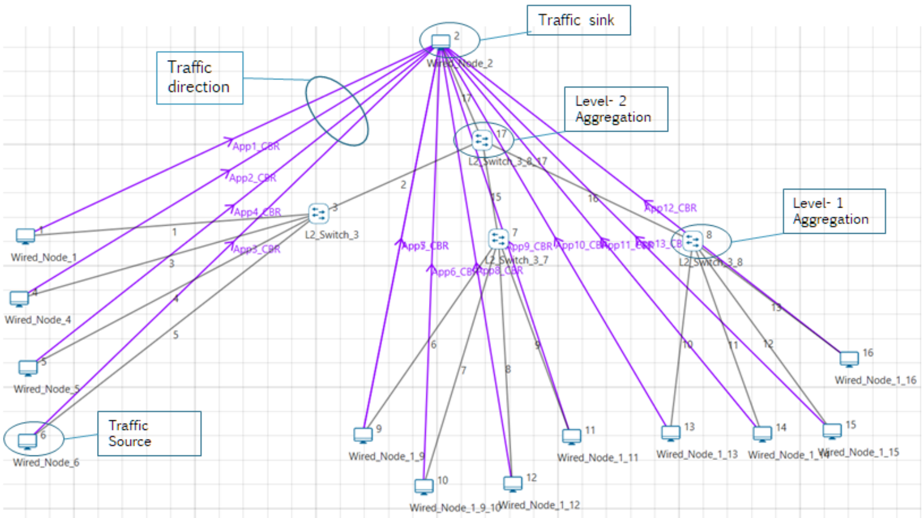


Fig. 3.2: Screenshot of the packet network in NetSim®for the network in Fig. 3.1.

Table 3.1: Link bit rates for simulation cases.

Case	Source to Level-1 link		Level-1 to Level-2 link		Level-2 to destination link	
	No of links	Bit rates (Mbps)	No of links	Bit rates (Mbps)	No of links	Bit rates (Mbps)
1	4	10	3	40	3	120
2	2 2	10 40	3	50	3	150
3	2 2	10 40	3	10	3	30
4	2 2	10 40	3	1000	3	3000
5	2 3	10 20	5	50		

Table 3.2: Normalised link bit rates for simulation cases.

Case	Source to Level-1 switch link		Level-1 to Level-2 switch link	
	No of links	Normalised input bit rates	No of links	Normalised input bit rates
1	4	0.25	3	0.3
2	2 2	0.2 0.8	3	0.3
3	2 2	1 4	3	0.3
4	2 2	0.01 0.04	3	0.3
5	2 3	0.2 0.4		

Different network cases exist, for example, in which all the links at level-1 Ethernet switches/aggregation nodes have the same input link bit rates or those in which the input link bit rates differ. Also, there may be cases where the output link bit rates of Ethernet switches may have exceeded or be equal to the sum of the input bit rates, or where they may be less than the sum, taking advantage of statistical multiplexing gains. There may also be cases where Ethernet switches may aggregate different numbers of input links at the same level. For these reasons, the network configuration in Fig. 3.1 was simulated (using NetSim®) with different input link bit rates listed in Table 3.1.

The simulations were carried out in the range of 10 Mbps to 3 Gbps to save time and memory resources. The input link bit rates of Ethernet switch links in Table 3.1 were normalised to the output link bit rates of Ethernet switches and listed in Table 3.2. The normalisation was carried out by dividing the input link rates by the output link rates for each Ethernet switch. The normalised link rates in Table 3.2 were calculated to analyze the results for the end-to-end delays for a generalised packet network, rather than a specific network with specific link rates.

As a starting point (case-1 in Table 3.1), the output bit rates of the source Ethernet nodes were set at 10 Mbps, the output bit rate of level-1 Ethernet switches was set at 40 Mbps and the output bit rate of the level-2 Ethernet switches was set to 120 Mbps for the network in Fig. 3.1. This was done so that the output link bit rates are equal to the total input link bit rates at both level-1 and level-2 Ethernet switches.

In the second simulation case (case-2 in Table 3.1), the level-1 Ethernet switches had mixed input link bit rates. The input link bit rates were 10 Mbps and 40 Mbps to level-1 Ethernet switches for the network in Fig. 3.1. The output link bit rates of the level-1 Ethernet switch were 50 Mbps and the output link bit rate of the level-2 Ethernet switches linked to the destination node was 150 Mbps in Fig. for the network in 3.1.

In the third simulation case (case-3 in Table 3.1), the input bit rates of the links to the level-1 Ethernet switches were the same as in case-2 at 10 and 40 Mbps in Fig. 3.1. However, the bit rate on the output link of level-1 Ethernet switches was set at 10 Mbps in Fig. 3.1. The output bit rate of the level-2 Ethernet switches linked to the destination node was 30 Mbps in Fig. 3.1.

In the fourth simulation case (case-4 in Table 3.1), the input link bit rates were the same as in case-2 and case-3, at 10 and 40 Mbps for the network in Fig. 3.1, however, the output link bit rate of level-1 Ethernet aggregation node was increased to 1000 Mbps. The level-2 Ethernet switches were linked to the destination node via a 3000 Mbps link for the network in Fig. 3.1.

Table 3.3: Load variation on links.

Case	Source to Level-1 links	Level-1 to Level-2 links	Level-2 to destination node links
	Load variation	Load variation	Load variation
1	0.1∼0.9	0.1∼0.9	0.1∼0.9
2	0.05∼0.45	0.1∼0.9	0.1∼0.9
3	0.01∼0.09	0.1∼0.9	0.1∼0.9
4	0.1∼0.9	0.01∼0.09	0.01∼0.09
5	0.1∼0.9	0.1∼0.9	

In the fifth case (case-5 in Table 3.1), a network with only two levels of Ethernet aggregation was assumed. In case-5, the number of input links to the level-1 Ethernet switch was increased from four to five. The objective of case-5 was to evaluate the model of end-to-end latency in the fronthaul, where the variation in packet inter-arrival times is likely higher compared to case-1, due to the increased number of input links at the level-1 aggregation node. This higher number of input links in case-5, compared to case-1, could also lead to greater contention among packets in the Ethernet switch queues. Additionally, only two levels of aggregation were tested in case-5 because the length of the packet traces generated from NetSim® simulations exceeded the random access memory capacity of the computers used for analysis.

In case-5 in Table 3.1, the source Ethernet nodes were linked to the level-1 Ethernet switch via two links of 10 Mbps and three links of 20 Mbps. The level-1 Ethernet switch was linked to the destination node via a 50 Mbps link.

To evaluate the validity and accuracy of the model for end-to-end latency, simulations were conducted using various combinations of load variations on the input and output links of the aggregation nodes, of the network in Fig. 3.1. These combinations are detailed in Table 3.3. The load variations reflect different traffic conditions with different variances in the inter-arrival times of packets. Different waiting times in the queues at the nodes within the network shown in Fig 3.1 are caused by these variations in the inter-arrival times of packets. By analysing these scenarios, a better understanding of how different traffic loads impact overall network latency can be achieved.

In simulation case-1, the third row of Table 3.3, the loads at the input links to the level-1 Ethernet switches, in Fig. 3.1 were swept from a low load of 0.1 to a high load of 0.9, so that the model for end-to-end latency in fronthaul can be used to predict delays for a wide range of loads on aggregation nodes. The load variations of (0.1 $\sim$ 0.9) resulted in the same load variations at level-1 and level-2 Ethernet switches for the network in Fig. 3.1, as listed in the third row of Table 3.3. In this, case-1, the variation of inter-arrival times of packets at the input ports of level-1 Ethernet switches was higher than its output ports as the variations are smoothed out by the serialisation of packets at the output ports of the Ethernet switches as discussed in Section 2.5.

In simulation case-2, the fourth row of Table 3.3, the load on each input link to level-1 Ethernet switches were swept from a low load of 0.05 to a moderate load of 0.45. This resulted in the total load on the output link of the level-1 Ethernet switches varying from a low load of 0.1 to a high load of 0.9. The purpose of testing this simulation case-2 was to test the model for the end-to-end latency in a fronthaul network, where the load variations on aggregation nodes become more sensitive to the load variations at input links in comparison to case-1. Lower load variations at input links also indicate longer inter-arrival times of packets with higher variance, at the aggregation node's queues, in comparison to case-1 in Table 3.3.

In simulation case-3, in the fifth row of Table 3.3, the load on each input link of level-1 Ethernet switches was swept from a low load of 0.01 to a low load of 0.09. This resulted in the total load on the output link of the level-1 Ethernet switches varying from a low load of 0.1 to a high load of 0.9. The purpose of testing this simulation case-2 was to test the model for the end-to-end latency in a fronthaul network, where the load variations on aggregation nodes become more sensitive to the load variations at input links in comparison to both case-1 and case-2. Lower load variations at input links also indicate longer inter-arrival times of packets with higher variance, at the aggregation node's queues, in comparison to both case-1 and case-2 in Table 3.3.

In simulation case-4, the sixth row of Table 3.3, the load on the input links to the level-1 Ethernet aggregation nodes was varied from a low load of 0.1 to a higher load of 0.9, which resulted in a very low load variation of 0.01 to 0.09 at the output links of level-1 Ethernet switches. The purpose of testing this simulation case-2 was to test the model for the end-to-end latency in a fronthaul network, where the load variations on aggregation nodes become far less sensitive to the load variations at input links in comparison to case-1, case-2 and case-3. Higher load variations at input links, compared to output links, also indicate smaller inter-arrival times of packets with low variance, at the aggregation node's queues, in comparison to case-2 and case-3 in Table 3.3.

In simulation case-5 in Table. 3.1, the load on the input links to the level-1 Ethernet switch varied from 0.1 to 0.9, so the load on the output link of the level-1 Ethernet node varied from a low load of 0.1 to a high load of 0.9.

Table 3.4: Simulation parameters for NetSim®.

Average packet length	$\overline{L_p}$	735 bytes
Standard deviation of packet lengths	$\sigma_{\overline{L_p}}$	300 bytes
Mean inter-generation times of packets	$^1/\lambda_{(t,1)}$	$^{735\text{ bytes}}/(\rho \times R_{o,1})$
Output link rates of source Ethernet nodes	$R_{o,1}$	10,40, 20 Mbps
Standard deviation of inter-generation times of packets	$\sigma_{(a,1)}$	300μs

In 5G networks and beyond, different service/slice traffic may show bursty or continuous behaviour. Aggregating such traffic patterns will result in traffic having a normal distribution of inter-arrival time of packets from the central limit theorem. For simulations, the traffic, originating from source Ethernet nodes (Fig. 3.1), followed a Normal distribution of packet inter-arrival times.

Although queuing theory models such as G/G/1 and M/D/1, assume random packet lengths with Generalised and Poisson distribution for analytical simplicity, the research in the paper aims at the analysis beyond simple statistical assumptions. Considering that the fact Ethernet frames/packet lengths can vary between 64 and 1518 bytes, the packet lengths were set to have a Normal distribution with a mean of 735 bytes and a standard deviation of 300 bytes as listed in Table 3.4. The mean inter-generation time of packets was adjusted using the desired load variations on the output links as listed in Table 3.3. The parameters for simulations in Netsim® are listed in Table 3.4.

3.3 Results for the Estimation of Mean Waiting Times Using G/G/1 Queuing Model

As discussed in Section 2.5, serialisation delay is defined as the time required by Ethernet switches and source nodes, to process or transmit the packets from the output ports of the nodes. It is defined as the ratio of the average packet length and output link bit rate of the nodes, or the inverse of the packet service rate as defined in Equation 2.7.

Presenting the waiting times, as well as end-to-end delays, normalised to the serialisation delays of the nodes in a network, ensures the scalability of the results to various fronthaul network configurations with different bit rate links than those used to generate the results. The delays can be scaled by using the product of the normalised delays and the serialisation delays of the target network. The normalised waiting times represent the mean waiting time of packets in the queue, adjusted for the bit rate of the output link. This metric is essential for understanding the efficiency of packet transmission and the overall performance of the network. Specifically, it indicates how long packets are expected to wait in the queue relative

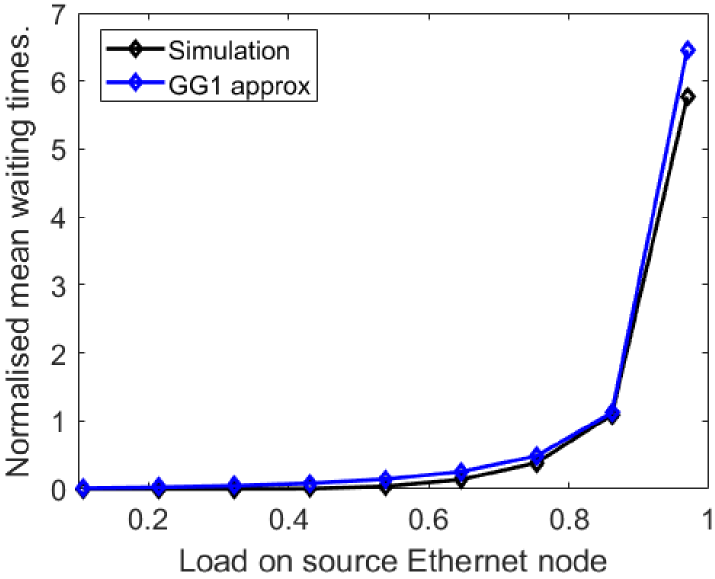


Fig. 3.3: Mean waiting times at source Ethernet node and G/G/1 approximation vs load (case-1).

to the capacity of the link. A lower normalised delay suggests a more efficient network with quicker packet transmission, while a higher normalised delay points to potential congestion and longer waiting times.

In this section, the mean waiting times of different nodes in the network in Fig. 3.1, normalised to serialisation delays of the output links rates are presented using the G/G/1 approximation in Equation. 2.5. The normalised mean waiting times of the source Ethernet nodes in Fig. 3.1, using the G/G/1 approximation in Equation. 2.5 are shown in Fig. 3.3. From the observation of Fig. 3.3, it can be concluded that the G/G/1 approximation can estimate the mean waiting times in source Ethernet nodes very accurately.

The mean waiting times in the source nodes for the case (2∼5) are also given in the following Fig. 3.4 to Fig. 3.7. From the analysis of Fig. 3.4 to Fig. 3.7 it was concluded that the G/G/1 approximation overestimates the mean waiting time at low loads; however, the overestimation reduces as the load increases.

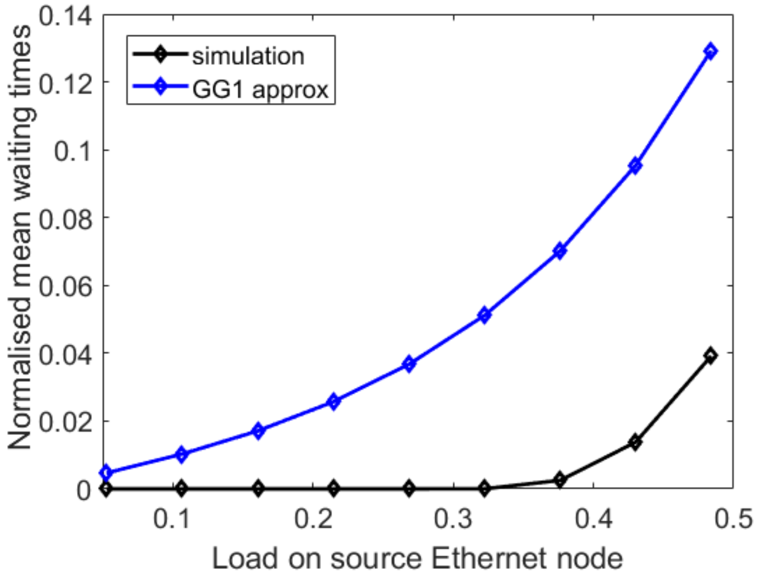


Fig. 3.4: Mean waiting times at source Ethernet node and G/G/1 approximation vs load (case-2).

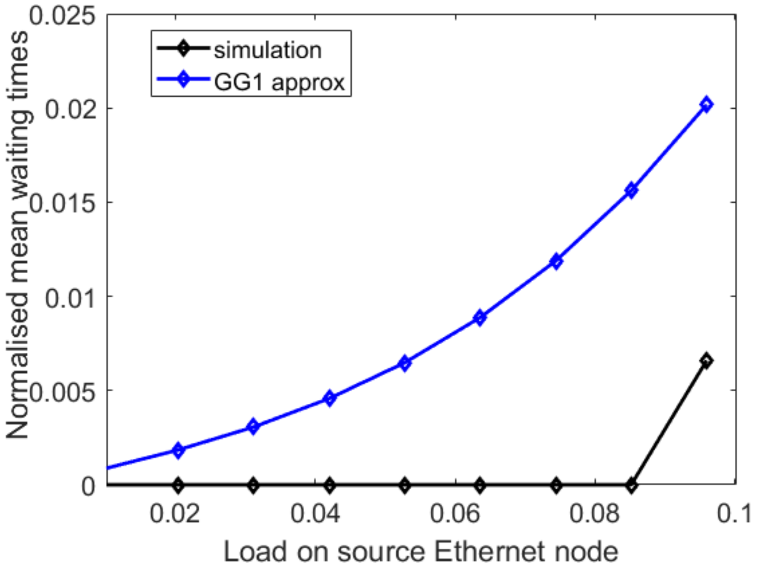


Fig. 3.5: Mean waiting times at source Ethernet node and G/G/1 approximation vs load (case-3).

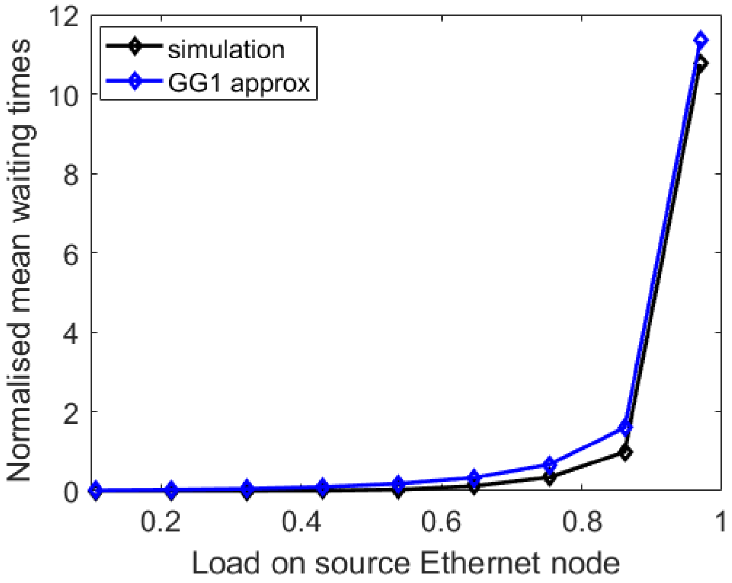


Fig. 3.6: Mean waiting times at source Ethernet node and G/G/1 approximation vs load (case-4).

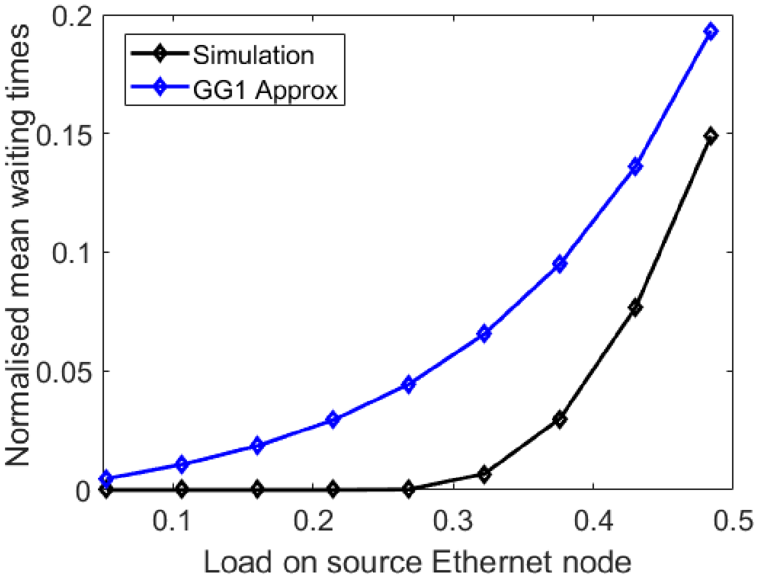


Fig. 3.7: Mean waiting times at source Ethernet node and G/G/1 approximation vs load (case-5).

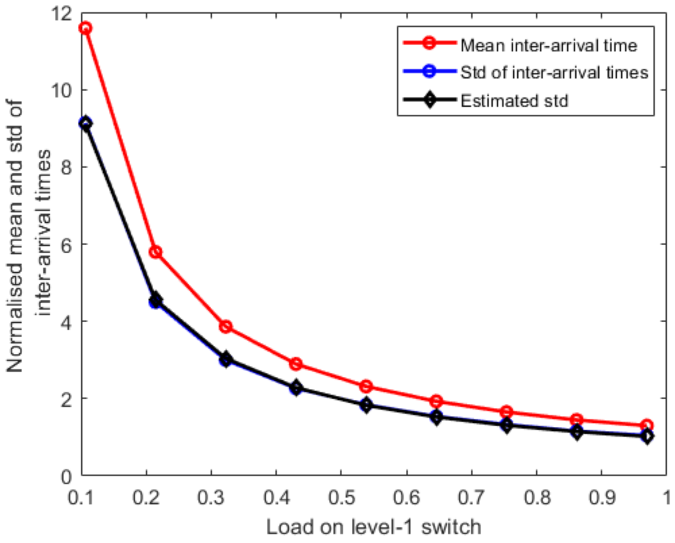


Fig. 3.8: Mean and standard deviation of inter-arrival times of packets at level-1 Ethernet switch (case-1).

At a source Ethernet node, $\sigma_{(a,1)}$ may be known as it is defined solely by the characteristics of the uplink mobile traffic and its aggregation by the RU. However, at level-1 and level-2 Ethernet switches, the standard deviation of inter-arrival times of arriving traffic will be smoothed out by serialisation at preceding nodes as shown in Fig. 2.18 in Section 2.5. Thus, $\sigma_{(a,i)}$ at the Ethernet switches must be estimated. This can be done via the modelling of $\sigma_{(a,i)}$ as a nonlinear function of the inter-arrival time of packets given as follows,

$$\sigma_{(a,i)} = a + \ b\, \Delta t_i + \ c\, \Delta t_i^2 + \ d\, \Delta t_i^3 \ \forall \ \ x \in \{2,3\} \tag{3.1}$$

where Δt_i is the inter-arrival times of packets at any node in the path in fronthaul is given as follows,

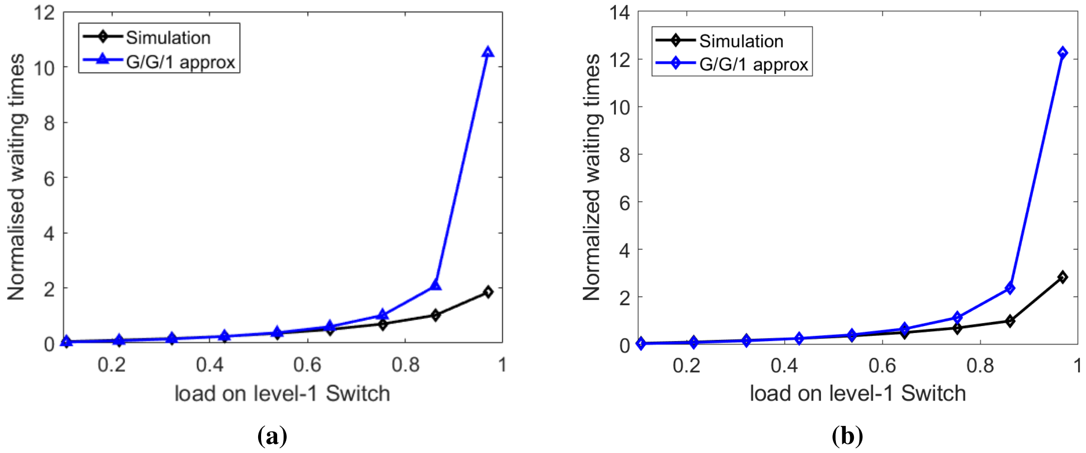
$$\Delta t_i = \frac{1}{\lambda_{t,i}} \ \forall \ i \in \{2,3\}, \tag{3.2}$$

where a, b and c are the coefficients of the nonlinear function obtained by curve fit. The mean and standard deviation of the arrival rate of packets, normalised to the output link bit rate of level-1 Ethernet switch, for simulation case-1, for different loads are shown in Fig. 3.8. The values of constants in Equation 3.1 using nonlinear curve fit are given as follows,

$$a = 5.728\text{x}10^{-6},\ b = 0.6713,\ c = 0.6892,\ d = 0.7482.$$

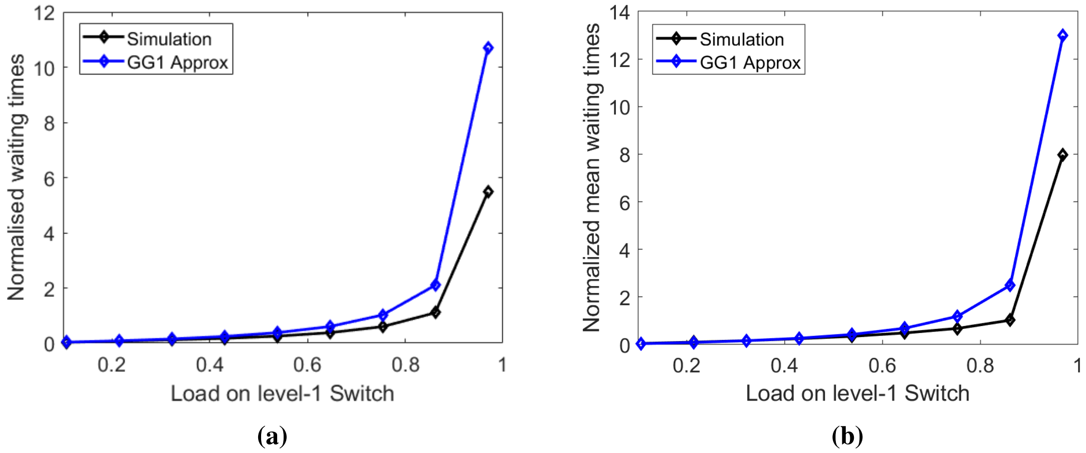
Note that the values for the coefficients a, b and c will remain fixed throughout the analysis. From the observation of Fig. 3.8, we can conclude that as the load on the switch increases, the standard deviation of the inter-arrival times of the packet reduces. This is because, at earlier aggregation nodes or source Ethernet nodes, the variation in inter-arrival times of packets is smoothed out in the serialisation buffers at the egress ports.

The estimated values of the standard deviation of inter-arrival times of packets in Equation 3.1 are used as an input parameter for the G/G/1 queuing model in Equation 2.5. The mean waiting times in level-1 and level-2 Ethernet switches using the G/G/1 queuing model in Equation 2.5, normalised to output link rates, are given Fig. 3.9. From observing the two figures, it can be concluded that the G/G/1 approximation overestimates the mean waiting times at loads higher than 0.85. This might be because the G/G/1 approximation of mean waiting times is based on the assumption that the inter-arrival times of packets and packet lengths are generalised rather than specific. This overestimation will result in extremely high overestimation of the percentile delays. This may result in the over-provision of buffering resources at the destination nodes in the network to reduce the number of packets dropped.



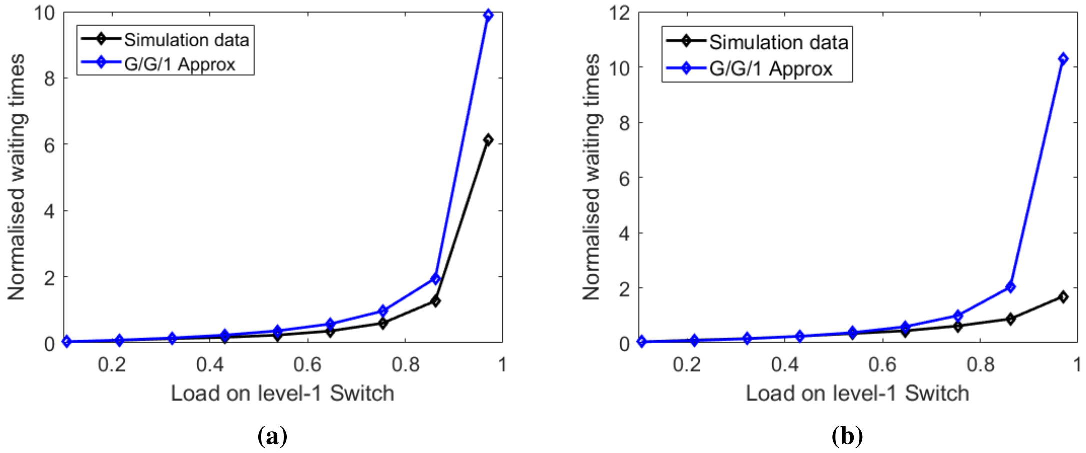
(a) **(b)**

Fig. 3.9: Normalised mean waiting times of packets at level-1 (a) and level-2 (b) Ethernet switches for case-1.



(a) **(b)**

Fig. 3.10: Normalised mean waiting times of packets at level-1 (a) and level-2 (b) Ethernet switches for case-2.



(a) **(b)**

Fig. 3.11: Normalised mean waiting times of packets at level-1 (a) and level-2 (b) Ethernet switches for case-3.

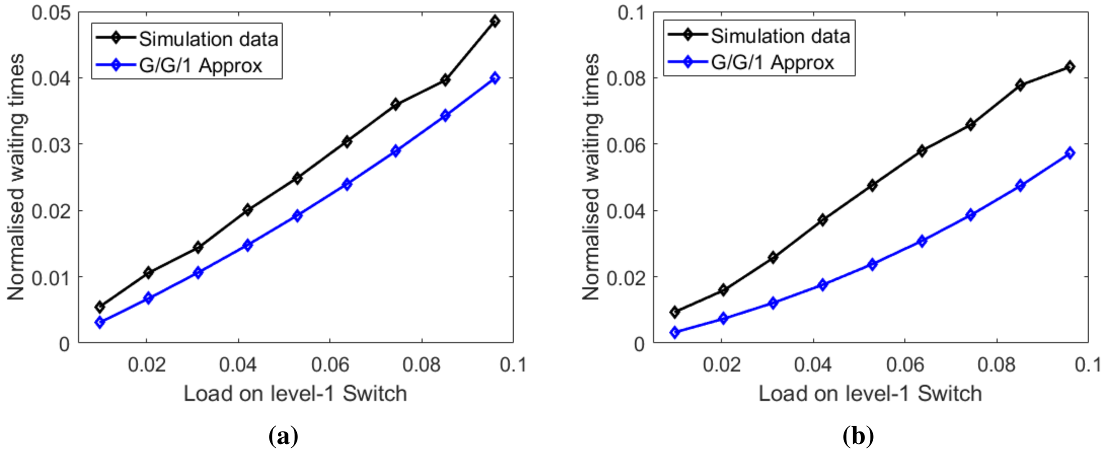


Fig. 3.12: Normalised mean waiting times of packets at level-1 (a) and level-2 (b) Ethernet switches for case-4.

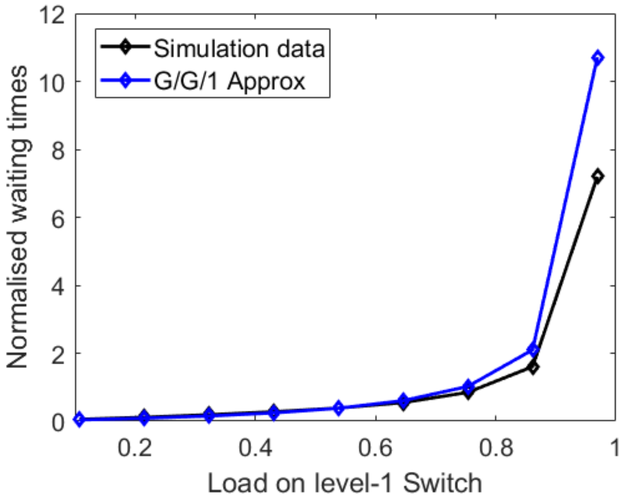


Fig. 3.13: Normalised mean waiting times of packets at level-1 Ethernet switches for case-5.

Similar trends for the estimation of mean waiting times using the G/G/1 model were observed for case-2, case-3 and case-5 and shown in Fig. 3.10, Fig. 3.11 and Fig. 3.13 respectively. However, for case-4 the G/G/1 queuing model underestimates the mean waiting for the level-1 and level-2 Ethernet switches, with results shown in Fig. 3.12. The underestimation at extremely low loads in case-4 will not have a significant impact on estimation percentile delays since the major portion of the total mean end-to-end waiting times arises in the source Ethernet nodes which are more heavily loaded than the switches.

3.4 Results for the Estimation of Mean Waiting Times Using Lindley's Recursions

To overcome the overestimation of mean waiting time at high loads, Lindley's recursion method in Equation .2.12 will be used for the estimation of mean waiting times in the level-1 and level-2 Ethernet switches. As mentioned in Section 2.5.2, the pdf, $f_{B_n}(t)$, depends on the statistical distribution of the packet lengths configured in the simulation setup which has a Normal distribution given as follows,

$$f_{B_n}(t) = \frac{1}{\sigma_{r,i}\sqrt{2\pi}} e^{-\frac{(t-{}^{1}/_{\mu_{r,i}})^2}{2\ \sigma_{r,i}^2}} , \tag{3.3}$$

where $\sigma_{r,i}$ is the standard deviation of service time of packets defined in Equation.2.9 and $\mu_{r,i}$ is the mean service rate of packets defined in Equation. 2.7. Note that the parameters $\mu_{r,i}$ and $\sigma_{r,i}$ can be calculated using the parameters of average packet lengths $\overline{L_p}$ and standard deviation of packet lengths $\sigma_{\overline{L_p}}$ defined in Table 3.4.

Similarly the PDF, $f_{A_n}(t)$, in Equation. 2.12 in Section 2.5.2 depends on the statistical distribution of inter-arrival times of packets at the source node or the aggregation node. Since it was assumed that the PDF of inter-arrival times of packets at an RU or in this case at source Ethernet node have a Normal distribution, in simulations in NetSim ®, the inter-generation times of packets at source Ethernet nodes were configured to have a Normal distribution. The PDF of inter-generation times of packets at the source Ethernet nodes is given as follows,

$$f_{A_n}(t) = \frac{1}{\sigma_{(a,1)}\sqrt{2\pi}} e^{-\frac{(t-{}^{1}/_{\lambda_{t,1}})^2}{2\sigma_{(a,1)}{}^2}} , \tag{3.4}$$

where $\sigma_{(a,1)}$ is the standard deviation of inter-generation times of packets given in Table 3.4 and ${}^{1}/_{\lambda_{t,1}}$ is the mean inter-generation time of the packet also defined in Table 3.4.

Lindley's recursion requires the PDF of inter-arrival times of packets at a node to model the PDF of waiting times of packets, which is known in the source Ethernet node as the generated traffic was assumed to have a Normal distribution. However, at the Ethernet switches, the PDFs of inter-arrival times of packets need to be investigated through the analysis of simulation data, since the variations of inter-arrival times of packets smooth out at the output ports of prior nodes in the path. For this reason, the inter-arrival times of packets at level-1 Ethernet switches were analyzed and compared with different PDFs. For the simulation case-1 in Table 3.2 it was observed that using Generalised Pareto Distribution (GPD) resulted in the least RMS error between the measured and estimated PDFs inter-arrival times of packets at level-1 Ethernet switches. The comparison between the two PDFs is

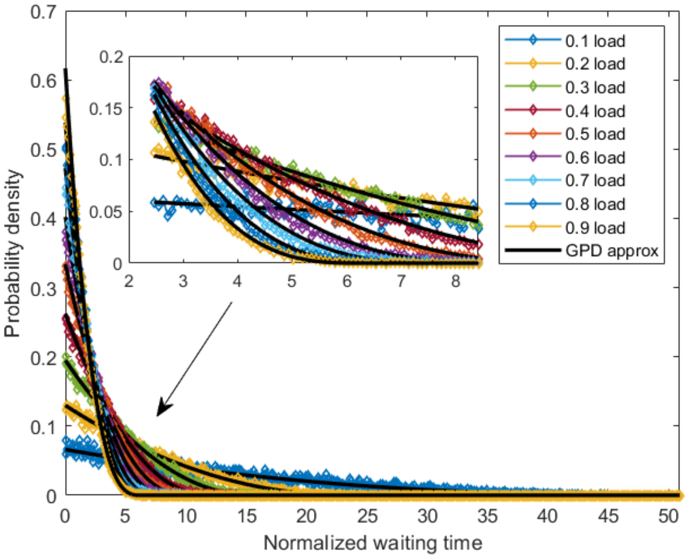


Fig. 3.14: PDFs of inter-arrival times of packets at level-1 Ethernet aggrergation case-1.

shown in Fig. 3.14 for simulation case-1 in Table 3.2, from which it can be concluded that the PDFs of inter-arrival times packets at Ethernet switches follow GPD distribution. The derivation of the parameters for GPD dependent on the simulation parameters in Table 3.4 is as follows;

The PDF of inter-arrival times of packets, $f(A_{-n})$, at level-1 and level-2 Ethernet switches, using GPD distribution are given as follows [8],

$$f_{A_n}(t) = \frac{1}{\beta}\left(1+\frac{\xi(t-\propto)}{\beta}\right)^{\left(-\frac{1}{\xi}-1\right)}. \tag{3.5}$$

The mean, μ_{GPD}, and variance, σ^2_{GPD}, of a GPD, $f_{A_n}(t)$, are related to parameters $\propto, \xi$ and β is defined as follows,

$$\mu_{GPD} = \propto + \frac{\beta}{1-\xi}, \tag{3.6}$$

and

$$\sigma^2_{GPD} = \frac{\beta^2}{(1-\xi)^2(1-2\xi)}. \tag{3.7}$$

If it is assumed that the mean inter-arrival times of packets, Δt_i, in Equation 3.2 is equal to the mean, μ_{GPD}, in Equation 3.6 and the variance of the inter-arrival time of packets, $\sigma^2_{(a,i)}$, in Equation 3.1 is equal to the variance, σ^2_{GPD}, in Equation 3.7, then the parameters of, ξ, and, β, in Equation 3.5 can be calculated as follows,

$$\xi = \frac{1}{2}\left(1 - \frac{\Delta t_i^2}{\sigma_{a,i}^2}\right), \tag{3.8}$$

and

$$\beta = \Delta t_i \,(1 - \xi)\,. \tag{3.9}$$

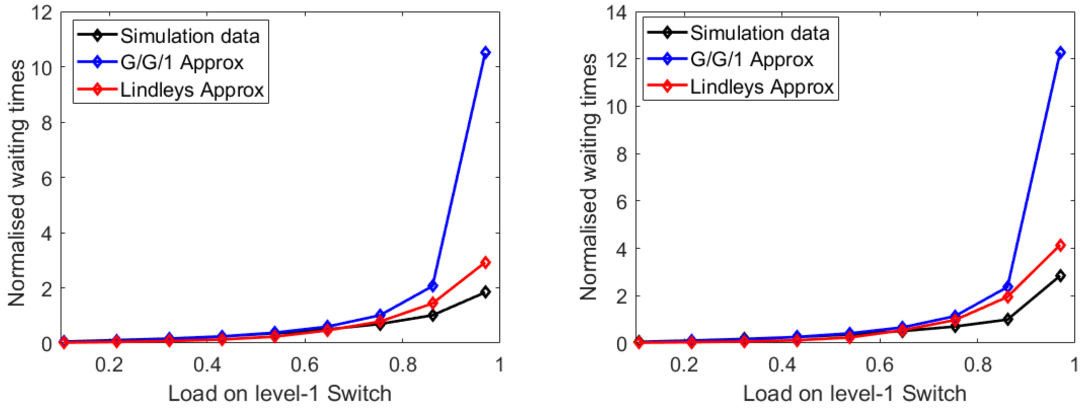
After modelling the PDFs of, $f_{A_n}(t)$ and $f_{B_n}(t)$, the PDF of the overall waiting times in the queue, $f_{T_{w,n+1}}(t)$, was calculated via iterative convolutions given in Equation 2.12 using the Algorithm. 1, [9]. After modelling the PDF of waiting times, $f_{T_{w,n+1}}(t)$, the mean waiting times in queues were calculated using the following relation,

$$\bar{T}'_{w,i} = \int f_{T_{w,n+1}}(t) * t \; dt, \tag{3.10}$$

where i is the index of aggregation nodes in the fronthaul and t is the period over which the PDF is modelled.

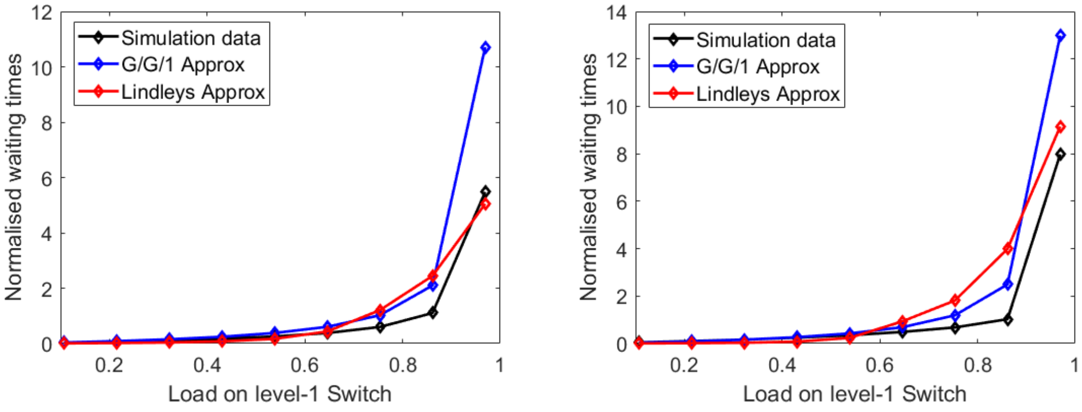
The results for the estimation of normalised mean waiting times in the level-1 and level-2 Ethernet switches using Lindley's recursion for the simulation cases (1∼5), using Equation 3.10, are shown in the following Fig. 3.15 ∼ Fig. 3.19. The results from the estimates of mean waiting times using Lindley's recursion are re-plotted in red colour over the results for the mean waiting times using the G/G/1 model in Fig. 3.9 to Fig. 3.13 for comparison. Specifically, for the results of case-1 in Fig. 3.15, the mean waiting times are overestimated by a factor of 1.43 at a load of 0.85 and a factor 1.58 at a load of 0.96 for the level-1 switch. Similarly, the mean waiting times are overestimated by a factor of 1.96 times and 1.46 times for the same loads at the level-2 switch. The overestimation by the G/G/1 model is higher than the overestimation from Lindley's recursion model by factors of 7.95 and 7.25 at the load of 0.96 at the level-1 and level-2 switch.

The trends of over-estimation of mean waiting times at level-1 and level-2 switches can also be observed for case-2, case-3 and case-5 in Fig. 3.16, Fig. 3.17 and Fig. 3.19 ∼ Fig. 3.19, from which it can be concluded that the overestimation of mean waiting times is far less than that from the G/G/1 model for loads above 0.85, whereas, for loads below 0.85, the overestimation from G/G/1 queuing model is less than Lindley's recursion. So Lindlely's recursion-based estimation can be considered as a more accurate upper bound on the estimates of the mean waiting times than the G/G/1 model for load above 0.85.



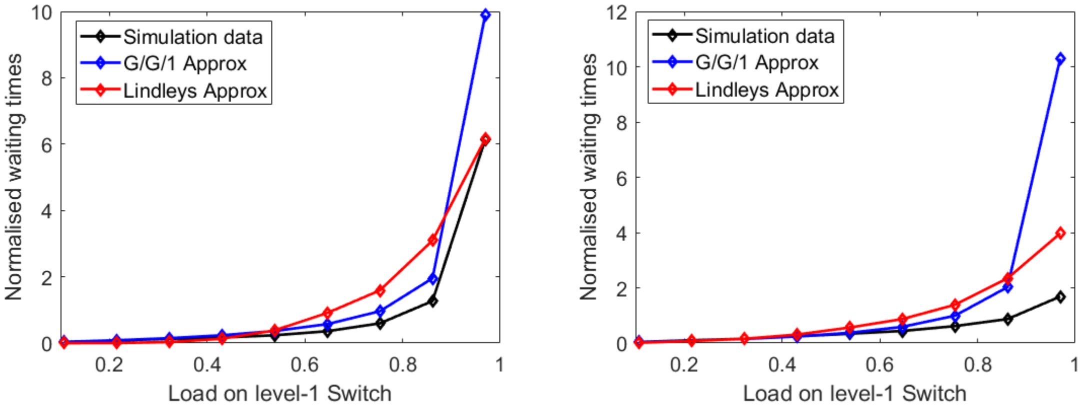
(a) **(b)**

Fig. 3.15: Normalised mean waiting times of packets at level-1 (a) and level-2 (b) Ethernet switches for case-1.



(a) **(b)**

Fig. 3.16: Normalised mean waiting times of packets at level-1 (a) and level-2 (b) Ethernet switches for case-2.



(a) **(b)**

Fig. 3.17: Normalised mean waiting times of packets at level-1 (a) and level-2 (b) Ethernet switches for case-3.

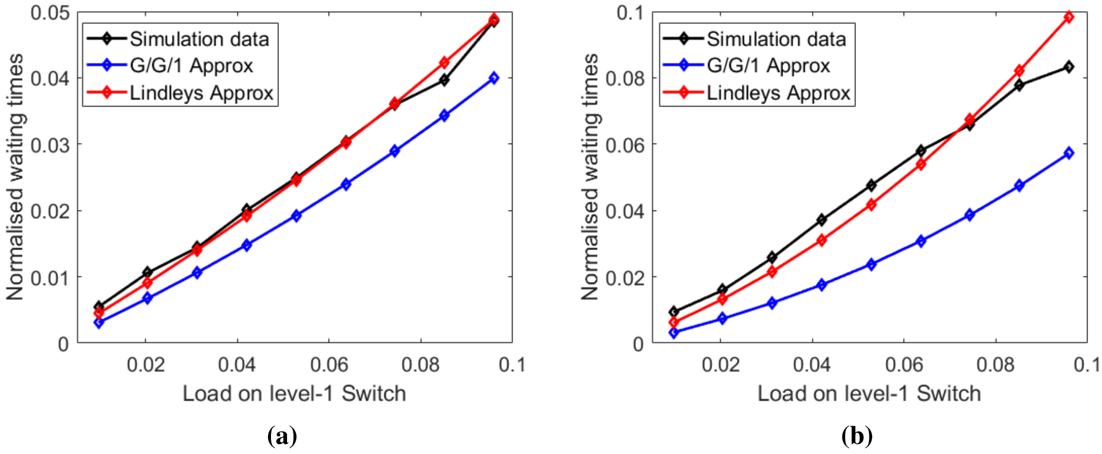


Fig. 3.18: Normalised mean waiting times of packets at level-1 (a) and level-2 (b) Ethernet switches for case-4.

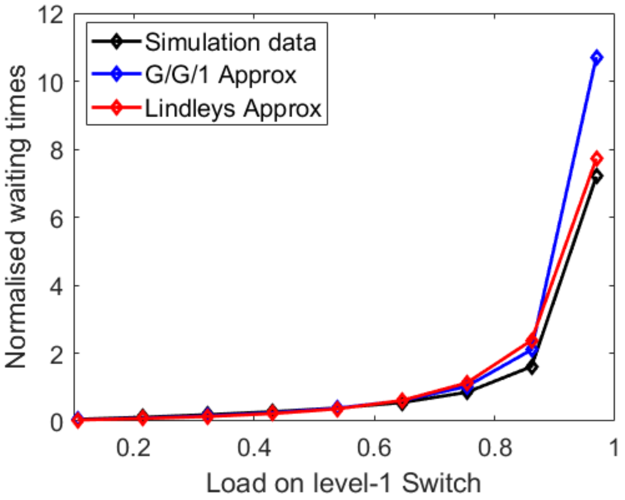


Fig. 3.19: Normalised mean waiting times of packets at level-1 Ethernet switches for case-5.

For the case-4 in Fig. 3.18 the G/G/1 queueing model underestimated the mean waiting times in Ethernet switches at level-1 and level-2 aggregation, however, the Lindley's recursion-based model was able to estimate mean waiting times from the simulation results more accurately.

3.5 Estimation of End-to-End Mean, Standard Deviation, PDF and Percentile Delays in Fronthaul

In the previous Section, it was noted that the G/G/1 queuing model overestimates mean waiting times more significantly than Lindley's recursion-based model when the load exceeds 0.85. To compute the total mean waiting time for a path from a source to a destination mode: the G/G/1 queuing model is utilised for loads below 0.85, while Lindley's based recursion model is utilised for loads above. This method is detailed as follows,

$$\overline{T_w} \approx \begin{cases} \sum_{i=1}^{i=N} T_{w,i} \ , \{0 < \rho_i \leqslant 0.85\} \\ \\ \sum_{i=1}^{i=N} T'_{w,i} \ , \{0.85 < \rho_i \leqslant 1\} \end{cases} \tag{3.11}$$

where N is the total number of nodes in the path between the source Ethernet node and destination CU/DU, $T_{w,i}$ is the mean waiting time from the G/G/1 approximation from Equation 2.5 and $T'_{w,i}$ is the mean waiting time from the Lindley's recursion model is given in Equation 3.10.

The total mean end-to-end time, $\bar{T_e}$, spent by packets in a path from source Ethernet node to destination Ethernet node Fig. 3.1 is then equal to the sum of the mean waiting times in queues from Equation 3.11, the sum of the mean serialisation delays from Equation 3.13 given as follows,

$$\bar{T_e} = \overline{T_w} + \bar{T_s}, \tag{3.12}$$

where $\bar{T_s}$ is the average total serialisation delay for a path in fronthaul is defined as follows

$$\bar{T_s} = \overline{L_p} \sum_{i=1}^{i=N} \frac{1}{R_{o,i}}, \tag{3.13}$$

where N is the number of aggregation nodes in the link between RU and CU/DU, $R_{o,i}$ is the output rate of the i^{th} aggregation node and $\overline{L_p}$ is the average length of packets.

The total mean end-to-end delays for a path in Fig. 3.1 using Equation 3.12 were normalised to the total serialisation delay using Equation 3.13, $\bar{T_s}$, given as follows,

$$\overline{T_{e,n}} = \frac{\bar{T_e}}{\bar{T_s}}. \tag{3.14}$$

The normalisation was done to make the results scalable to paths having different serialisation delays or output port bit rates of the nodes, from the ones in Fig. 3.1. The total mean end-to-end delays for a path having different bit rates can be obtained by using the product of normalised mean end-to-end delays in Equation 3.14 and total serialisation delays of the target network from Equation 3.13.

The normalised total mean end-to-end delays for a path in Fig. 3.1, estimated from Equation 3.14 and compared with the simulation results for the five simulation cases (2∼5) are shown in the Fig. 3.20 ∼ Fig. 3.24. The results from the estimates of mean end-to-end delays are given in blue curves while the total mean end-to-end delays from simulations are given in black curves. Specifically for the results of case-1 in Fig. 3.20, the percentage error between measured and estimated total mean end-to-end delays is 0.95% at 0.1 load, 7% at 0.75 load, 0.02% at 0.96 load. From the analysis of the results in Fig. 3.21 ∼ Fig. 3.24 we can also conclude that the estimates for the total mean end-to-end delays are higher than the estimated total mean end-to-end-delays.

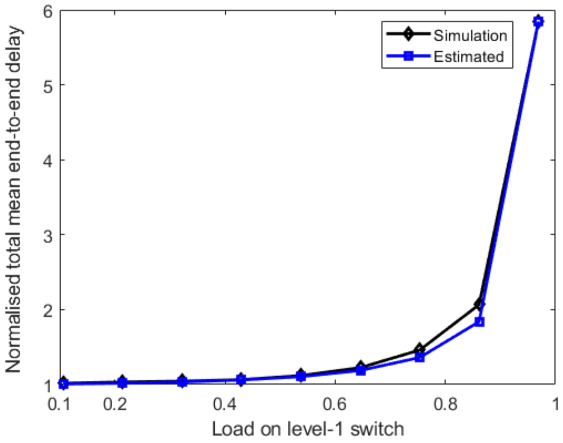


Fig. 3.20: Normalised total mean end-to-end delays vs load (case-1).

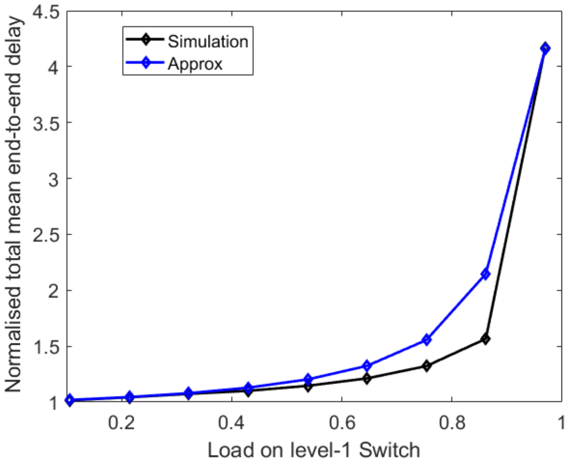


Fig. 3.21: Normalised total mean end-to-end delays vs load (case-2).

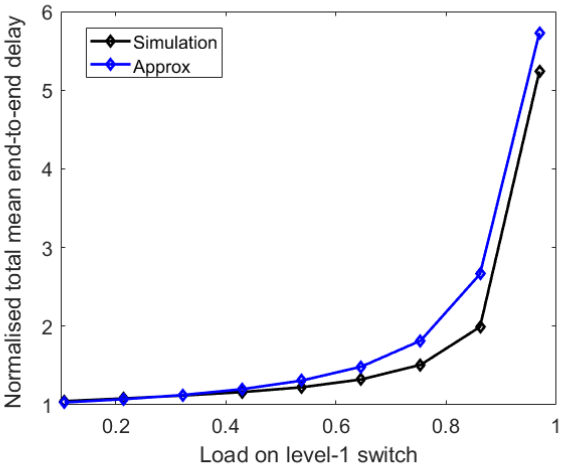


Fig. 3.22: Normalised total mean end-to-end delays vs load (case-3).

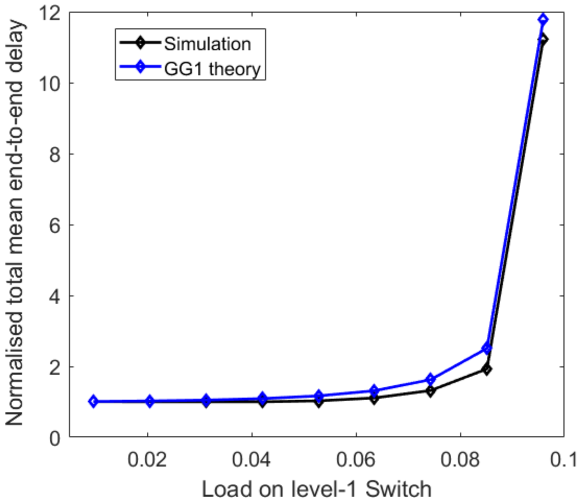


Fig. 3.23: Normalised total mean end-to-end delays vs load (case-4).

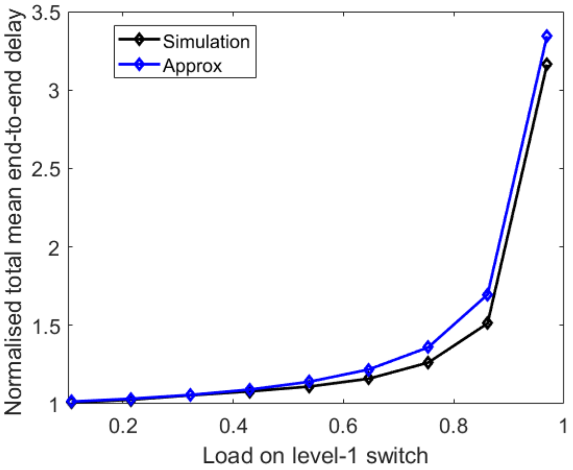


Fig. 3.24: Normalised total mean end-to-end delays vs load (case-5).

In the next step, the standard deviation of end-to-end delay was modelled. For this, both the mean and standard deviation of total waiting times were observed for the simulation case-1 for an end-to-end path in Fig. 3.1. The mean and standard deviation of total waiting times, normalised to total serialisation delay of packets are shown in Fig. 3.25. In Fig. 3.25, it can be noticed that the standard deviation of waiting times is almost equal to the mean waiting times calculated from packet traces. Therefore it was assumed that the standard deviation of end-to-end delays is approximately equal to the sum of the mean of waiting times in queues in the nodes as well as the standard deviation of serialisation delays in the nodes in a path in fronthaul. The total standard deviation, σ_t, of end-to-end delays is therefore expressed as follows,

$$\sigma_t \approx \overline{T_w} + \sum_{i=1}^{i=N} \sigma_{r,i}, \tag{3.15}$$

where, $\overline{T_w}$, is the total mean end-to-end waiting time from Equation 3.12, $\sigma_{r,i}$ is the standard deviation of service time at the node in the path in fronthaul and N is the number of nodes in the path.

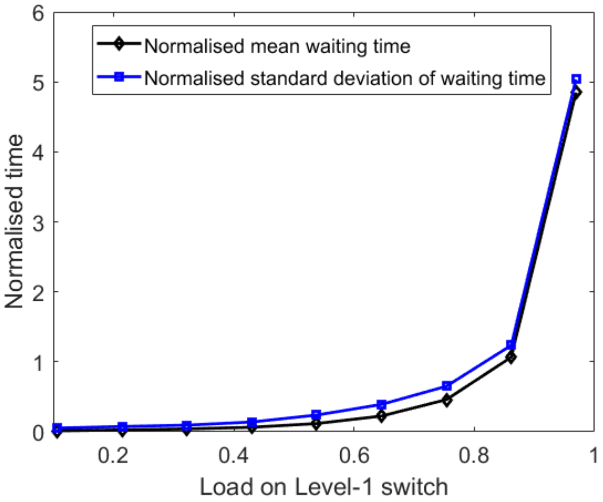


Fig. 3.25: Means and standard deviation of waiting times in source Ethernet node (case-1).

In the next step, the probability density function of the overall end-to-end delays was estimated to aid the estimation of percentile delays. For this different probability density functions were tested for the best match between the measured PDFs from the packet traces from NetSim®. It was concluded that the lognormal distribution had the least root means square (RMS) error from the measured PDFs. The lognormal PDF for the total end-to-end times T_e is given as follows [8],

$$f_{T_e}(t) = \frac{1}{t\sigma\sqrt{2\pi}} e^{-\frac{(log(t)-\mu)^2}{2\sigma^2}}. \tag{3.16}$$

For the lognormal distribution in Equation. 3.16, the parameters of μ and σ are defined using the total mean end-to-end delay T_e in Equation. 3.12 and standard deviation of total mean end-to-end delay σ_t in Equaion. 3.15,

$$\mu = \ln\left(\frac{\bar{T_e}^2}{\sqrt{\bar{T_e}^2 + \sigma_t^2}}\right), \tag{3.17}$$

$$\sigma = \sqrt{\ln\left(1 + \frac{\sigma_t^2}{\bar{T_e}^2}\right)}. \tag{3.18}$$

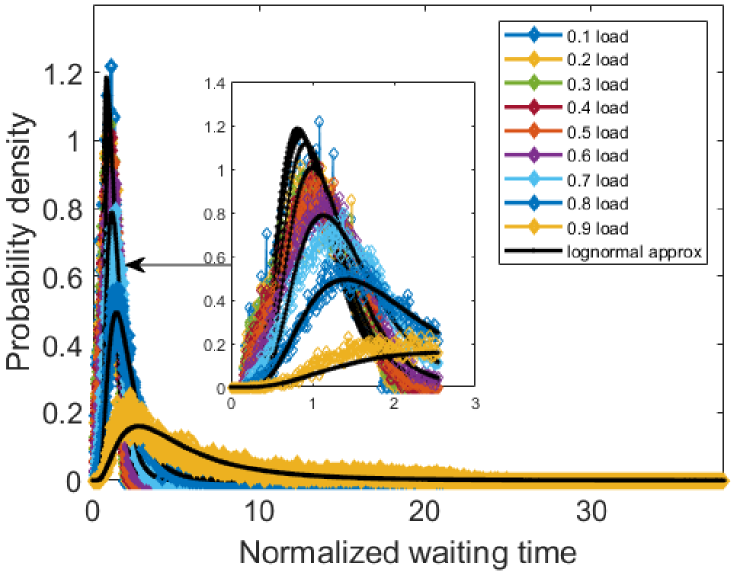
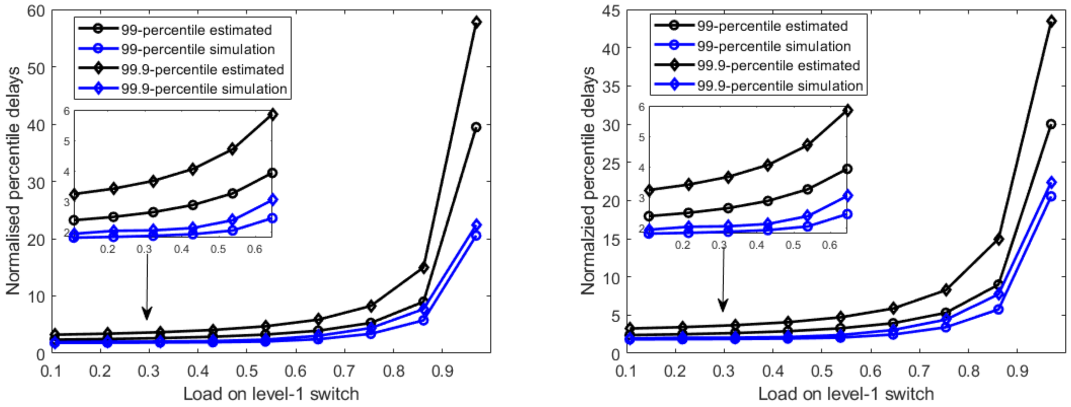


Fig. 3.26: PDFs of end-to-end waiting times (case-1).

The probability density functions of end-to-end delays normalised to total serialisation delay for case-1 are shown in Fig. 3.26, comparing those obtained from simulations with log-normal distributions in Equation 3.16. In Fig. 3.26 it can be observed that the PDFs of the estimated end-to-end delays are correlated with the ones obtained from the simulations at low loads since the RMS error between the estimated and measured PDFs in Fig. 3.26 is 1.5% at the load of 0.1 whereas it was 5.13%at a load of 0.96.



(a) **(b)**

Fig. 3.27: Predicted and measured percentile delays vs load on switch-1 (case-1). Using mean delays from G/G/1 model (a) and Lindley's recursion for loads above 0.85 (b).

Using the estimated PDFs, the 99-and 99.9 percentile delays T_p are estimated using the following relation [10],

$$T_p = \exp(\mu + \sqrt{2}\sigma \cdot \mathrm{erf}^{-1}(2p-1)) \quad \forall \quad p \in \{0.99, 0.999\}, \tag{3.19}$$

where p is the required percentile, erf^{-1} is the inverse error function and μ and σ are defined in Equation 3.17 and Equation 3.18.

To assess the accuracy of the proposed method till Equation 3.19, the estimated percentile delays are also compared with the case when only the G/G/1 queuing model in Equation 3.11 is used to model the total mean end-to-end delays. The estimated percentile delays normalised to total serialisation delays are shown in Fig. 3.27(a). It is observed that the model over-predicts percentile delays in comparison to percentile delays measured from packet traces. However, the over-estimation using the mean delays from Lindley's recursion in Equation 3.11 for loads above 0.9, results in less over-estimation than using only mean delays from G/G/1 queuing model as shown in Fig. 3.27(b). The difference between estimated and measured percentile delays of 99% for a load of around 0.96 is 18.9 sec using the mean from the G/G/1 queuing model and the difference is 9.45 sec using the mean queuing delay from Lindley's recursion. Similarly, the difference is 35.33 sec and 21.06 sec for the percentile delays of 99.9%. So it can be concluded that predicting percentile delays for loads above 0.85 using means from Lindley's recursion presents a better upper bound than G/G/1 approximation only.

Note that network planning using under-predicted percentile delays can result in a higher packet loss ratio than the maximum threshold for the provisioning of different network functional splits and services, so the model ensured over-prediction.

3.5.1 End-to-End Percentile Delay Model Algorithm

The overall process of modelling percentile delays is summarized in the form of the following Algorithm 4. The process of modelling latency starts by collecting the network parameters listed in Table 3.4, which include the arrival rate of packets, $\lambda_{t,1}$, at aggregation nodes at the lowest level or RUs in Fig. 3.1, the average and standard deviation of packet lengths, $\overline{L_p}$, $\sigma_{\overline{L_p}}$, the output bit rate of all the nodes in the network $R_{o,i}$ and the standard deviation of arrival rate of packets $\sigma_{(a,1)}$.

Using the collected parameters, the mean waiting times are calculated using the G/G/1 queuing model and Lindley's recursions, $\bar{T}_{w,i}$ and $\bar{T}'_{w,i}$. The overall mean waiting times are calculated using both means from the G/G/1 model and Lindley's recursion depending on the load, $\overline{T_w}$. The total mean end-to-end delays, $\bar{T}_e$, are computed using the total serialisation, propagation and mean waiting times. The standard deviation of end-to-end delays, σ_t, is computed using the mean waiting times and standard deviation of service rates. From the mean and standard deviation, the PDFs of end-to-end delays, $f_{T_e}(t)$, are computed which leads to the calculation of percentile delays, $P_{0.99}$, $P_{0.999}$.

Algorithm 4 Steps for modelling of percentile delays of a path in fronthaul network

```
**Input:** $\overline{L_p}$,$\sigma_{\overline{L_p}}$,$\lambda_{t,1}$, $R_{o,i}$,$\sigma_{(a,1)}$.
**Output:** $P_{0.99}$, $P_{0.999}$.
Calculate end-to-end serialisation delays $\bar{T_s}$ of the path using Equation 3.13.
Set the total number of nodes in the path in the fronthaul network to $N$
**for** $i = 1,2,\ldots,N$ **do**
    Calculate the load $\rho_i$ and service rate of packets $\mu_{r,i}$ of the nodes in the path using Equation 2.6 and Equation 2.7.
    Estimate the coefficients for nonlinear approximation of standard deviation of the inter-arrival time of packets $\sigma_{a,i}$ using Equation 3.1.
    Calculate the coefficient of variation of the inter-arrival time of packets $C_a$, and the coefficient of variation of service rate of packets $C_r$ using Equation 2.8.
    Calculate the mean waiting times $\bar{T}_{w,i}$ for the nodes in the path using G/G/1 queuing mode using Equation 2.5.
    Calculate the mean waiting times $\bar{T}'_{w,i}$ for the nodes in the path using Lindley's recursion-based queuing model using Algorithm 1.
**end for**
Calculate the total mean waiting time of the path $\overline{T_w}$ using Equation 3.11
Calculate the total mean end-to-end delay $\bar{T_e}$ of the path using Equation 3.12
Calculate the standard deviation of total end-to-end delay $\sigma_t$ of the path using Equation 3.15
Model the PDF of the end-to-end delay $f_{T_e}(t)$ of the path using lognormal distribution using Equation 3.16.
From the modelled PDF of the end-to-end delay $f_{T_e}(t)$ of the path in the fronthaul network, calculate the percentile delays $P_{0.99}$, $P_{0.99}$ using Equation 3.19
```

3.6 End-to-End Percentile Delay Model Deployment Results

Using the fitness of the model for case-1, it was tested for cases-2 to case-5 in Table 3.1 to check its accuracy. The predicted normalised percentile delays for cases (2∼5) are shown in Fig. 3.28 to Fig. 3.31. From the observation of Fig. 3.28 to Fig. 3.31 it can concluded that the model over-estimates the percentile delays and therefore can be considered as an upper bound on the estimates for 99 and 99.9 percentile delays.

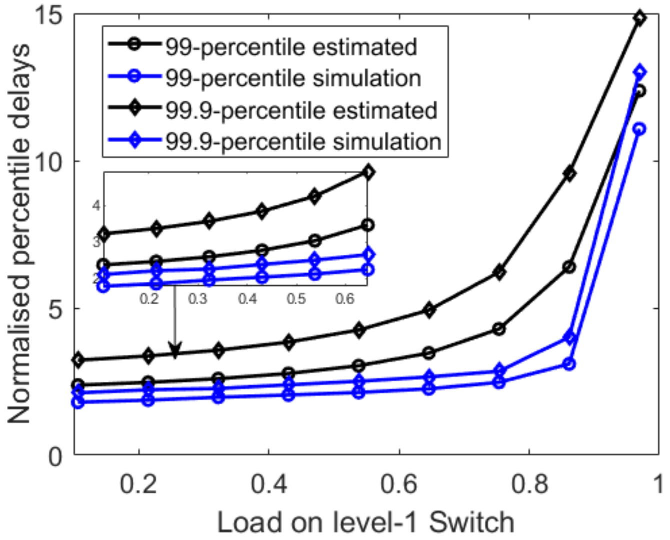


Fig. 3.28: Predicted and measured percentile delays vs load on switch-1 (case-2).

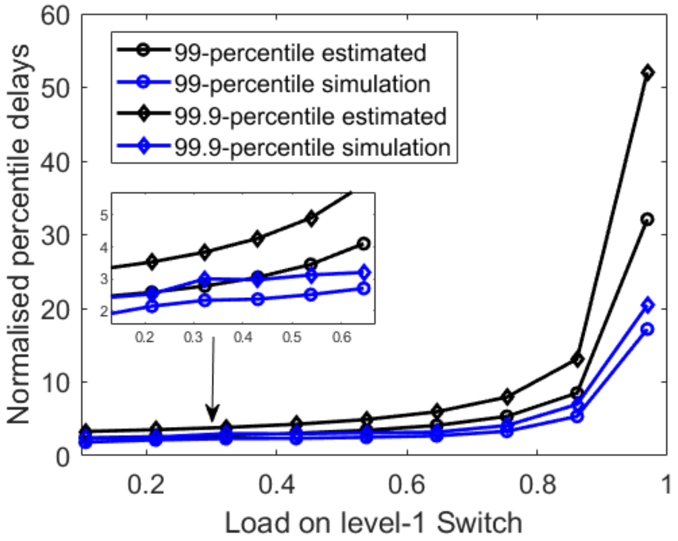


Fig. 3.29: Predicted and measured percentile delays vs load on switch-1 (case-3).

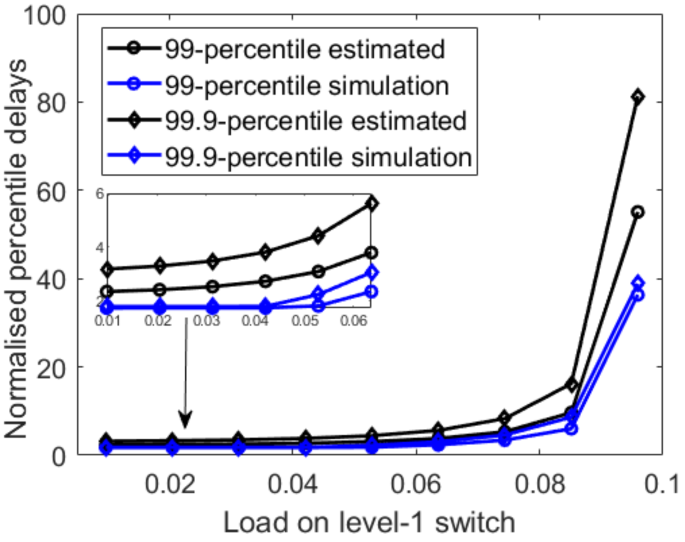


Fig. 3.30: Predicted and measured percentile delays vs load on switch-1 (case-4).

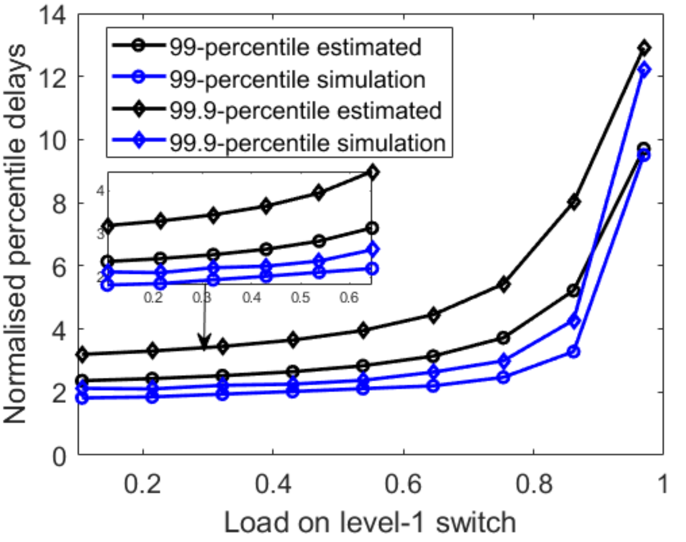


Fig. 3.31: Predicted and measured percentile delays vs load on switch-1 (case-5).

The accuracy of the prediction in Fig. 3.28 to Fig. 3.31, was analyzed in terms of error between measured (blue plots in Fig. 3.28 to Fig. 3.31) and predicted delays (black plots in Fig. 3.28 to Fig. 3.31). The errors were normalised to measured percentile delays and are shown in Fig. 3.32 and Fig. 3.33. From the analysis of Fig. 3.32, it can be observed that the normalised error for 99th percentile delays does not follow a specific trend, yet the errors are positive. The percentage error, in this case, ranges from 2% to 100% in Fig. 3.32. Similarly, for 99.9 percentile delays, the errors for cases (1∼5) are shown in Fig. 3.33. The errors are significantly higher than the 99 percentile delays for all simulation cases and follow almost the same trend as for 99 percentile delays. The errors are also positive in this case. The percentage error, in this case, ranges from 8% to 150% in Fig. 3.33. Since there are no negative errors or under estimations for both 99 and 99.9 percentile delays, the model presented in this paper can be considered as providing an upper bound on the predictions for percentile delays.

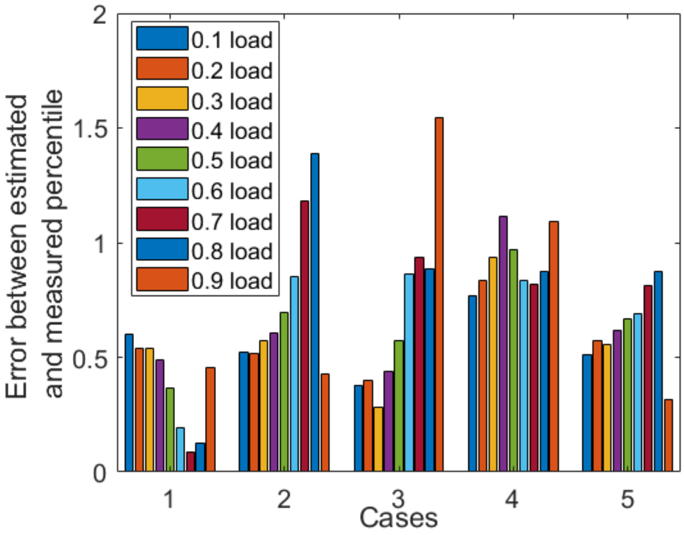


Fig. 3.32: Error between estimated and measured 99 percentile delays.

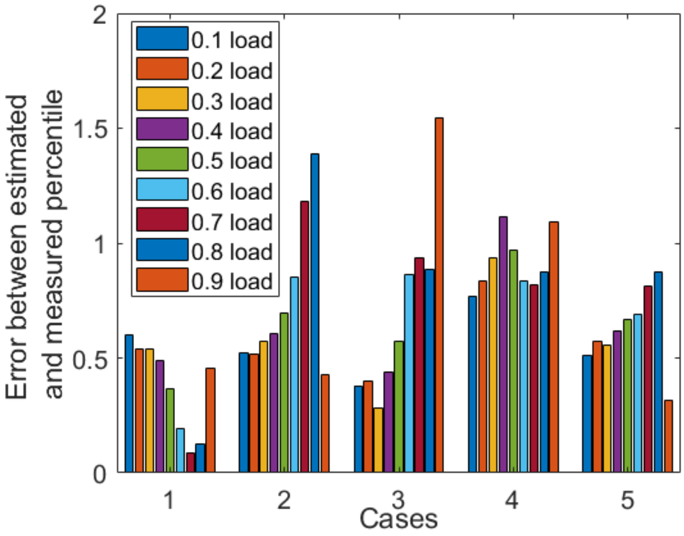


Fig. 3.33: Error between estimated and measured 99.9 percentile delays.

3.7 Conclusions

A model was developed for predicting the end-to-end percentile delays for a packet network similar to a fronthaul, using the network parameters of loads, input arrival rate, packet service rate, statistics of inter-arrival times of packets and service time of packets. The estimates of percentile delays also required the modelling of the mean delays and delay distributions of the individual nodes in the end-to-end path. Different packet network scenarios were simulated to test the model. The scenarios comprised the same and mixed input link bit rates with different variations of loads in the links.

From the simulation results it can be concluded that, for the load above 0.85, Lindley's recursion-based estimation proved to be a better upper bound on the estimation of mean waiting times. For the estimates of percentile delays, it can be concluded from the simulation results, that the combination of using G/G/1 and Lindley's recursion proved to be a more accurate upper bound for the percentile delay estimates than using G/G/1 approximations only. The model did not underpredict the percentile delays when deployed in different network configurations, so it can be concluded that the model can be considered a better upper bound on the percentile estimates for a broader range of network configurations.

The predicted percentile delays were limited to 99% and 99.9% due to the limitations of time and memory resources. The results can be extended to higher percentiles i.e., 99.999% using the model proposed. However, testing these cases will require very long simulation times and very large packet traces to gather enough statistics.

In the simulation cases for the testing of the end-to-end delay, the packet lengths were random with a Normal distribution. For networks with different statistics of packet lengths and inter-arrival times of packets, the model can be easily modified by using relevant PDFs in Lindley's recursions for the estimation of mean waiting times.

References

[1] ‘Study on scenarios and requirements for next-generation access technologies.’, European Telecommunications Standards Institute, Technical specification 38.913 (Release 14). Accessed: May 04, 2024. [Online]. Available: lhttps://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2996

[2] G. Pocovi, K. Pedersen, and P. Mogensen, ‘Joint Link Adaptation and Scheduling for 5G Ultra-Reliable Low-Latency Communications’, IEEE Access, vol. PP, pp. 1–1, May 2018, doi: 10.1109/ACCESS.2018.2838585.

[3] G. O. Pérez, J. A. Hernández, and D. Larrabeiti, “Fronthaul Network Modeling and Dimensioning Meeting Ultra-Low Latency Requirements for 5G,” J. Opt. Commun. Netw., vol. 10, no. 6, p. 573, Jun. 2018, doi: 10.1364/JOCN.10.000573.

[4] N. Molner, A. de la Oliva, I. Stavrakakis, and A. Azcorra, “Optimisation of an integrated fronthaul/backhaul network under path and delay constraints,” Ad Hoc Networks, vol. 83, pp. 41–54, Feb. 2019, doi: 10.1016/j.adhoc.2018.08.025.

[5] M. Klinkowski, “Optimisation of latency-aware flow allocation in NGFI networks,” Computer Communications, vol. 161, pp. 344–359, Sep. 2020, doi: 10.1016/j.comcom.2020.07.044.

[6] G. O. Pérez, D. L. López, and J. A. Hernández, “5G New Radio Fronthaul Network Design for eCPRI-IEEE 802.1CM and Extreme Latency Percentiles,” IEEE Access, vol. 7, pp. 82218–82230, 2019, doi: 10.1109/ACCESS.2019.2923020.

[7] A. Anand, G. de Veciana, and S. Shakkottai, ‘Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks’, IEEE/ACM Transactions on Networking,

[8] K. Krishnamoorthy, Handbook of Statistical Distributions with Applications, 2nd ed. New York: Chapman and Hall/CRC, 2015. doi: 10.1201/b19191.

[9] P. Schulz, ‘Queueing-Theoretic End-to-End Latency Modeling of Future Wireless Networks’, PhD thesis, Technische Universität Dresden, Dresden, 2020. [Online]. Available: https://d-nb.info/1227833598/34

[10] M. Pavlovic, ‘Quantile function of the log-normal distribution’, The Book of Statistical Proofs. Accessed: Nov. 08, 2023. [Online]. Available: https://statproofbook.github.io/P/lognorm-qf.html

Chapter 4

Fronthaul Network Model and Simulation Results

4.1 Introduction

The Chapter discusses the simulation results of using DRL for resource allocation in fronthaul for beyond-5G mobile networks with Ethernet and FlexE-enabled aggregation nodes and user allocation managed by CoMP. It describes the flow of traffic, with UEs connected to RUs, and discusses the use of DRL agents to optimise slot allocation and user offloading to maximize network throughput. It demonstrates the advantage of using DRL for user offloading from a path of higher latency to a path of lower latency for overall user throughput enhancement.

The overall fronthaul network architecture for the beyond-5G mobile network for simulations is outlined in Section 4.2. Additionally, Section 4.2.3 briefly touches upon the Coordinated Multi-Point (CoMP) beamforming function within the RAN. The Section also highlights the impact of fronthaul latency on overall user throughput. The latency in the model is calculated via the model presented in Chapter 2 and Chapter 4. The details of the implementation of the fronthaul network model in Simulink MATLAB® are discussed in Section 4.3.

The training of the FlexE agent dedicated to traffic prediction and slot allocation in the FlexE shim is described in Section 4.4. The impacts of using different reward functions and hyperparameters of reinforcement learning on the training of the FlexE agent are discussed in Section 4.4. Similarly, the training of the agent responsible for user allocation is discussed in Section 4.5. The Section highlights the enhancement of overall user throughput when the controller agent is deployed for user allocation from a path of high latency to a path of lower latency.

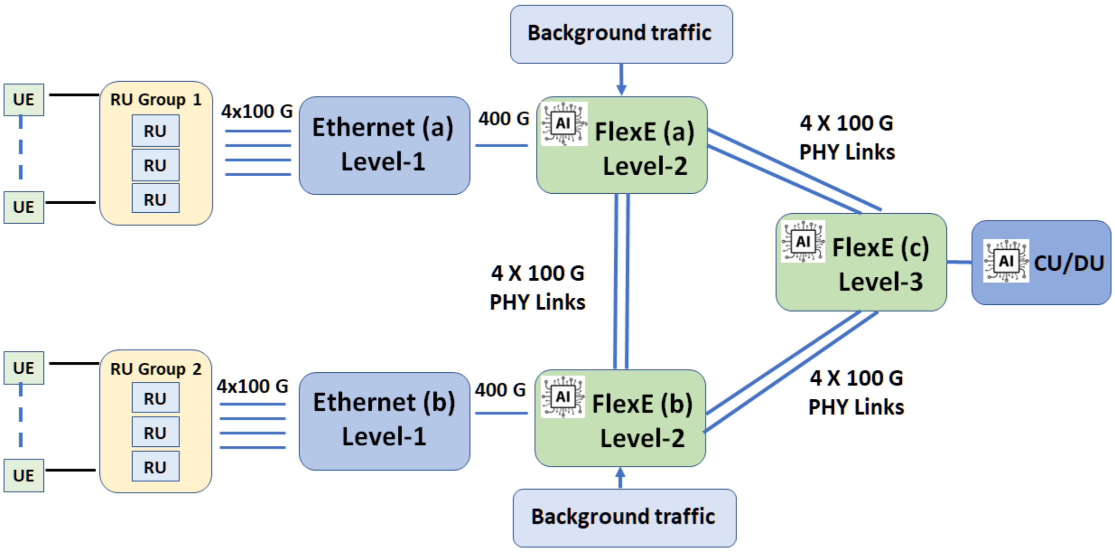


Fig. 4.1: Network architecture for DRL-based optimisation of fronthaul.

4.2 Fronthaul Network Model

For the research into deploying DRL in fronthaul for beyond-5G mobile networks, a combined RAN/Fronthaul network is modelled and shown in Fig. 4.1. The fronthaul network comprises Ethernet and FlexE aggregation nodes. In the network in Fig. 4.1, UEs are connected to the RUs, grouped in a set of three, assuming they are collocated. The traffic from the UEs is packetized in each RU and then transported over the fronthaul. The link between the RU groups 1,2 in Fig. 4.1 is linked to Ethernet switches via 4x100G PHY links. These are labelled as Ethernet (a,b) at the level-1 aggregation node in Fig. 4.1. Ethernet-based aggregation was used at level-1, assuming that the RUs within the RU-group in Fig. 4.1 are collocated and that RU traffic is aggregated by Ethernet switches for transport over the fronthaul links. Although FlexE-based aggregation can also be used at level-1 for efficient resource and bandwidth allocation in the physical layer, due to the variability of user bit rates, Ethernet-based aggregation was chosen to reduce the complexity of the DRL system within the computational limits of the systems used for simulations.

The traffic from the level-1 aggregation is then aggregated by the FlexE aggregation nodes labelled as FlexE (a,b) at level-2 aggregation. The FlexE aggregation nodes aggregate traffic from the users as well as background traffic from other sources in the network. The traffic from the level-2 aggregation nodes is aggregated by a FlexE aggregation node at level-3 labelled as FlexE (c) in Fig. 4.1. The final destination of the traffic in the uplink is the CU/DU connected to the FlexE aggregation node at level 3. The total traffic from UEs

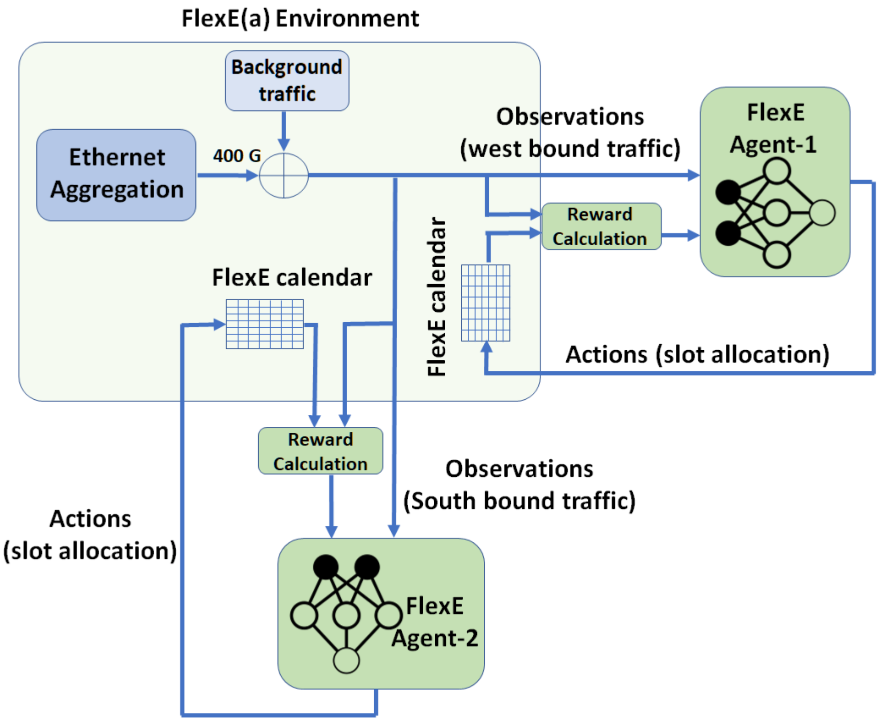


Fig. 4.2: Agents and environment in FlexE(a) node in Fig. 4.1.

linked to RU groups can thus be switched or routed to the CU/DU via different paths in the fronthaul.

4.2.1 Environment, Observation and Actions for FlexE Agent

The conceptual diagram of the FlexE aggregation nodes in Fig. 4.1 is shown in Fig. 4.2. Within each FlexE aggregation node of Fig. 4.1, there is a separate calendar for traffic flows in different directions. Each sub-calendar is managed by a separate DRL agent termed a ‘FlexE agent’ The main objective of the FlexE agents is to predict the incoming user and background traffic and allocate slots such that there are the least under and over-allocations of slots to the input traffic. The FlexE agents in Fig. 4.2 receive the traffic from the Ethernet aggregation nodes and the background traffic from other sources in the network. The FlexE agent discretizes the input traffic with intervals of 5 Gbps, which forms the input observation space and is defined as follows,

$$R_i \in \{5, 10, 15, ..., 400\}, \tag{4.1}$$

where R_i is defined as follows,

$$R_i = R_{Eth} + R_{Back}, \tag{4.2}$$

where R_{Eth} is the total user traffic from an RU group at the output of level-1 Ethernet aggregation in Fig. 4.1 and the R_{Back} is the background traffic added to the FLexE at level-2 aggregation in Fig. 4.1. The reward function used for the training of the FlexE agents is defined as follows,

$$R_{FlexE} = \begin{cases} -20 + ((S - R_i) - 20) & \forall\ 0 \leqslant S - R_i \leqslant\ 20 \\ -1000\left(1 - e^{-1*(S-R_i)}\right) & \forall\ S - R_i > 20 \\ -1000\left(1 - e^{(S-R_i)}\right) & \forall\ S - R_i < 0 \end{cases} \tag{4.3}$$

where R_i is the observed input traffic demand defined in Equation 4.2 and S is the bit rate allocated in increments of 5 Gbps in the FlexE calendar. The action space, S, for the FlexE agent, can be defined as follows,

$$S \in \{5, 10, 15, ..., 400\} \tag{4.4}$$

The actions and observations are updated every 4ms which is the delay for the reconfiguration of the FlexE calendar. The slot allocations for the incoming traffic are calculated using Deep-Q learning from past actions and rewards in an experienced buffer. Deep-Q learning was used since it observes input traffic both from RAN and other sources in the network. This traffic has large variations making it impossible for conventional Q-table-based reinforcement learning agents to relate observations to actions. In this way, the FlexE agent pre-configures the calendar in the aggregation node, to help reduce any buffering delay of packets due to the under-allocation of 5Gbps slots, while maintaining efficiency.

4.2.2 Environment, Observation and Actions for Controller Agent

Within the CU/DU a second DRL agent termed a ‘controller agent’ is trained and deployed using deep-Q learning for the allocation of users to the two RU groups based on the latency of their paths to the CU/DU. The controller agent carries out the offloading of the users linked to the RU groups such that the total bit rate of the users in the RU groups is maximized. The conceptual block diagram of the CU/DU is shown below in Fig. 4.3. The controller agent receives the total bit rate of the two RU groups as an input and then takes actions which are the number of users to be offloaded from the two RU groups. Therefore the observation space for the controller agent can be defined as follows,

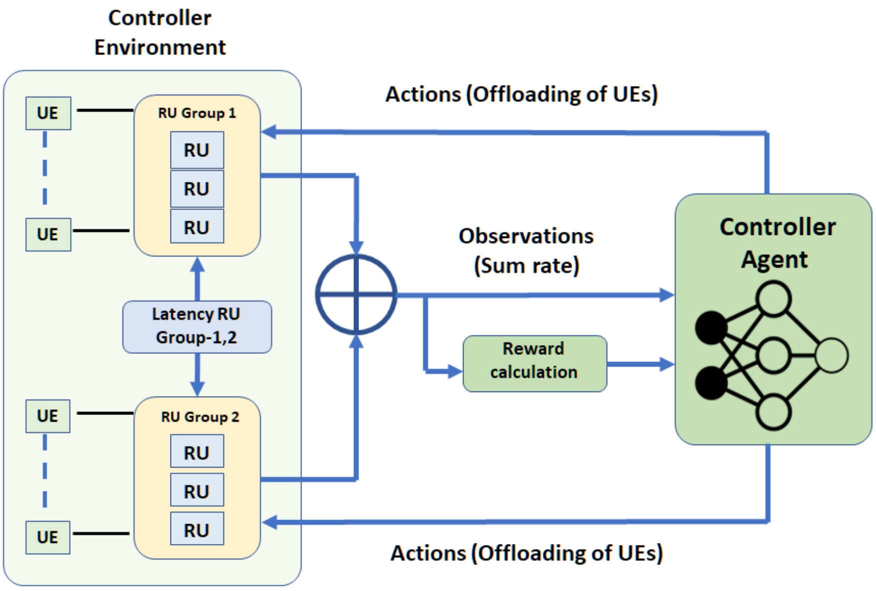


Fig. 4.3: Agent and environment for CU/DU in Fig. 4.2.

$$R_{sum} = R_{ZF_1} + R_{ZF_2}, \tag{4.5}$$

where, R_{sum} is the total user bit rate of the two RU groups, where R_{ZF_1} and R_{ZF_2} are the user bit rates of the two RU groups calculated from the CoMP model discussed in Section 4.2.3. The reward function for the controller agent resulting in monotonically increasing reward is given as follows,

$$R_{ctr} = 500\left(\bar{R}_{sum}\right)^2 + \left(-5\times10^5\right)\left(\bar{R}'_{sum}\right), \tag{4.6}$$

where, $\bar{R}_{sum}$ is the average value, over 5 samples, of the total bit rate for the two RU-groups defined in Equation 4.5. $\bar{R}'_{sum}$ is the first-order difference of the average bit rate in consecutive timesteps of 4ms, indicating the variations in the total bit rate of the RU groups and given as follows,

$$\bar{R}'_{sum} = \bar{R}_{sum}(t) - \bar{R}_{sum}(t-0.004). \tag{4.7}$$

The first-order difference term, $\bar{R}'_{sum}$, in the reward function in Equation 4.6 was introduced to reduce the frequent redundant offloading among RU groups. However, the penalty due to the rate variation is far less than the positive reward from the first term, $\bar{R}_{sum}$, in Equation 4.6. If the path between the RU and the CU/DU is heavily loaded, then it will cause large delays in the CSI from the UEs to reach the CU/DU. This has the impact of reducing the total bit rate of users in the two RU groups. To circumvent this the controller agent after training attempts to offload users from a path of high load to a path of low load. The actions of the controller agent are the offloading commands for each RU group. In any training step, the number of users offloaded was limited to 5 users. This resulted in a monotonically increasing average reward earned by the controller agent during the training with training results mentioned in Section 4.5. The action space for the controller agent can then be defined as follows,

$$U_{off} \in \{0,1,2,3,4,5\}, \tag{4.8}$$

where, U_{off}, is the discrete number of users that can be offloaded from one RU group to another. Similar to the FlexE agent, the controller agent was trained using deep-Q learning due to the large observation space of the RU bit rate and its variations. This would help the agent, to generalise the transition from states to actions, resulting in the reduction of the memory resources required to map states to actions as discussed in Section 2.7.2.

4.2.3 CoMP and Latency Model

The users in the network model are linked to the RUs via CoMP since it will be used in the future 5G and beyond networks for effective interference minimisation and throughput maximisation of the users. CoMP relies on sending the CSI to the network controllers. The performance of the CoMP then becomes highly dependent on the delay of the path via which CSI is fed back to the network controllers. The fronthaul network, presented in the previous Section, will comprise FlexE and Ethernet-enabled aggregation nodes. The delays in the RUs network interface cards, FlexE and Ethernet nodes comprising mainly serialisation and queuing delays, will add to the delay in the feedback being sent to the network controllers.

For generating user traffic it is assumed that each user is equipped with a single antenna and beamforming vectors calculated via zero-forcing (ZF) to reduce interference between beams generated by the RUs. In this case, the achievable user bit rate was calculated from a pre-developed model for CoMP beamforming. In this model, in the uplink, the total achievable user bit rate, R_{ZF}, for users, k, linked to an RU, calculated using the signal-to-interference noise ratio (SINR) of a single user [2] and the Shannon capacity model is given as follows [3],

$$R_{ZF} = \sum_{k=1}^{K} log_2 \left(1 + \frac{|\left(\sqrt{\rho}\hat{h}_k^T + \sqrt{\sigma_{\varepsilon,k}^2 + \sigma_{\eta,k}^2}\right) f_k|^2}{\sum_{i \neq k} |\left(\sqrt{\rho}\hat{h}_k^T + \sqrt{\sigma_{\varepsilon,k}^2 + \sigma_{\eta,k}^2}\right) f_i|^2 + \sigma_n^2} \right), \tag{4.9}$$

where the channel estimate, $\hat{h}(t)$, is altered, given as follows,

$$\hat{h}(t) = \rho \hat{h}(t-D) + \varepsilon(t), \tag{4.10}$$

where, $\rho = J_0(2\pi f_D D)$ is the channel correlation vector affected by user mobility, and, $\varepsilon(t)$ is the error vector, with variance, σ_ε^2, given as follows,

$$\sigma_\varepsilon^2 = \sigma_{\hat{h}}^2 \left(1 - |\rho|^2\right). \tag{4.11}$$

Note that the factor, D, in Equation 4.10 represents the latency in fronthaul calculated from the model for latency in Chapter 3. The noise term, $\sigma_{\eta,k}^2$, for a user k in Equation 4.9 has a complex normal distribution dependent on the length of the pilot symbols, data symbols and Doppler frequency [4]. The terms, f_K, and, f_i, are the beamforming vector for users assuming zero-forcing equalisation defined in [4].

The traffic generated by users is first received at RUs that add encapsulation overheads dependent on the split option being used. Further overheads are added by the Ethernet and FlexE-based aggregation nodes in fronthaul. The traffic increase due to the encapsulation was ignored in the analysis. It was assumed that an intra-PHY functional split option 7.2 in fronthaul was used which resulted in upscaling or an increase in the user data rate [5]. The functional split upscaling factor, F_{split}, is calculated using the relation given as follows [7],

$$F_{split} = \frac{2 \times N_{ant} \times N_{bit} \times \frac{N_{sc}}{T_s}}{N_{ant} \times log_2(M) \times BW}, \tag{4.12}$$

where, N_{ant} is the number of antenna elements, N_{bit} is the sampling width (no of bits per sample), N_{sc} is the number of data subcarriers comprising the OFDM symbol, M is the modulation order, BW is the bandwidth of the signal transmitted by a UE and, T_s is the OFDM symbol length in time. As mentioned previously the encapsulation overheads were also ignored in the analysis and Equation 4.12.

The numerator in Equation 4.12 indicates the bit rate of fronthaul for the functional split option of 7.2, where data is transmitted in the frequency domain and depends on the number of antenna layers, the bits per symbol, the number of subcarriers and the time period of an OFDM symbol. The denominator in Equation 4.12, indicates the user bit rate in the RAN, which depends on the number of antennae, the bandwidth of each user and bits per symbol for a QAM modulation level. The ratio of the two quantities will give the upscaling factor of the user bit rate in the fronthaul where the functional split factor of 7.2 is deployed.

For calculating the end-to-end latency of the path between an RU and CU/DU in the uplink in Fig. 4.1, the mean waiting times were calculated using only the G/G/1 queuing model in Equation 2.5, discussed in Section 2.5.1. The total mean end-to-end delay, $\bar{T}$, of a path in the fronthaul is given as follows,

$$\bar{T} = T_{s,RU} + T_{s,E} + 2\,T_{s,F} + T_{s,F} + T_{q,RU} + T_{q,E}, \tag{4.13}$$

where $T_{s,RU}$, $T_{s,E}$ and $T_{s,F}$ are the fixed serialisation delays in the RU's Ethernet interface, Ethernet aggregation node and two FlexE aggregation nodes in a path in fronthaul. $T_{q,RU}$ and $T_{q,E}$ are the waiting times in the RU's Ethernet interface and Ethernet aggregation node of the path in fronthaul using only the G/G/1 queuing model in Equation 2.5. The distribution of the total delay of a path is modelled by a log-normal distribution using the total mean delay, $\bar{T}$ (as discussed in the Section 3.5).

4.3 Simulation Setup and Parameters

The overall RAN/fronthaul model discussed in Sections 4.2 and 4.2.3, was implemented in Simulink MATLAB® using the reinforcement learning toolbox. The snapshot of the implemented Simulink MATLAB® model is shown in Fig. 4.4. Two agents were implemented in Simulink MATLAB® for traffic predictions for slot allocation, which are labelled 'FlexEAgent2N' and 'FlexEAgent2E' in Fig. 4.4. One controller agent was implemented for offloading UEs from the path of higher latency to a path of lower latency to maximize the total bit rate of the system labelled as 'ControllerAgent' in Fig. 4.4. The agents were linked to an environment which contained the pre-implemented CoMP beamforming model. The environment is labelled as labelled as 'FlexEControllerEnv' in Fig. 4.4. The outputs of the environment are the observations and rewards, which are fed as the inputs to the three FlexE agents. The observations are, the total traffic, user and background traffic, in the two paths in fronthaul and the total user bit rate of the two RU groups in Fig.4.1 labelled as 'Observation2N', 'Observation2E' and 'ObservationCA' respectively in Fig 4.4. The rewards for the DRL agents are also calculated in the environment block in Fig. 4.4, which are fed as input to the DRL agents. The rewards are labelled as 'Reward2N', 'Reward2E' and 'RewardCA' respectively in Fig. 4.4. The inputs to the environment were the actions from the two FlexE agents labelled as 'Actions2N' and 'Actions2E' which are the predictions of the input traffic from FlexE agents with a granularity of 5 Gbps slots. The third input is labelled as 'ActionCA', which is the number of users to be offloaded from one path to another in Fig. 4.1.

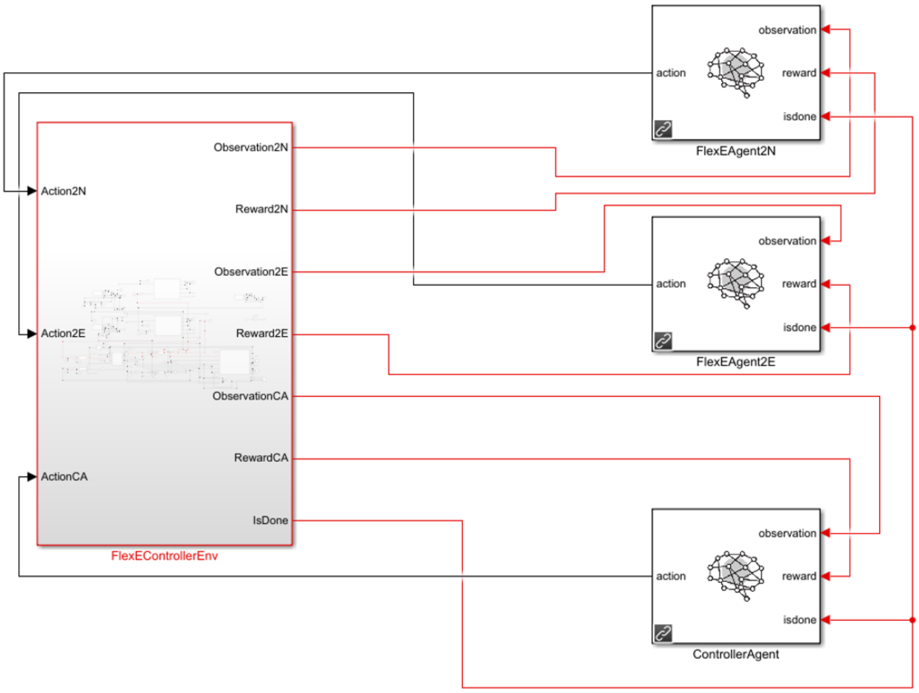


Fig. 4.4: Simulink model of fronthaul with reinforcement learning agents.

4.3.1 RAN and CoMP Model Parameters

The parameters for the wireless system are listed in Table 4.1. The CoMP model was initialised by assuming 15 users in each RU group (so, a total of 30 users in the two groups). Each RU was modelled to transport 16 independent layers/beams of user data in fronthaul. The locations of users were uniformly distributed over the 200×200m grid, shown in Fig. 4.5, to maintain a balance of achievable bit rates between the RU groups. The locations of RUs were in a hexagonal formation with a radius of 50m so that the RUs were equidistant from each other and on average equidistant from UE locations. The signal-to-noise ratio (SNR) for each user was set at 13 dB in Table 4.1, which aligns with an AWGN channel, suitable for the initial design but likely unrealistic. A power constraint per RU of 40 W reflects real-world limitations and allows for power-performance trade-off analysis in Table 4.1. A bandwidth of 50 MHz per user allows for adequate data transfer rates, supporting various applications. User mobilities at 6 m/s suggest consideration for dynamic scenarios where users might be in motion. The rate of arrival and departure, at 10^{-6} and 4×10^{-6} respectively, denotes the frequency at which users join and leave the network, impacting system dynamics in Table 4.1.

Table 4.1: Simulation Parameters for RAN.

Number of users per RU-group	15
Avg bit rate of RU-groups	117.04 Gbps
Avg background traffic	110.7 Gbps
Number of RUs Groups	2
Number of RUs per Group	3
SNR per User	13 dB
Antennas per RU	16
Power Constraint per RU	40 W
Bandwidth per user	50 MHz
User Mobilities	6 m/s
Rate of Arrival	10^{-6}
Rate of Departure	4×10^{-6}

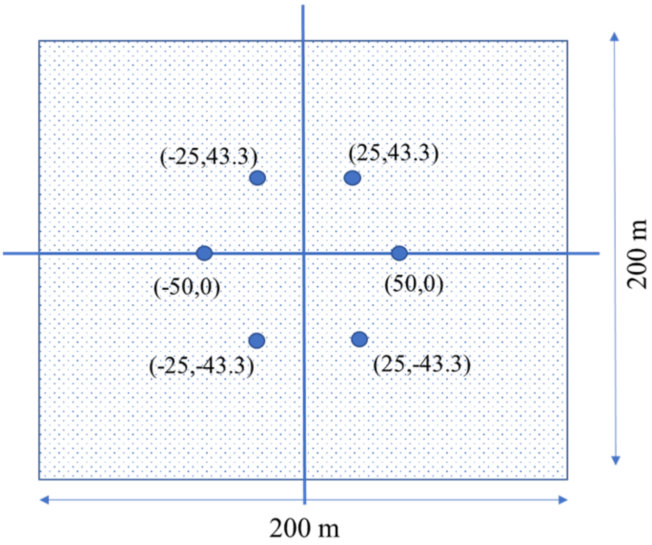


Fig. 4.5: RU coordinates in x,y plane (Case-1).

Table 4.2: Functional split upscaling parameters.

Number of antenna elements N_{ant}	15
Number of bits per symbol N_{bit}	8
Number of subcarriers N_{sc} in 50 MHz	250×12
Modulation level (M)	16
Interval of the OFDM symbol T_s	66.7 μs
Bandwidth per user	50 MHz
Functional split upscaling factor F_{split}	3.5
Total upscaling factor for user bit rates	10.5

Equation 4.12, was used to calculate the increase or upscaling of the user bit rates in fronthaul. For the input parameters in Equation 4.12, it was assumed that the model in Fig. 4.4 utilised a subcarrier spacing of 15 kHz, with the modulation level of 16-QAM, 8 bits per sample for sampling of the user data symbols, with the parameters listed in Table 4.2. It was also assumed that the 50 MHz spectrum could accommodate approximately 250 resource blocks, where each resource block had 12 subcarriers. Using the above-mentioned parameters in Equation 4.12, the upscaling factor of the user bit rate was found to be approximately 3.5. Since it was assumed that each RU group comprised 3 RUs, the upscaling factor was further multiplied by a factor of 3 resulting in a total upscaling factor of approximately 10.5 as listed in the last row of Table 4.2. The total upscaling factor of 10.5 in Table 4.2, will be applied to upscale the user bit rates generated by the CoMP model using Equation 4.9 in the simulations.

4.3.2 Fronthaul Traffic Parameters

Equal background traffic of 110 Gbps was added to the FlexE aggregation nodes at level 2 by the ‘Background traffic blocks’ as shown in Fig. 4.1. The traffic was added to the user traffic from RU groups 1 and 2 as shown in Fig. 4.1. The addition of background traffic resulted in a balanced impact of latency in the two paths. The total achievable bit rate for both RU groups was found to be 22.3 Gbps. This data rate was upscaled by a factor of 3.5 using Equation 4.12 and functional split parameters from Table 4.2. Further increased by a factor of 3 was carried out to accommodate the fact that there are three RUs in an RU group. This increase resulted in a load of 234.089 Gbps in the fronthaul.

The underlying background traffic pattern was extracted from the Widely Integrated Distributed Environment-Measurement and Analysis on the WIDE Internet (WIDE-MAWI) project packet traces repository [6] specifically for the date of 08/04/2020. WIDE MAWI project is an online repository of packet traces that have been collected by monitoring

Table 4.3: Queuing Delay Parameters.

Average packet length	735 bytes
Standard deviation of packet length	300 byes
Output links rate	400 Gbps
Standard deviation of inter-arrival time of packets	$300\mu s$
Distance of optical links	1 km

and measuring the traffic from a backbone optical network operated in Japan. The traces contain information on packets such as packet types, packet header and data lengths, packet source and destination IP addresses, and packet arrival and departure times at the node. The average bit rate of the traffic was 6 Gbps which is presumed less than the expected aggregate background and user traffic in beyond 5G fronthaul networks. Therefore, the traffic patterns were increased by multiplication with discrete factors (5∼11) to emulate a high load on the fronthaul aggregation nodes.

4.3.3 End-to-End Fronthaul Latency Parameters

The parameters for modelling the mean waiting times in the end-to-end delay in fronthaul, using the G/G/1 queuing model are listed in Table 4.3 and are similar to the parameters listed in Table 3.4. It was assumed that the average packet length was 735 bytes. It was also assumed that the length of Ethernet frames had a Normal distribution with a standard deviation of 300 bytes. The inter-arrival times of packets at source RUs had a Normal distribution with a mean set to give the required load and standard deviation of $300\mu s$. Normal distribution was selected since its standard deviation is independent of mean interarrival times, which were varied to vary the load.

4.4 FlexE Agent Training Parameters and Results

The objective of the research is bandwidth allocation optimisation as well as transmitter power saving through prediction of the traffic in fronthaul using DRL and allocation of resources in terms of 5 Gbps slots in a FlexE shim. Specifically for the context of traffic prediction and slot allocation, the DRL agent training aimed to allocate slots following the input traffic demand. Over-allocation can result in bandwidth wastage on the physical layer (PHY) links, and under-allocation can result in extremely high loads on FlexE aggregation nodes. As discussed in Chapter 3 for modelling of end-to-end latency, extremely high loads can result in extremely high end-to-end latency, which can result in the outage of

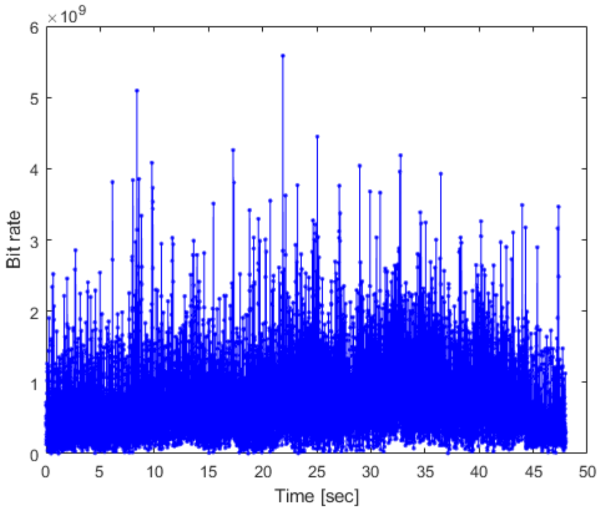


Fig. 4.6: Sample of bit rates of WIDE MAWI project packet trace.

different services of 5G network slices and the outage of different functional splits deployed in fronthaul [8].

In a DRL system, the design of reward functions is crucial and presents a significant challenge. Inadequately designed reward functions can lead the DRL system to converge to a local minimum, which may not yield the desired performance, such as in the form of high over and under-allocation of bandwidth in fronthaul. This issue is particularly relevant when training a DRL-based agent on real-world traffic patterns, which can have abrupt transitions or high deviations from the mean. For instance, consider a traffic pattern from the WIDE MAWI project repository, as shown in Fig. 4.6. If the reward function is not carefully designed, training a DRL-based agent on such a traffic pattern for prediction may not increase the rewards collected during the training process. This may lead to high over and under-bandwidth allocation if agents are deployed after the training process. To address this, the reward function should stimulate the agent with positive rewards for equal and some over-prediction as well as negative rewards for under-prediction.

Initially, an arbitrary traffic pattern was generated for the development of a reward function for traffic prediction and slot allocations by the FlexE agent. This traffic pattern was generated by a source that produced normally distributed random numbers with a mean of 300 and a standard deviation of 5000. The traffic pattern was then filtered using a raised cosine filter with a roll factor of 0.5, as shown in Fig. 4.7. The output traffic pattern is depicted in Fig. 4.8. Compared to real-world traffic patterns, the traffic pattern in Fig. 4.8 did not have sharp transitions, which facilitated the increase in rewards collected by the FlexE agent with every episode and ultimately the convergence of the rewards. However, the aim

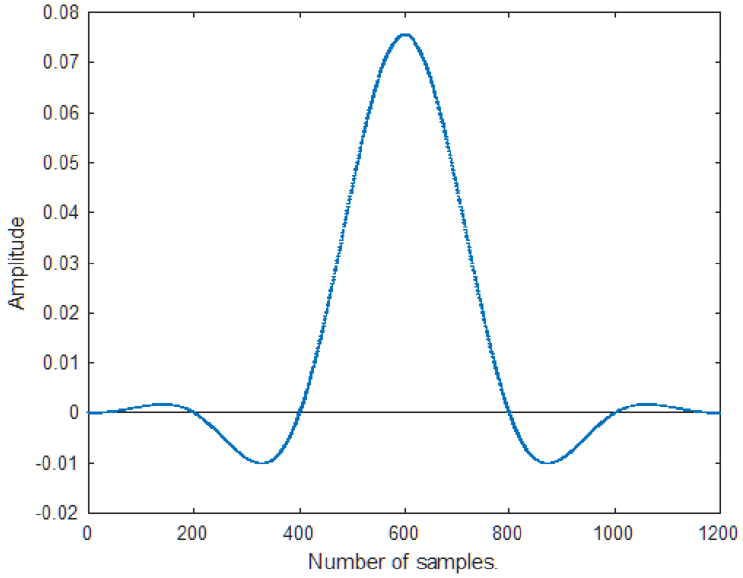


Fig. 4.7: Impulse response of interpolation filter.

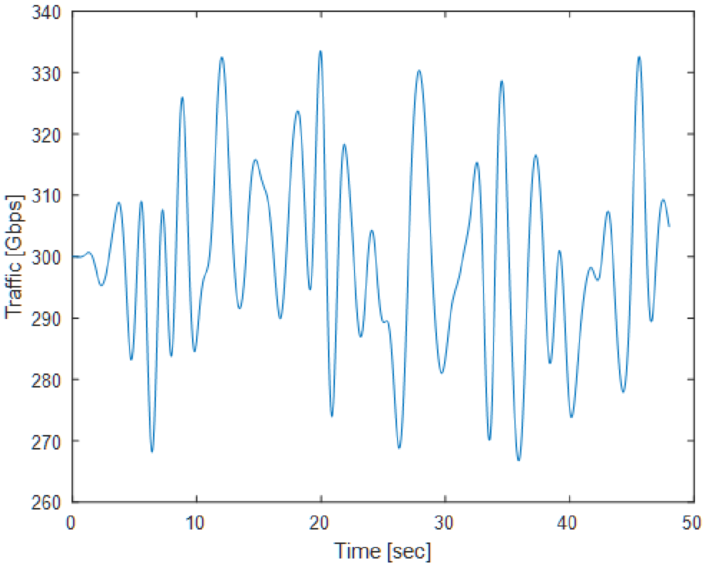


Fig. 4.8: Filtered input traffic pattern for the training of FlexE agent.

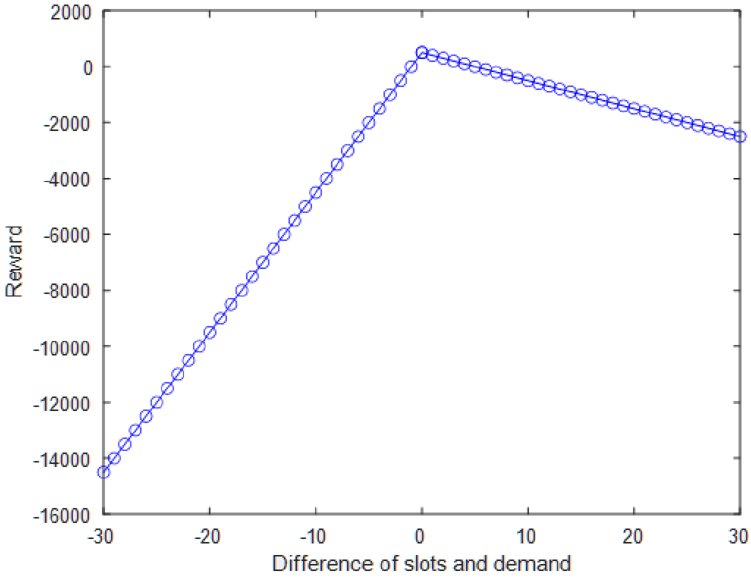


Fig. 4.9: Reward function for FlexE agent.

was to optimise the parameters of deep-Q learning for traffic patterns with higher degrees of variations from the mean.

Different reward functions and parameters of deep-Q learning were tested such as mini-batch size of the experience buffer, number of steps to look ahead, and discount factor. The reward function for the training of the FlexE agent is given as follows,

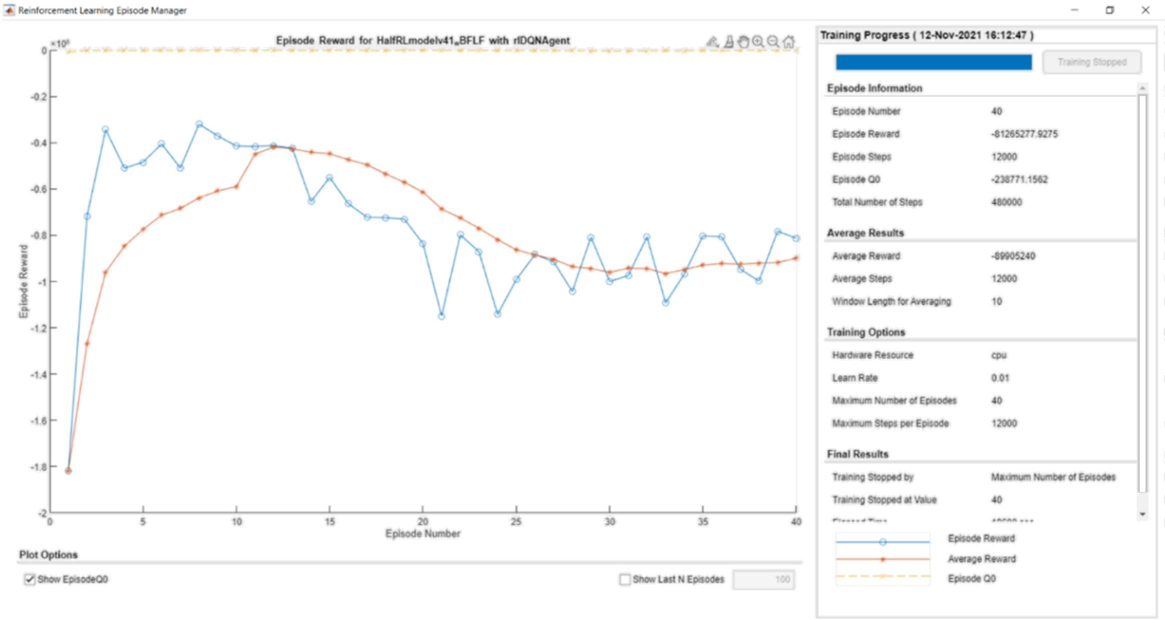
$$R_{FlexE} = \begin{cases} -100*((S-R_i)-5) & \forall \; S-R_i > 0 \\ -500+500*((S-R_i)) & \forall \; S-R_i \leqslant \; 0 \end{cases} \tag{4.14}$$

where R_i is the traffic demand in continuous form and S is the number of slots allocated in increments of 5 Gbps. The reward function was set up such that if the difference between the slots allocated and traffic demand exceeds 5 Gbps, the agent gets negatively rewarded. Similarly, if the difference was negative, then the FlexE agent also got negatively rewarded, but the decrement in reward is five times higher when the difference is positive. The reward function is plotted in Fig. 4.9

The parameters of the deep-Q learning for the training of the FlexE agent are also given in Table 4.4. The agent was trained for 40 episodes, where each episode spanned over 48 seconds and the step size in an episode was 4ms, corresponding to the reconfiguration time of the FlexE calendar [10]. The performance of the DRL was monitored by the rewards collected during the training process. The rewards collected in each training episode are given in Fig. 4.10 extracted from the reinforcement learning toolbox in MATLAB®. From the observation

Table 4.4: Deep-Q network parameters for FlexE agent training.

Learning Rate	0.01
Gradient Threshold	1
Number of Hidden Layers	6
Number of Neurons per Layer	98
Sample Time	4×10^{-3}
Experience Buffer length	30×6000
Mini batch size	128
Epsilon	1
Epsilon Decay factor	0.005
Number of steps to look ahead	32
Discount factor	0.9



Training Progress (12-Nov-2021 16:12:47)

Episode Information

Episode Number	40
Episode Reward	-81265277.9275
Episode Steps	12000
Episode Q0	-238771.1562
Total Number of Steps	480000

Average Results

Average Reward	-89905240
Average Steps	12000
Window Length for Averaging	10

Training Options

Hardware Resource	cpu
Learn Rate	0.01
Maximum Number of Episodes	40
Maximum Steps per Episode	12000

Final Results

Training Stopped by	Maximum Number of Episodes
Training Stopped at Value	40

Fig. 4.10: Rewards vs episodes of training of FlexE agents.

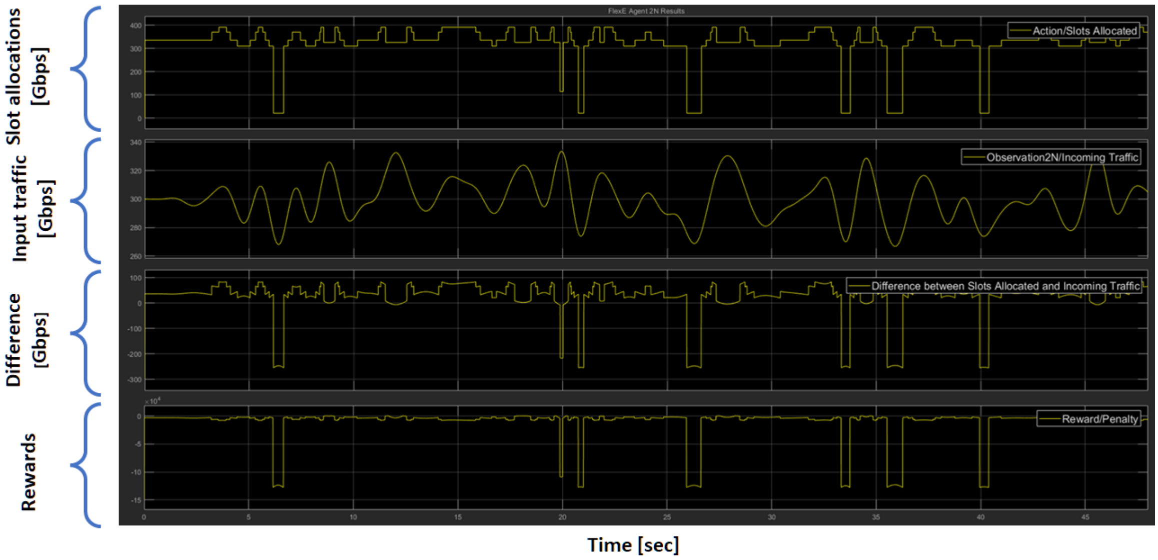


Fig. 4.11: Deployment results of FlexE agent.

of Fig. 4.10, we can conclude that as the number of episodes increases the average reward collected by the agent decreases, as shown in red, which is counterintuitive to the training process. The deployment results for the FlexE agent trained after 48 episodes are shown in Fig. 4.11. The first subplot shows the allocated slots to the input traffic pattern which is shown in the second subplot. The third subplot shows the difference between the input traffic and allocated slots. From the observation of the first subplot, we can conclude that the trained FlexE agent can allocate slots to the traffic pattern, yet there are under-allocations.

In the next step, the reward function was modified so that the over-allocation and under-allocation both have the same negative reward. Secondly, it was also ensured that the reward linearly decreases for the difference between the input traffic and allocates slots between 0 and 20 Gbps. The reward function is given in Equation 4.3 in Section 4.2.1 and plotted in Fig. 4.12

The agent was retrained again with deep-Q parameters given in Table 4.4. The rewards collected after training FlexE agents for 60 episodes are plotted in Fig. 4.13. From the observation of the results in Fig. 4.13, we can conclude that the reward starts exceeding the value of zero rewards after 20 training episodes and the average reward in the red plot saturates after 30 training episodes. This was not the case in the previous reward function. The results after the deployment of trained agents after training 44 episodes are given in Fig. 4.14. From the observation of Fig. 4.14, we can conclude that the allocated slots shown in the first subplot follow the input traffic pattern and there are no under-allocations. The difference between the input traffic and allocated slots remains between 0 and 20 Gbps. The

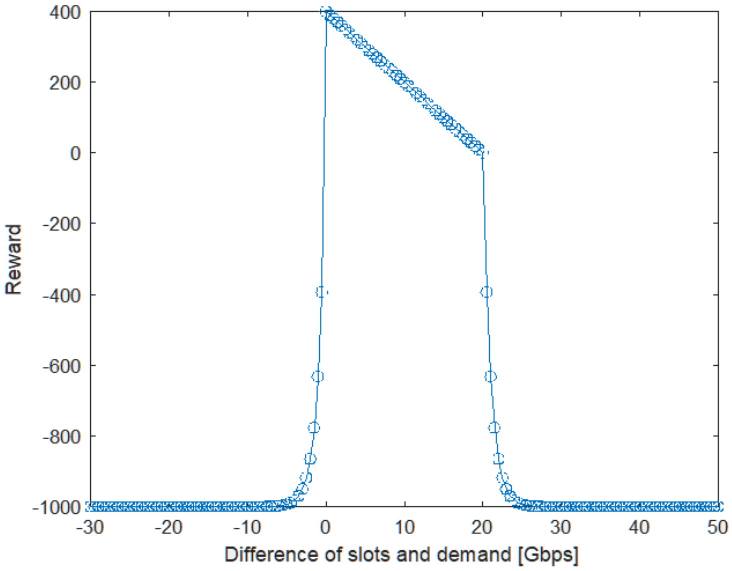
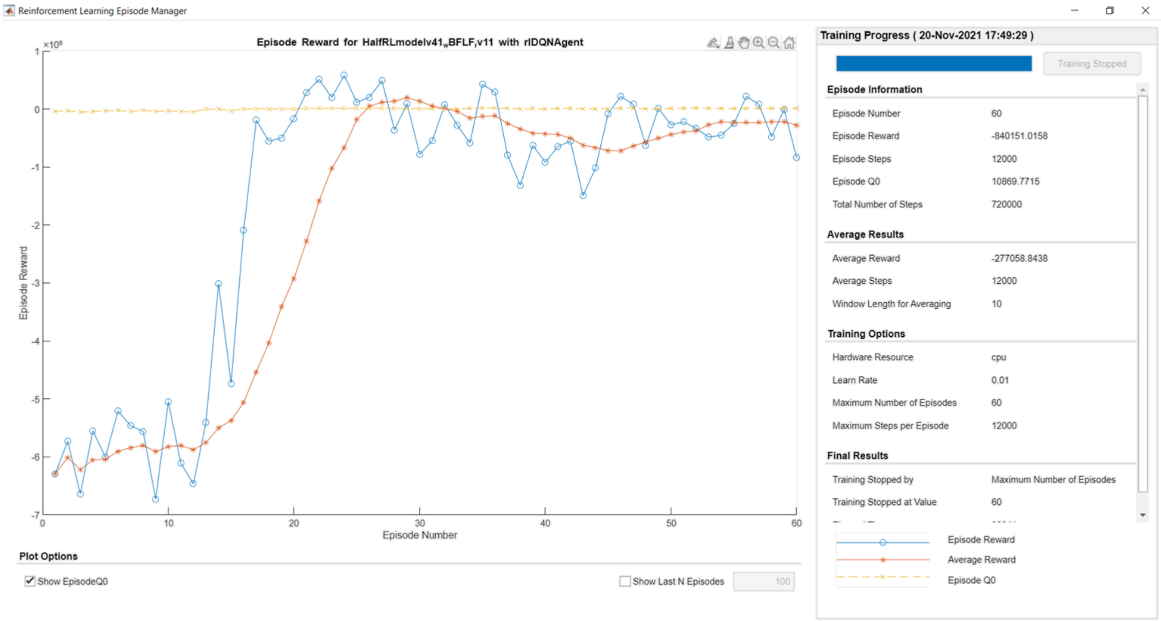


Fig. 4.12: Reward function for the FlexE agent.



Training Progress (20-Nov-2021 17:49:29)

Episode Information

Episode Number	60
Episode Reward	-840151.0158
Episode Steps	12000
Episode Q0	10869.7715
Total Number of Steps	720000

Average Results

Average Reward	-277058.8438
Average Steps	12000
Window Length for Averaging	10

Training Options

Hardware Resource	cpu
Learn Rate	0.01
Maximum Number of Episodes	60
Maximum Steps per Episode	12000

Final Results

Training Stopped by	Maximum Number of Episodes
Training Stopped at Value	60

Plot Options

☑ Show EpisodeQ0

□ Show Last N Episodes 100

Fig. 4.13: Rewards vs episodes during training of FlexE agent.

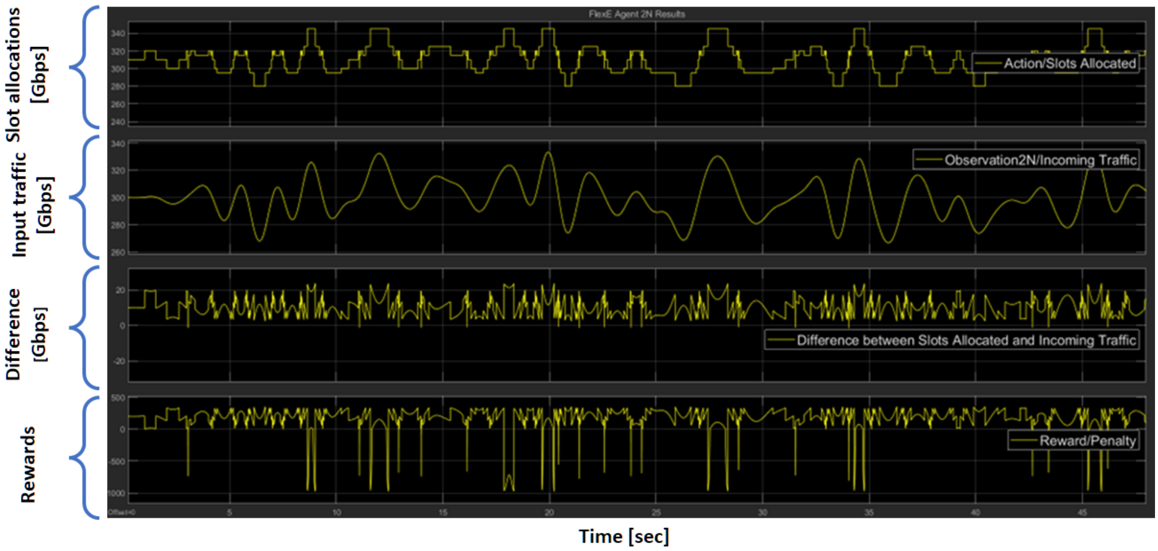


Fig. 4.14: Rewards vs episodes during training of FlexE agent.

only shortcoming of using the reward function in Equation 4.14 is that the average rewards collected by the agent saturate towards negative rewards after 30 episodes. However, the average reward is higher than in the previous case after 30 episodes of training.

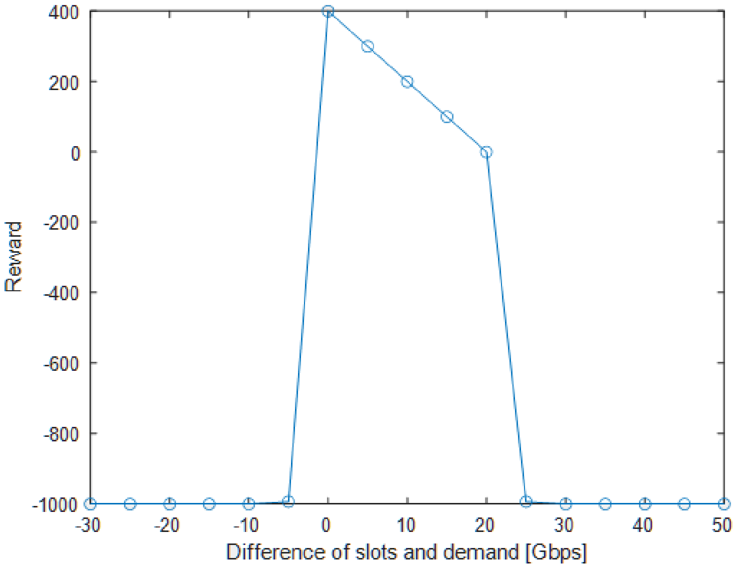
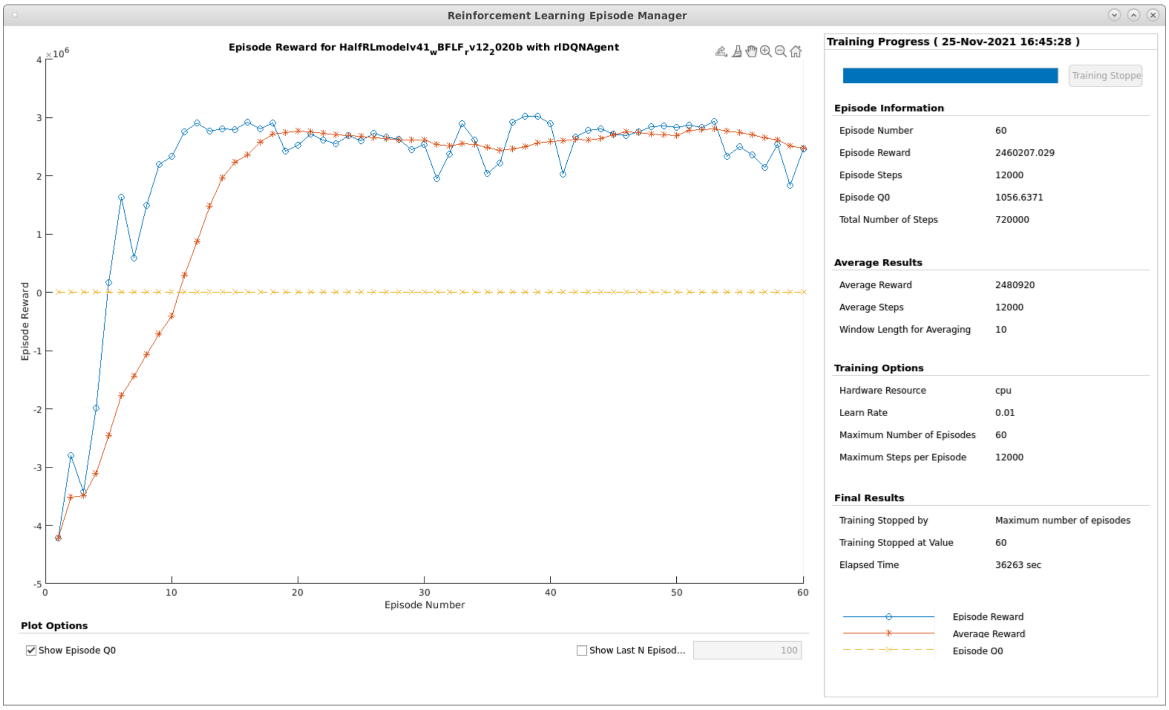


Fig. 4.15: Discretised reward function.

In many DRL applications, input data, such as traffic patterns, sensor readings, or financial data, often exist in continuous spaces. Dealing with continuous input spaces poses challenges for DRL algorithms, especially when it comes to learning meaningful representations and making accurate predictions. Discretisation involves dividing the continuous input space into a finite set of discrete states or bins. This process can facilitate faster learning of the DRL agent by reducing the complexity of the input space and making the computation more tractable [11]. Discretisation can help the agent to focus on important features or patterns in the data, leading to better decision-making and ultimately gathering increasing cumulative rewards with the progression of the training process.

Considering the benefits of discretization of input sample space and to enhance the rewards collected by the agent towards collecting average positive rewards, the input traffic was discretised before being given as an input observation to the agent. The input traffic was discretised with intervals of 5 Gbps and a ceil function was used. The reward function discussed previously in Equation 4.3 was used again. However, to increase the reward earning with every training episode and stabilise the earned rewards, the parameters of the number of steps to look ahead were reduced to 8 and the discount factor was also reduced to 0.7. The discretised reward function is plotted in Fig. 4.15. The rewards collected after 60 training episodes of the FlexE agent are given below in Fig. 4.16.

From the observation of Fig. 4.16, we can conclude that the average rewards earned by the FlexE agent increase with increasing episodes, as well as it earns positive rewards for episode number 5. This indicates that the discretisation of the input traffic stabilises the



Reinforcement Learning Episode Manager

Training Progress (25-Nov-2021 16:45:28)

Episode Information

Episode Number	60
Episode Reward	2460207.029
Episode Steps	12000
Episode Q0	1056.6371
Total Number of Steps	720000

Average Results

Average Reward	2480920
Average Steps	12000
Window Length for Averaging	10

Training Options

Hardware Resource	cpu
Learn Rate	0.01
Maximum Number of Episodes	60
Maximum Steps per Episode	12000

Final Results

Training Stopped by	Maximum number of episodes
Training Stopped at Value	60
Elapsed Time	36263 sec

Episode Reward
Average Reward
Episode Q0

Plot Options

☑ Show Episode Q0

□ Show Last N Episod... 100

Fig. 4.16: Rewards vs episodes during training of FlexE agent.

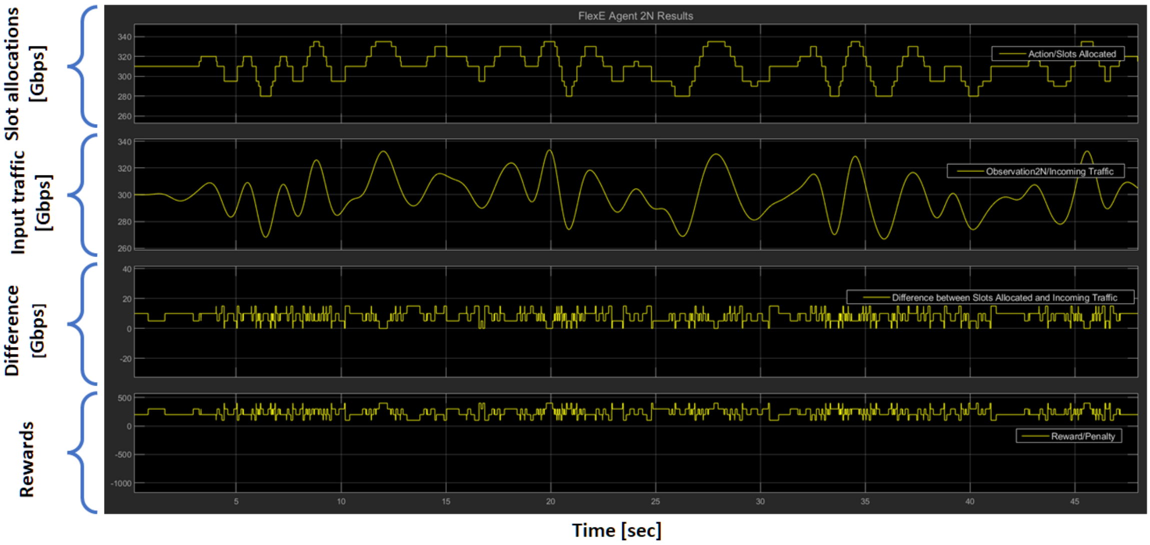


Fig. 4.17: Deployment results of FlexE agent trained after 38 episodes.

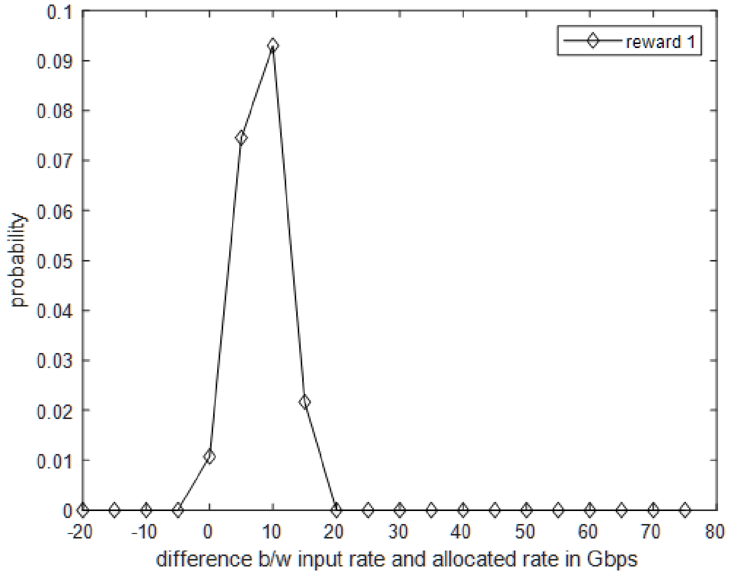


Fig. 4.18: Probability density of difference between input bit rate and allocated slots.

process of earning reward functions as the training process continues. It can also be observed that the earned rewards stabilise after 15 episodes till the end of the training process that is 60 episodes. The results after deployment of the agent trained for 38 episodes are given in Fig. 4.17.

From the observation of Fig. 4.17, specifically the third subplot, we can conclude that the difference between the input traffic and allocated slots does not exceed 15 Gbps. From the fourth subplot in Fig. 4.17, we can observe that the agent did not earn any negative reward, thus avoiding under-allocation, which can be catastrophic for delays in the fronthaul. The PDF of the difference between slot allocation and input traffic is given below in Fig. 4.18. From the observation of Fig. 4.18, we can conclude that the highest probability density is observed for the difference of 10 Gbps and the probability density for the negative difference is zero. This ensures that the agent learns not to under-allocate slots, thus avoiding catastrophic delays due to high loads in aggregation nodes in fronthaul.

The reward function discussed previously was tested for the traffic patterns extracted from the MAWI WIDE project [6]. However, the traffic traces were upscaled by the factor of (5∼11) to emulate high load on the FlexE aggregation nodes since the maximum input data rate of the traces was found to be 6 Gbps. The PDFs, the mean and the standard deviations of the input traffic patterns upscaled by factors of (5∼11) are shown in Fig. 4.19. From the observation of Fig. 4.19, we can conclude that as the upscaling factor increases the mean and standard deviation of the input traffic pattern also increases.

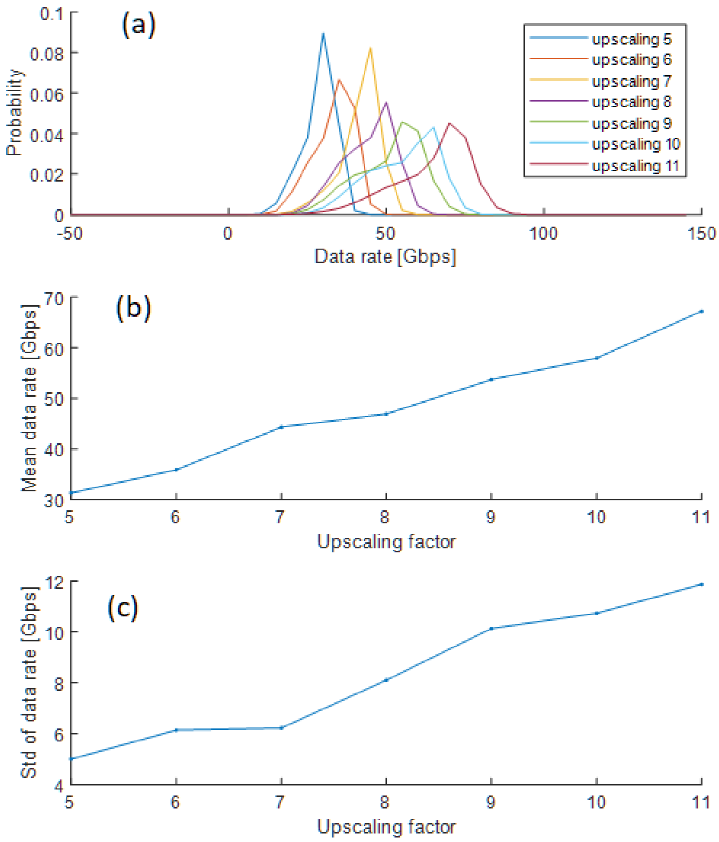
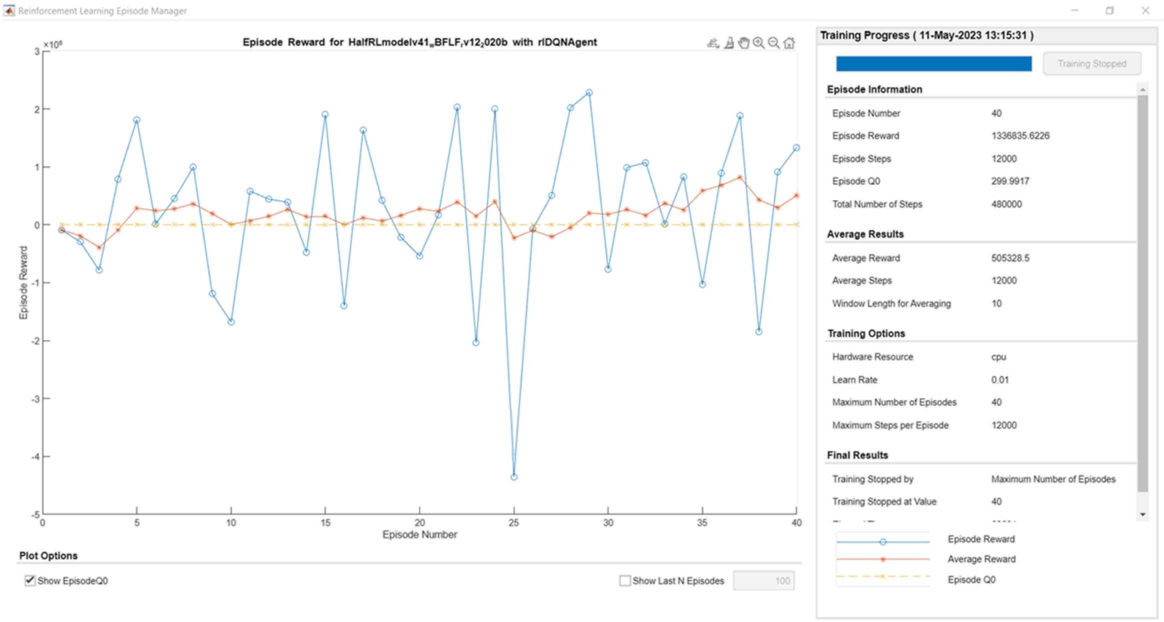


Fig. 4.19: PDFs (a), mean (b) and standard deviation (c) of input traffic patterns from WIDE MAWI projects with different upscaling factors.

During the training process, in every episode, the input traffic pattern was upscaled by a different factor. In this case, the experience buffer length was also increased to 40×12000 samples which corresponds to the total number of samples in 40 training episodes. The increase in the experience buffer length ensured that all of the samples of the input traffic states and action pairs with different properties of the mean and standard deviation were included in the experience buffer. The DRL agent could now sample from a larger experience buffer and train itself on different patterns, rather than a few available samples. The minibatch size was also increased to 2048 from 128, to improve the speed and accuracy of convergence of the training of deep-Q networks [12]. The rewards collected during the training process using the parameters given in Table 4.5 are given in Fig. 4.20.

Table 4.5: Deep-Q network parameters for FlexE agent training.

Learning Rate	0.01
Gradient Threshold	1
Number of Hidden Layers	6
Number of Neurons per Layer	98
Sample Time	4×10^{-3}
Experience Buffer length	40×6000
Mini batch size	2048
Epsilon	1
Epsilon Decay factor	0.005
Number of steps to look ahead	8
Discount factor	0.7



Plot Options

☑ Show EpisodeQ0 □ Show Last N Episodes 100

Training Progress (11-May-2023 13:15:31)

Episode Information

Episode Number	40
Episode Reward	1336835.6226
Episode Steps	12000
Episode Q0	299.9917
Total Number of Steps	480000

Average Results

Average Reward	505328.5
Average Steps	12000
Window Length for Averaging	10

Training Options

Hardware Resource	cpu
Learn Rate	0.01
Maximum Number of Episodes	40
Maximum Steps per Episode	12000

Final Results

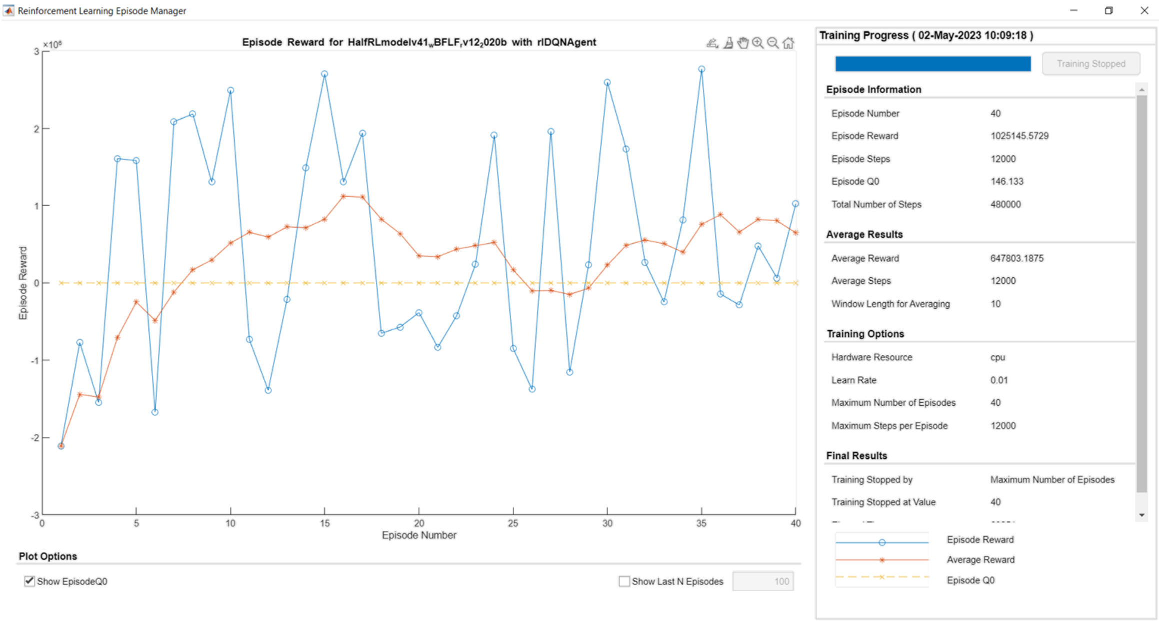
Training Stopped by	Maximum Number of Episodes
Training Stopped at Value	40

Fig. 4.20: Rewards vs episodes during training of FlexE agent.

Table 4.6: Deep-Q network parameters for FlexE agent training.

Learning Rate	0.01
Gradient Threshold	1
Number of Hidden Layers	6
Number of Neurons per Layer	98
Sample Time	$4x10^{-3}$
Experience Buffer length	$40x6000$
Mini batch size	2048
Epsilon	1
Epsilon Decay factor	0.005
Number of steps to look ahead	8
Discount factor	0.3

From the observation of Fig. 4.20, we can conclude that the cumulative rewards, plotted in red, do not stabilise and hardly exceed the zero-reward threshold, indicating that the agents might under or over-allocate the input traffic if deployed resulting in either high loads or bandwidth wastage. To improve the reward collections, the discount factor was reduced to 0.3 from 0.7 since the input observations are more decorrelated in every training episode. This reduction in discount factor resulted in more accurate predictions of input traffic patterns. The rewards collected by training the FlexE agent using the parameters in Table 4.6 are shown in Fig. 4.21.



Training Progress (02-May-2023 10:09:18)

Episode Information

Episode Number	40
Episode Reward	1025145.5729
Episode Steps	12000
Episode Q0	146.133
Total Number of Steps	480000

Average Results

Average Reward	647803.1875
Average Steps	12000
Window Length for Averaging	10

Training Options

Hardware Resource	cpu
Learn Rate	0.01
Maximum Number of Episodes	40
Maximum Steps per Episode	12000

Final Results

Training Stopped by	Maximum Number of Episodes
Training Stopped at Value	40

Plot Options

☑ Show EpisodeQ0

□ Show Last N Episodes 100

Fig. 4.21: Rewards for FlexE agent training.

By observing the training process in Fig. 4.21, we can conclude that the rewards collected by the FlexE agent vary as the training process proceeds towards the last 40 episodes, yet the FlexE agent could collect more positive cumulative rewards, than in the previous case in Fig. 4.20. The deployment results of the FlexE agent trained after 35 episodes on the traffic patterns for the upscaling factors of (5∼11) are shown in the following Fig. 4.22 till Fig. 4.28, which indicate that with the increasing upscaling factors the standard deviations of the input traffic also increases, as shown in the second subplots. It can also be observed in Fig. 4.22 till Fig. 4.28, that the traffic/slot allocations, shown in the first subplots, follow the input traffic shown in the second subplots, for all the upscaling factors. However, the difference between the input traffic and allocated traffic increases, shown in the third subplots of Fig. 4.22 till Fig. 4.28. This might be due to the statistically varying nature of the input traffic which increases with the upscaling factor. Further analysis of the results in Fig. 4.22 till Fig. 4.28 is carried out on page no. 138 using Fig. 4.29

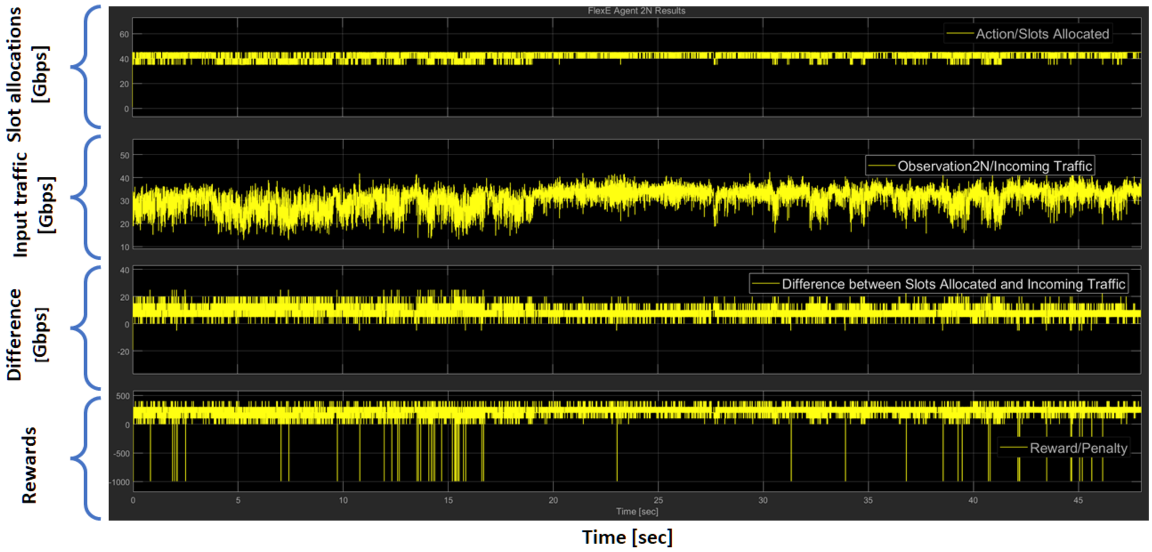


Fig. 4.22: Deployment results for the FlexE agent trained for 35 episodes for the traffic pattern upscaled by a factor of 5.

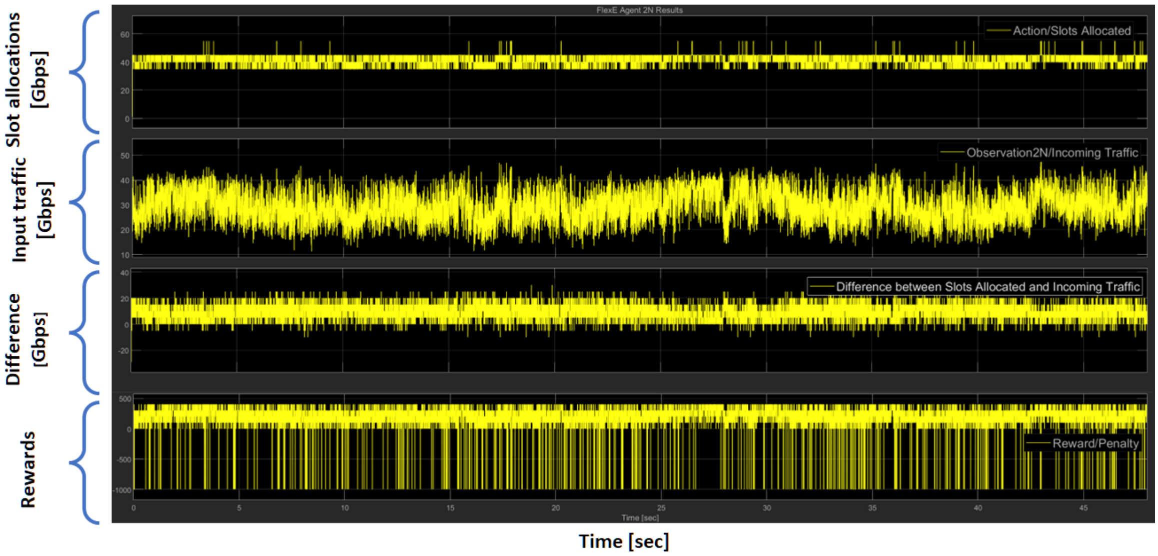


Fig. 4.23: Deployment results for the FlexE agent trained for 35 episodes for the traffic pattern upscaled by a factor of 6.

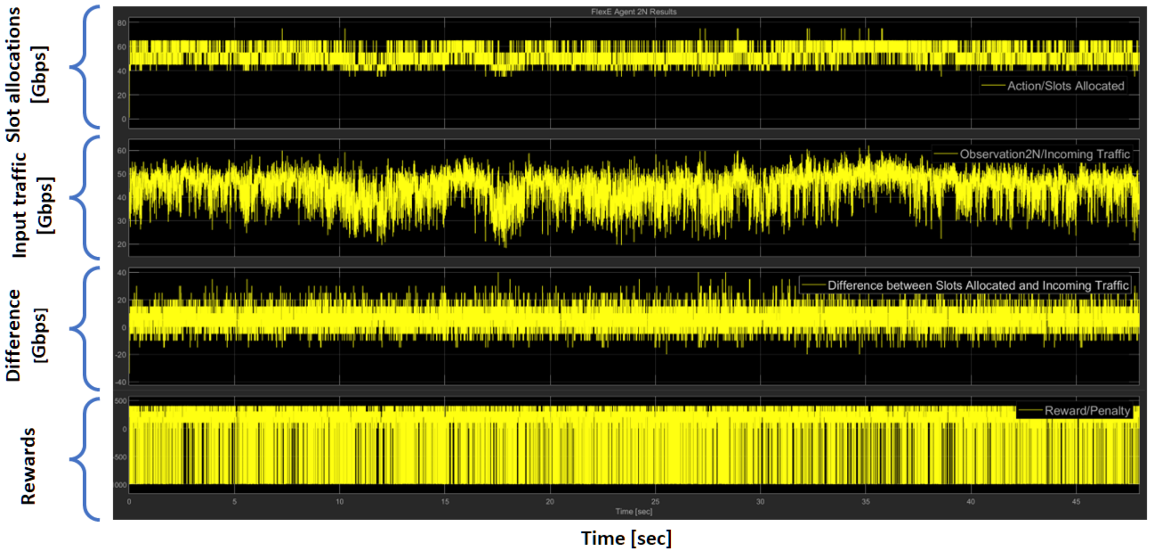


Fig. 4.24: Deployment results for the FlexE agent trained for 35 episodes for the traffic pattern upscaled by a factor of 7.

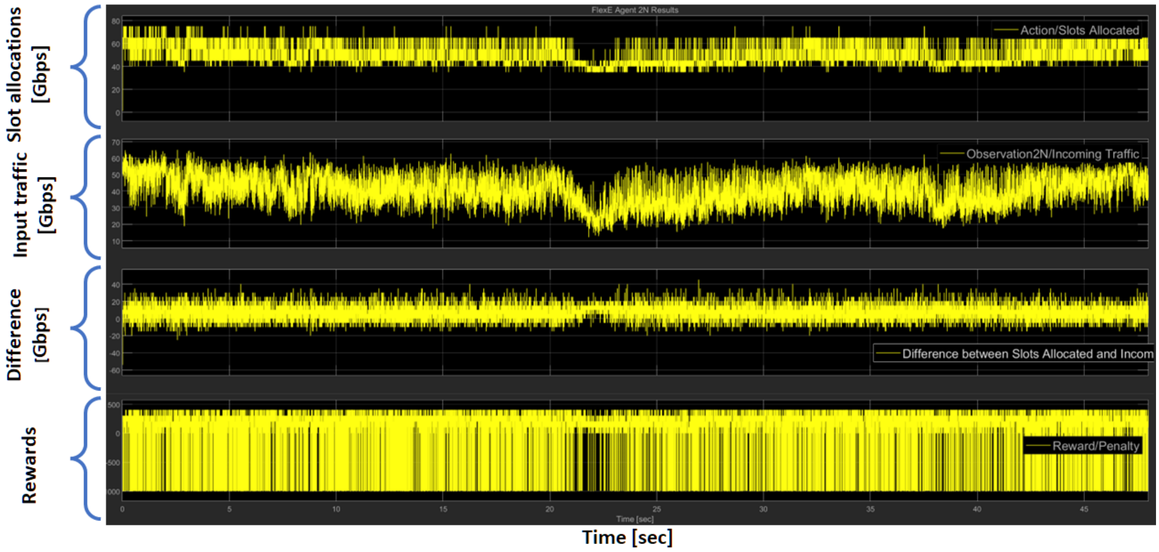


Fig. 4.25: Deployment results for the FlexE agent trained for 35 episodes for the traffic pattern upscaled by a factor of 8.

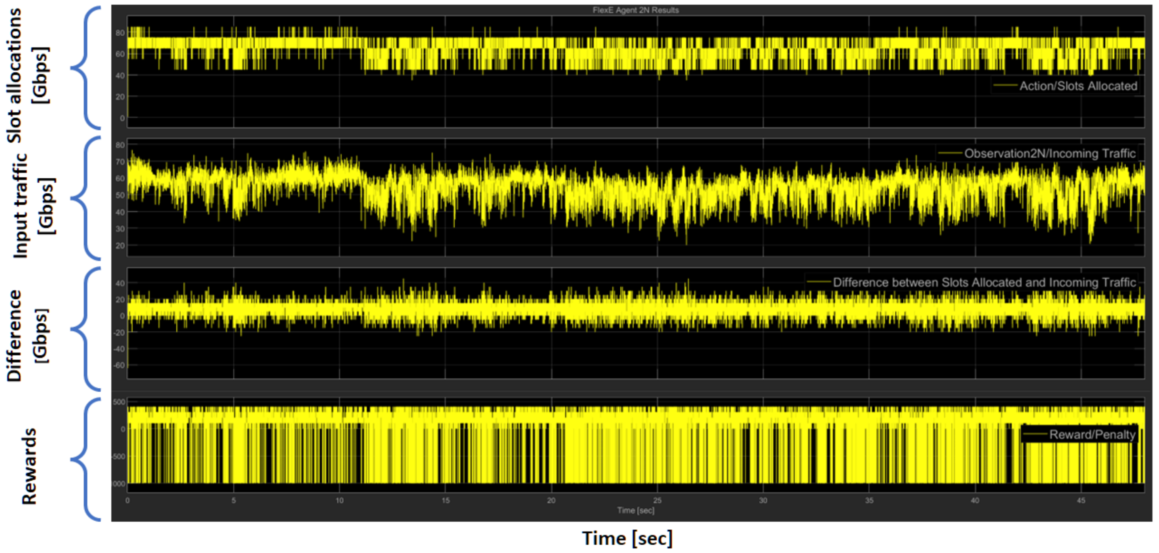


Fig. 4.26: Deployment results for the FlexE agent trained for 35 episodes for the traffic pattern upscaled by a factor of 9.

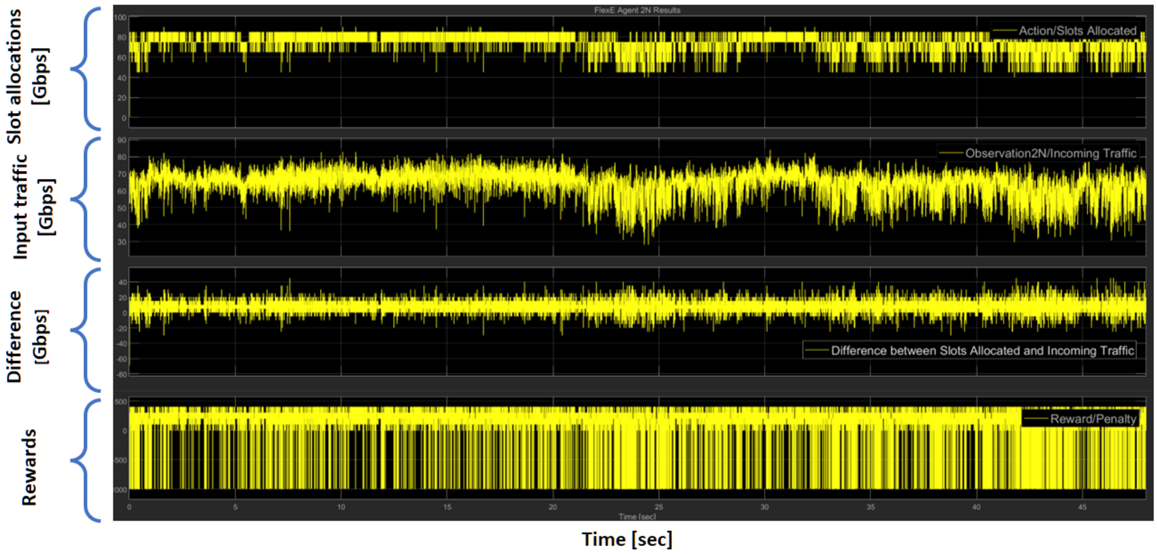


Fig. 4.27: Deployment results for the FlexE agent trained for 35 episodes for the traffic pattern upscaled by a factor of 10.

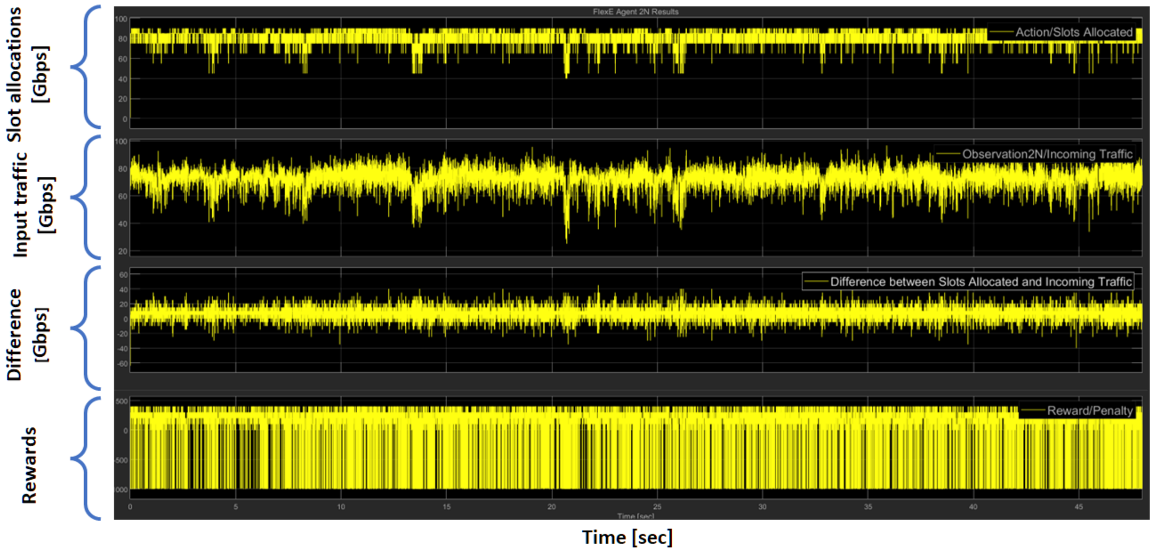


Fig. 4.28: Deployment results for the FlexE agent trained for 35 episodes for the traffic pattern upscaled by a factor of 11.

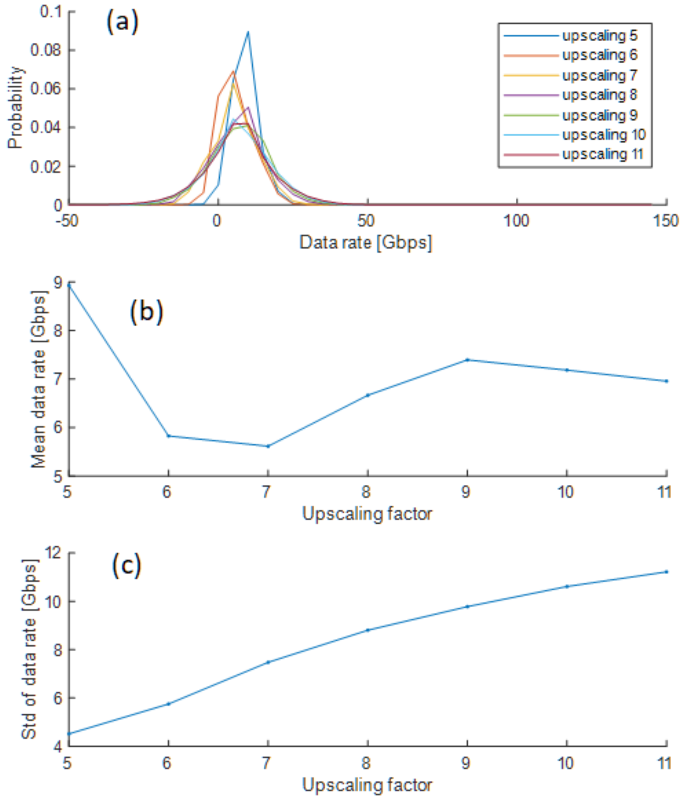


Fig. 4.29: PDFs (a), mean (b) and standard deviation (c) of the difference b/w allocated bit rates and input traffic.

To assess the performance of the FlexE agent for traffic prediction and slot allocation to input traffic in Fig. 4.22 till Fig. 4.28, the PDFs, the means and the standard deviations of the difference between the input traffic and slot allocations (in third subplots in Fig. 4.22 till Fig. 4.28) is given in Fig. 4.29. From the observation of Fig. 4.29, we can conclude that as the upscaling factor increases the standard deviation of the difference also increases thus resulting in more under-allocations in the system. Again, this might be due to the higher standard deviation of the input traffic with increasing upscaling factors. This also indicates that the FlexE agent might also require more episodes of training than in the analysis in the thesis.

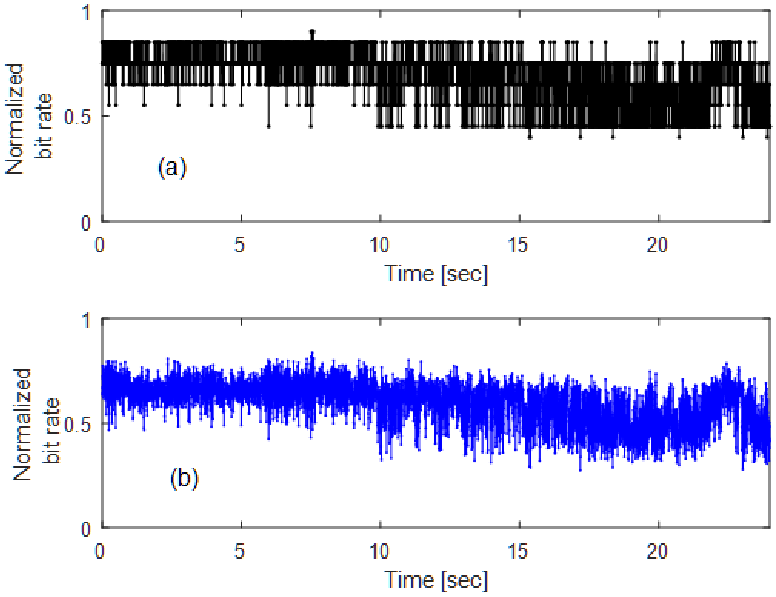


Fig. 4.30: FlexE Agent slot allocation (a), input traffic (b).

After rigorous testing of reward functions and training of the FlexE agent discussed above the FlexE agent was finally trained again with the input traffic both from the CoMP model discussed in Section 4.3.1 and traffic patterns from the WIDE MAWI project. The deployment results of the FlexE agent are given in Fig. 4.30. The input traffic pattern is upscaled by a factor of 10 and normalised to the output rate of the FlexE aggregation node, where Fig. 4.30 (b) shows the input traffic and Fig. 4.30 (a) shows the number of slots allocated to the input traffic. From the analysis of Fig. 4.30 (a), it can be observed that the agent attempts to track the input traffic and allocated slots.

4.4.1 Cumulative Under Allocation and Mean-Over Allocation

One of the primary research objectives of this thesis is to enhance bandwidth utilisation in fronthaul links. By equipping these links with FlexE aggregation nodes, the slots in the FlexE calendars can be optimally allocated to input traffic patterns, thereby improving bandwidth efficiency at the physical layer. This enhancement is measured using the mean value of over-allocation (MOA) of bit rates assigned by the DRL agent to the input traffic. In this research, MOA is defined as the average positive difference between the bit rates allocated by the DRL based on traffic predictions and the input traffic bit rates.

The mean over-allocation provides a central measure of the overall bandwidth over-allocation. By comparing the mean over-allocation to a predetermined threshold or industry standard, we can evaluate the extent of resource wastage. This information is valuable for identifying potential areas for optimisation and improving the efficiency of resource utilisation. Specifically, the MOA is defined as the mean or average value of the difference of allocated bit rates and input traffic in the FlexE calendar being greater than zero and is defined as follows,

$$MOA = \mathbb{E}[(S - R_D) \geqslant 0], \tag{4.15}$$

where, S is the bit rate allocated in the FlexE calendar by the FlexE calendar (first subplot in Fig. 4.22 till Fig. 4.28), R_D is the input traffic patterns or demand extracted from the MAWI wide project (second subplot in Fig. 4.22 till Fig. 4.28), and the difference $(S - R_D)$ is given in the third subplots in the Fig. 4.22 till Fig. 4.28.

The under-allocation of bit rates by the DRL agent in the FlexE aggregation node in the fronthaul in Fig. 4.1 can result in loads exceeding the value of 1, resulting in catastrophic infinite delays and packet drops in the network. To predict the likelihood of such events more accurately, the cumulative probability of under-allocation (CUA) was used to benchmark under-allocations.

The MOAs of the DRL-based bit rate allocations in FlexE were also compared with other traffic prediction techniques such as auto-regressive moving average (ARIMA) [13] and fixed slot allocation. To make the comparison fair, the MOAs were compared such that the difference in bit rate allocations was at the same levels as CUAs. The details of the fair analysis are given in Section 4.4.3. The CUAs are the total probability for the difference of allocated bit rates and input traffic in the FlexE calendar being less than zero and are defined as follows,

$$CUP = Pr\left[(S - R_D) < 0\right], \tag{4.16}$$

where, S is the bit rate allocated in the FlexE calendar by the FlexE calendar (first subplot in Fig. 4.22 till Fig. 4.28), R_D is the input traffic patterns or demand extracted from the MAWI wide project (second subplot in Fig. 4.22 till Fig. 4.28), and the difference, $(S - R_D)$ is given in the third subplots in the Fig. 4.22 till Fig. 4.28.

4.4.2 ARIMA-Based Traffic Prediction

The performance of slot allocations from the FlexE agents in terms of CUA and MOA is also compared with the slot allocations from the auto-regressive moving average (ARIMA) approach [13]. ARIMA is a filtering-based prediction technique where input traffic predictions are dependent on the past sample values. A prediction from the ARIMA approach is given as follows [14],

$$ARMA\,(p,q) : Y_t = c + \sum_{i=1}^{p} \Phi_i Y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t, \tag{4.17}$$

where (p,q) are the filter coefficient orders, Y_t is the prediction based on past and future samples, c is a constant, ε_t is the error between the current prediction and the observed sample, Φ_i are the coefficients of the filter that samples past inputs and θ_i are the coefficients of the filter that sample the past errors. In ARIMA the coefficients of the prediction filters are optimised using the Yule-Walker equations [13]. The coefficients for the (ARIMA (8,1)), were calculated using the same training patterns for which the FlexE agent was trained. In the thesis research and the results given in [13], the order of the filter coefficients (p,q) was (8,1) and the slot allocation was the maximum value of 10 future predictions of the input traffic from ARIMA. The sampling interval between the two predictions in the analysis in the thesis and the analysis in [13] was 4 ms corresponding to the reconfiguration time of the FlexE calendar.

Since ARIMA estimates the average values based on past observations, this will result in under-allocation in the FlexE calendar. To overcome this shortcoming, the predictions from ARIMA $(\hat{R}_{ARMA})$ were normalised by a multiplication factor $(\alpha\hat{R}_{ARMA} : \alpha > 1)$, using a similar approach in [13]. This normalisation also ensured that the predictions from ARIMA resulted in the same CUA as that from the FlexE agent's predictions, for the input traffic which was upscaled by a factor of 10. The process of allocation of slots by the ARIMA (8,1) is given in Fig. 4.31(a) in black plots whereas the input traffic patterns from the WIDE MAWI project are shown in blue plots, with the magnified view for the initial 2 seconds is given in Fig. 4.31(b). It can be observed in Fig. 4.31(a) and Fig. 4.31(b) that ARIMA attempts to allocate fixed slots for 10 samples of input traffic based on the maximum traffic in the 10 samples. This static over-allocation results in over-allocations and bandwidth wastage in fronthaul.

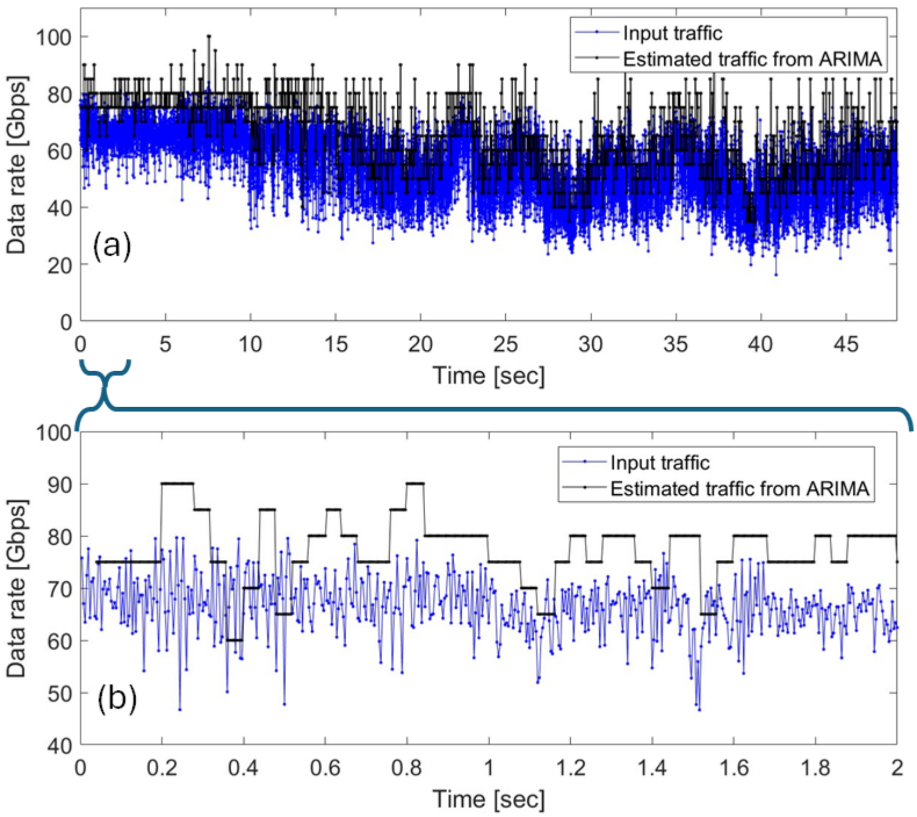


Fig. 4.31: ARIMA-based slot allocations (a), magnified view of ARIMA-based slot allocations (b).

4.4.3 Comparison of DRL-Based with ARIMA-Based and Fixed Traffic Allocation

The DRL-based predictions from the FlexE agent were compared with ARIMA-based predictions as well as fixed with the fixed traffic allocation-based approach where the number of slots allocated was equal to the maximum input traffic within 24 seconds, in Fig. 4.32 and Fig. 4.33. The results for the three approaches were compared in terms of MOAs of traffic as a measure of bandwidth over-consumption in the fronthaul for 24 seconds. The MOAs for upscaling factors of 5 to 11 of the input traffic are shown in Fig. 4.32. It can be observed in Fig. 4.32 that at an upscaling factor of 10, the MOAs from DRL-based predictions are 11.6 % less than the mean over-allocation from ARIMA-based predictions and 58% less than from fixed allocations. For the other upscaling factors of the input traffic patterns, the MOAs using DRL are also generally less than the mean over-allocations from ARIMA and fixed allocations (except for the case of an upscaling factor of 5, where ARIMA gave better performance).

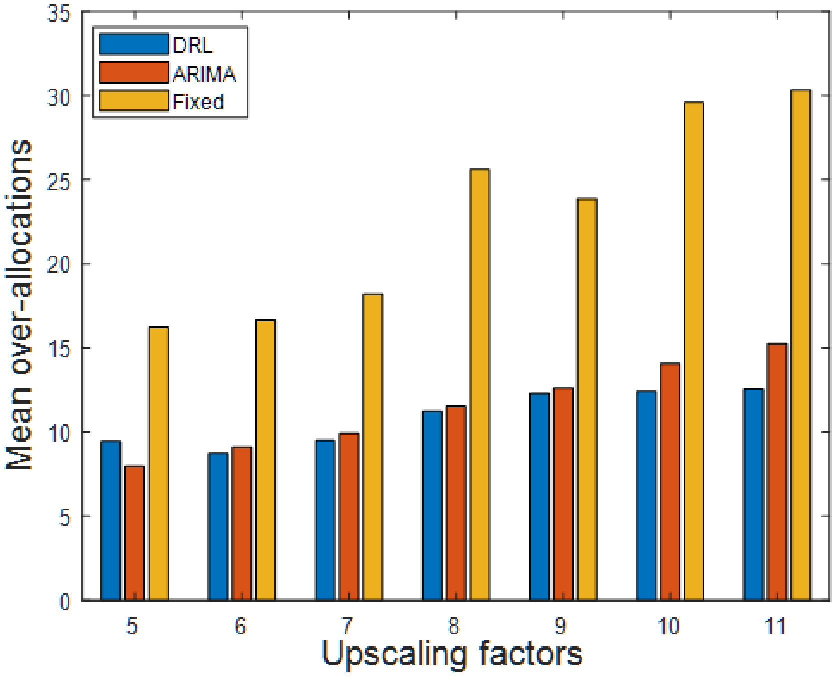


Fig. 4.32: MOAs for different upscaling factors and allocation schemes (for the same CUA at the upscaling factor of 10).

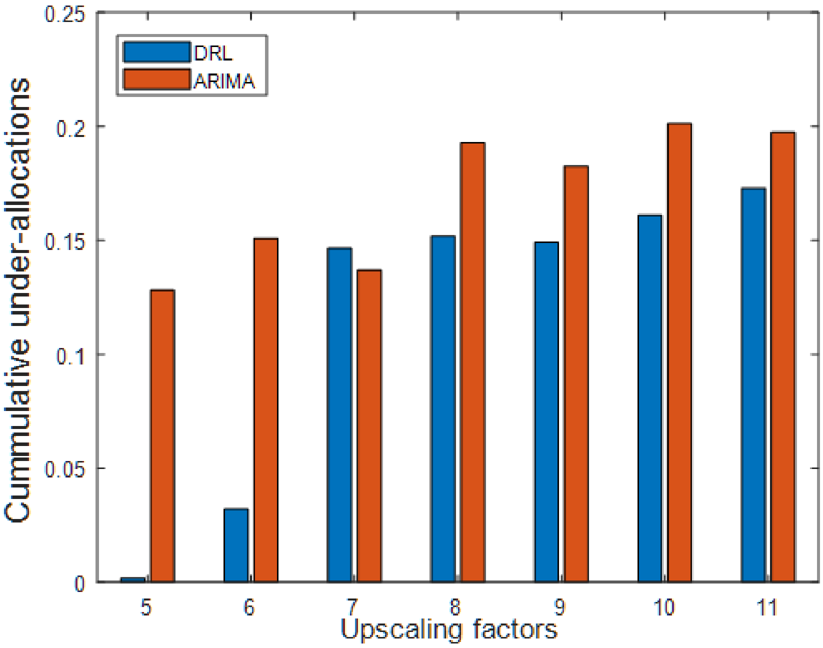


Fig. 4.33: CUAs for different upscaling factors and allocation schemes (for the same mean over-allocation at the upscaling factor of 10).

To compare the performance of the three approaches in terms of CUA, the predictions from ARIMA ($\hat{R}_{ARMA}$) were normalised by a positive multiplication factor ($\alpha\hat{R}_{ARMA} : \alpha > 1$) so that they result in the same MOA as that from the FlexE agent's predictions. The resulting CUAs for the upscaling factors of 5 to 11 of the input traffic are shown in Fig. 4.33. The CUAs for fixed allocation are not shown in Fig. 4.33 since they are very low in comparison to CUAs from DRL and ARIMA. It can also be observed in Fig. 4.33 that at an upscaling factor of 10, the CUAs from DRL are 25% less than the CUAs from ARIMA-based predictions. Therefore, DRL can outperform ARIMA-based predictions in terms of efficient bandwidth consumption.

4.5 Controller Agent Training Parameters and Results

The 'controller agent' was trained simultaneously with the FlexE agents, also using deep-Q learning in the reinforcement learning toolbox in Simulink MATLAB®. The controller agent aims to maximize the total achievable bit rate of the two RU groups by offloading users from a path with higher latency to a path with lower latency in the fronthaul network. The actions of the controller agent are the offloading commands for each RU group, of integer values ranging from 0 to 5. In any training step, the number of users offloaded was limited to 5 users. This resulted in a monotonically increasing average reward earned by the controller agent during the training. The reward function for the controller agent resulting in a monotonically increasing reward is given in Equation 4.6 and Equation 4.7 in Section 4.2.2.

Similar to the FlexE agent the controller agent was trained for 40 episodes, with each training episode spanning over 24 sec. The parameters for the deep-Q network for the training of the controller agent are given in Table 4.7. The major difference from the parameters of the training of the FlexE agent is that the number of steps to look ahead is 32, whereas, for the FlexE agent, it was 8. The discount factor for the controller agent is 0.99, whereas, for the FlexE agent, it was 0.7. The reason the significantly larger number of steps to look ahead and discount factor were chosen is so that the controller agent was able to earn monotonically increasing rewards during the progression of the training process.

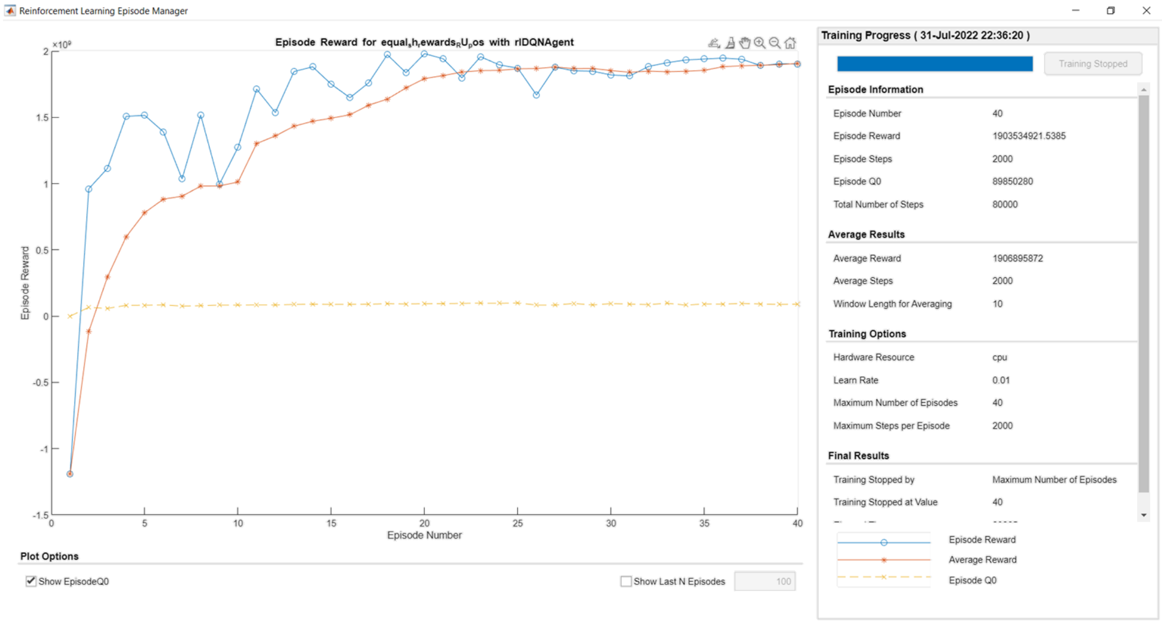
For the training of the controller agent, the locations of UEs were uniformly distributed over the 20×20 m grid to maintain a balance of effective bit rates of the RU groups. Also, the locations of RUs were in a hexagonal formation with a radius of 50m so that the RUs are equidistant from each other and on average equidistant from UE locations as shown in Fig. 4.5. Equal average background traffic of 110 Gbps was added into the paths of the RU groups to emulate equal loads resulting in a balanced impact of latency in the two paths. The total bit rate of the two RU groups added to the background traffic was around 234.089 Gbps after

Table 4.7: Deep-Q network parameters for controller agent training.

Learning Rate	0.01
Gradient Threshold	1
Number of Hidden Layers	6
Number of Neurons per Layer	98
Sample Time	4×10^{-3}
Experience Buffer length	40×6000
Mini batch size	2048
Epsilon	1
Epsilon Decay factor	0.005
Number of steps to look ahead	32
Discount factor	0.99

applying the functional split factor of 10.5. The rewards earned during the training process of the controller agent are given in Fig. 4.34, which indicates that the controller agent was able to earn monotonically increasing rewards with the progression of the training process, as expected from a DRL agent training.

A snapshot of the training process of the controller agent is shown in Fig. 4.35. The total bit rate of the UEs in the two RU groups is shown in the first subplot in Fig. 4.35(a) and the number of UEs in the two RU groups is shown in Fig. 4.35(b). From the analysis of Fig. 4.35(b), it can be observed that the controller agent offloads the UEs from one RU group to another when the total bit rate of the RU groups falls below the average value of 234 Gbps due to the impact of varying latency in fronthaul. It does so to learn to maximize the total bit rate of the two RU groups as well as maintain the total number of UEs constant at 30 and a total bit of 234 Gbps.


Reinforcement Learning Episode Manager
Episode Reward for equal$_s$h$_r$ewards$_R$U$_p$os with rlDQNAgent
×10⁹
2
1.5
1
0.5
0
-0.5
-1
-1.5
Episode Reward
0
5
10
15
20
25
30
35
40
Episode Number
Training Stopped

Training Progress (31-Jul-2022 22:36:20)

Episode Information

Episode Number	40
Episode Reward	1903534921.5385
Episode Steps	2000
Episode Q0	89850280
Total Number of Steps	80000

Average Results

Average Reward	1906895872
Average Steps	2000
Window Length for Averaging	10

Training Options

Hardware Resource	cpu
Learn Rate	0.01
Maximum Number of Episodes	40
Maximum Steps per Episode	2000

Final Results

Training Stopped by	Maximum Number of Episodes
Training Stopped at Value	40

Episode Reward
Average Reward
Episode Q0

Plot Options

☑ Show EpisodeQ0 □ Show Last N Episodes 100

Fig. 4.34: Rewards for controller agent training.

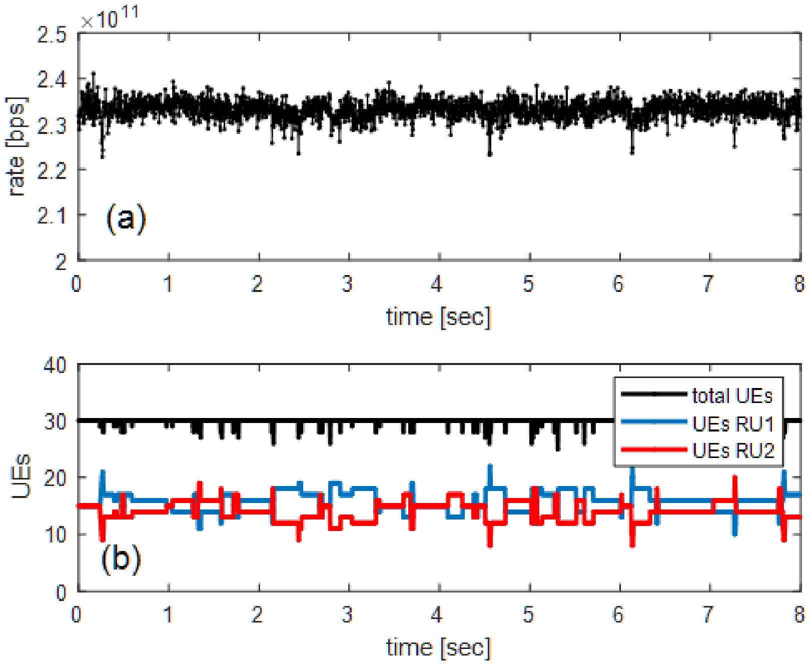


Fig. 4.35: Training results of controller agent, total user bit rate (a), number of UEs in the RU groups (b).

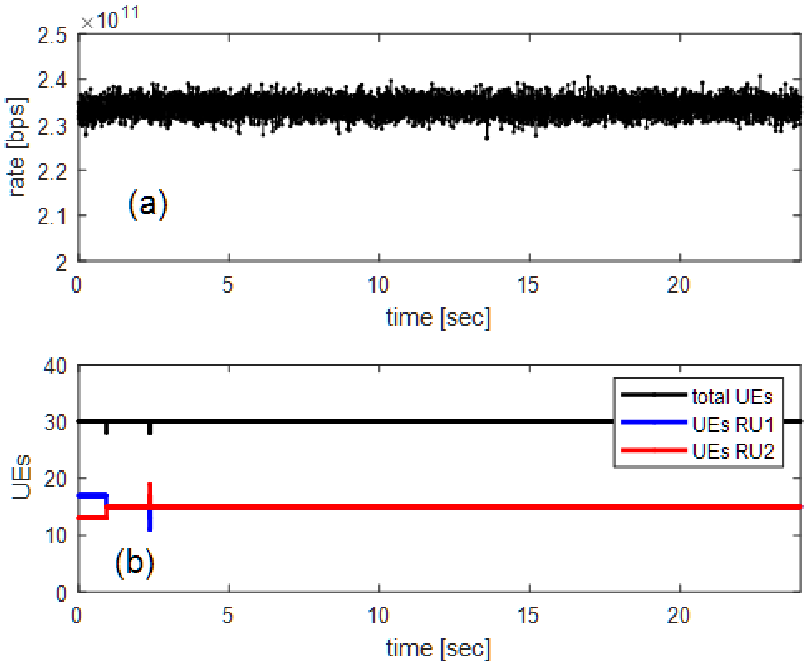


Fig. 4.36: Controller agent deployment results in Case-1(a) in Table 4.8, total user bit rate (a), number of UEs in RU groups (b).

4.5.1 Controller Agent Deployment Results

The controller agent trained for 36 episodes with parameters given in Section 4.5, rewards shown in Fig. 4.34 and the RU positions shown in Fig. 4.5 was deployed. Note that the RUs positions in Fig. 4.5 are in a hexagonal formation with a radius of 50m, and the simulation case for the RU positions in Fig. 4.5, is termed as 'case-1'. The deployment results for the controller agent are shown in Fig. 4.36. The first subplot in Fig. 4.36(a) shows the total bit rate of the two RU groups which is around 234.089 Gbps, the second subplot in Fig. 4.36(b) shows the total number of UEs in the two RU groups, along with UEs in the individual RU groups. It is apparent from Fig. 4.36(b) that the initial number of UEs in the two RU groups was 13 and 17 as a non-ideal starting distribution of UEs. The agent then offloads users so that both RU groups have the same number of users. This is so because the total achievable bit rate is maximum when the two RU groups have the same number of users.

The results in Figure 4.36 are also summarized in case-1(a) of Table 4.8, which lists the values of the background traffic in RU group paths, the number of users in the two RU groups after the deployment of the trained controller agent for 'case-1', the average user bit rates in the two RU groups and the total user bit rates in the two RU groups.

Table 4.8: Total user bit rates for different background traffics and RU position case-1.

RU position case-1	Back ground traffic (Gbps)		Number of users		Bitrate (Gbps)		Total bitrate (Gbps)
	RU-group		RU-group		RU-group		
	1	2	1	2	1	2	
a	110	110	15	15	117.04	117.04	234.1
b	285	110	15	15	86.16	115.8	201.8

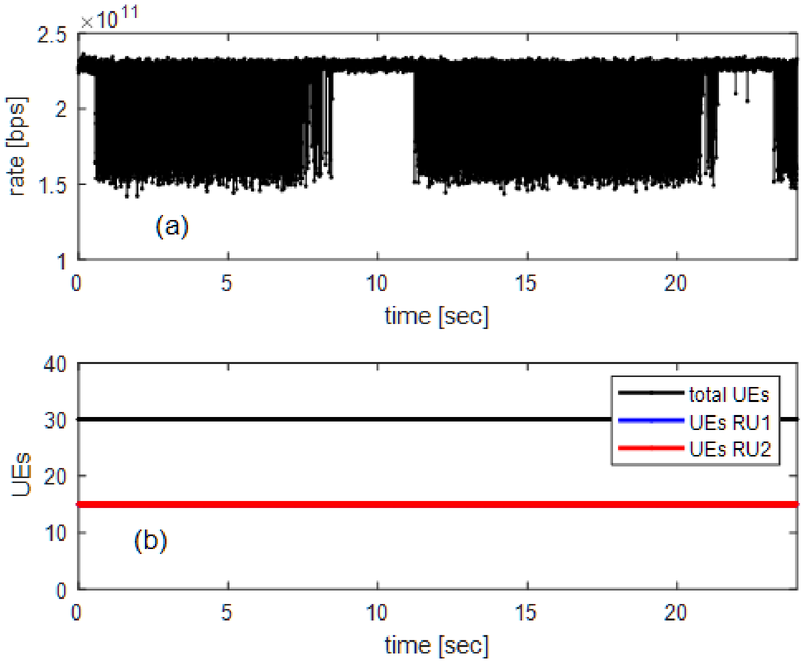


Fig. 4.37: Controller agent deployment results in Case-1(b) in Table 4.8, total user bit rate (a), number of UEs in RU groups (b).

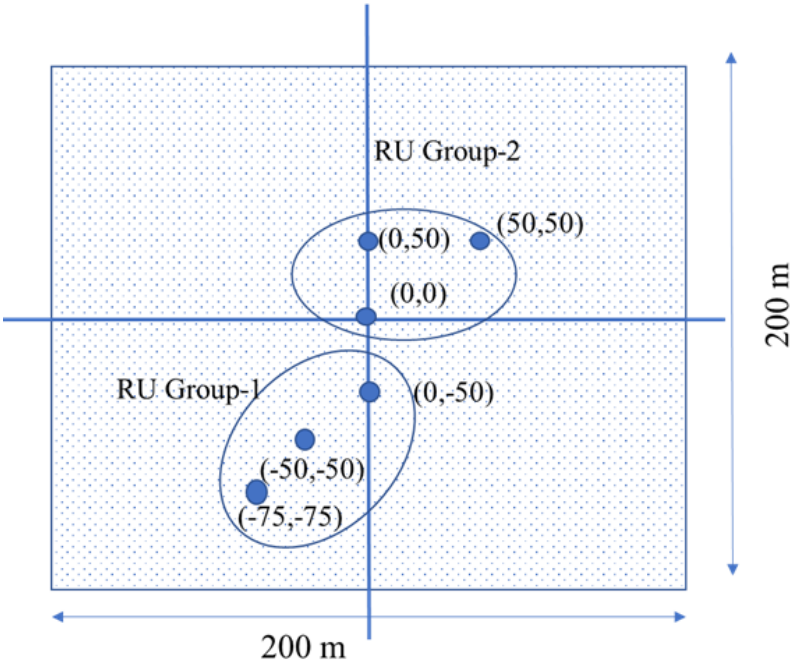


Fig. 4.38: RU coordinates in x,y plane (Case-2).

In the next step, higher background traffic of 285 Gbps was added in the path of RU group-1. Then the trained controller agent for case-1(a) only, in Table 4.8 and Fig. 4.36 was deployed. The controller agent did not offload users among RU groups since it was trained for case-1(a) with equal background traffic in both the RU groups. The resulting total bit rate of the two RU groups and the number of users in the two RU groups are shown in Fig. 4.37. It can be observed in Fig. 4.37(a) (first subplot), that the total bit rate of the RU groups reduces from 234.1 Gbps to a minimum of 115 Gbps and on average reduces to 201.8 Gbps. This shows that the addition of higher background traffic results in higher latency which then causes the reduction of the total bit rate. The results of the addition of higher background traffic in Fig. 4.37 are also listed in Table 4.8 as case-1(b).

In the next step, the positions of RUs were changed such that they have non-ideal positions and more non-uniform distances to the users distributed over the grid of 200×200m. The new coordinates of the RUs are shown in Fig. 4.38. Note that this simulation case for the RU positions in Fig. 4.38 is termed as 'case-2'. This configuration resulted in a higher average distance to users from RU-group-1 than from RU group-2. The controller agent was retrained for the new configuration of RU positions with different background traffic added to the path of RU group-1. The values of the background traffic in RU groups, the number of users in the two RU groups after the deployment of the trained controller agent in 'case-2', the average

Table 4.9: Total bit rates for different background traffics and RU position case-2.

RU positions case-2	Background traffic (Gbps)		Numberof users		Bitrate (Gbps)		Total bitrate (Gbps)
	RU-group		RU-group		RU-group		
	1	2	1	2	1	2	
a	110	110	15	15	113.3	115.8	229.1
b	114	111	14	16	107.8	121.2	229
c	250	110.7	9	21	77.25	145	222.3
d	285.5	105.67	7	23	62.9	153.5	216.5
e	331.5	111.5	3	27	31.38	166.67	198.5
f	113	265.26	19	11	125	100.2	225.2

user bit rates in the two RU groups and the total user bit rates in the two RU groups are given in Table 4.9 as case-2(a∼e).

In the simulation case-2(a∼e) in Table 4.9, the background traffic added to the paths of the RU groups was varied, such that the background traffic added to RU group-1's path was incremented and higher than the background traffic in RU group-2's path. The values of the background traffic in case-2(a∼e) are in Table 4.9, representing the average background traffic over the deployment period of 24 seconds. The background traffic in the RU group-2 path varied around 110 Gbps. It can be observed that for case-2(a) with the same background traffic of 110 Gbps in the two paths, the users in RU group-2 had a higher bit rate than RU group-1 since the users on average are closer to RU group-2. Despite this, the agent did not offload users from RU group-2 to RU group-1. It only carried the offloading from RU group-1 to RU group-2 when more background traffic was added in the path of RU group-1, and increasingly so as shown from case-2(b) to case-2(e) in Table 4.9. It can also be observed that as the background traffic is increased, due to the increase in latency, the bit rate of RU group-1 as well as the total bit rate of the two RU-groups also decreases. This indicates that the impact of latency in terms of reduction of the total bit rate of RU groups is more pronounced than the impact of unequal distance of users from the RU groups.

The deployment results for a trained controller agent for case-2(d) in Table 4.9 are also shown in Fig. 4.39 which had the same background traffic in both the paths of the two RU groups as that of case-1(b). In Fig. 4.39, it can be observed that the agent was able to learn to offload more UEs from the path having higher background traffic of 285 Gbps to a path of lower background traffic of 110 Gbps, to offset the effect of the reduction of the total bit rate of the RU groups. Due to the offloading by the controller agent, the average total bit rate in case-2(d) in Table 4.9 was 216.5 Gbps which was higher than in case-1(b) at 201.8 Gbps in

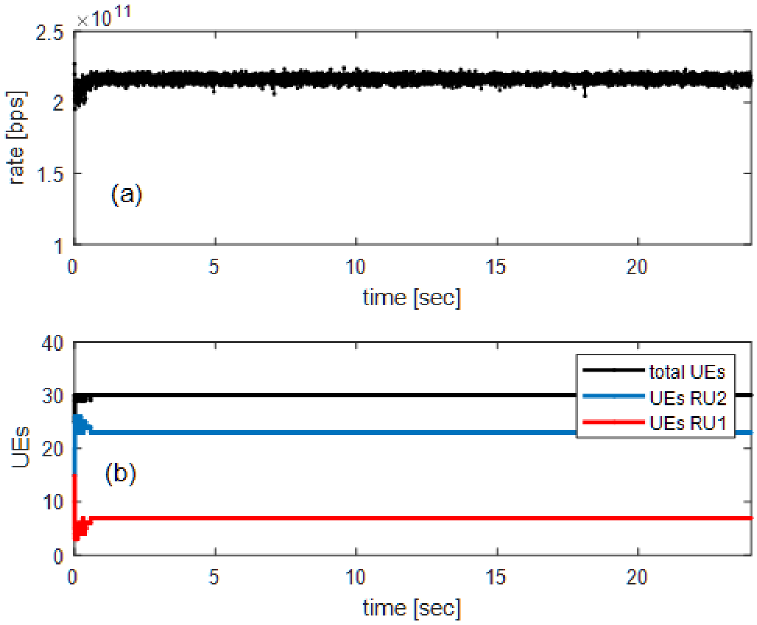


Fig. 4.39: Total bit rate with unequal background traffic (a) and UEs in RU groups (b) for case-2(d).

Table 4.8 where there was no offloading of UEs among RU groups for the same background traffic in RU group-1, resulting in a gain of 7.3% in throughput. This shows that the trained controller through offloading users among RU groups can improve the achievable bit rate of the system. In case-2(f) in Table 4.9, more background traffic was added to RU-group-2's path. This resulted in the offloading of users from RU group-2 to RU group-1 by the trained controller agent. This proves that the system has symmetry, and the trained controller agent can also learn to offload users in both directions.

4.6 Conclusions

In this Chapter, a combined mechanism for user and resource allocation based on DRL has been demonstrated as a promising way of achieving user bit rate maximisation in beyond-5G mobile networks. For this, a RAN using CoMP was implemented. The transport segment was assumed to comprise Ethernet and FlexE nodes. The achievable user bit rate was degraded by the latency in the fronthaul, as it added delay in the CSI sent to the cU/DU. The end-to-end latency in the fronthaul was modelled by 99.9 percentile delay instead of mean delay, to address the effect of buffering in the destination node in fronthaul, which in this case was CU/DU.

DRL was used for the prediction of traffic and slot allocation in FlexE calendars in the fronthaul. From the results it can be concluded that DRL can more efficiently allocate resources in FlexE-based fronthaul, outperforming more classical approaches based on ARIMA-based allocation schemes both in terms of under-allocation and over-allocation of available resources. Further, the FlexE agent could learn a wide variety of traffic patterns with different properties of mean and variance and was able to predict underlying patterns in incoming traffic. On the other hand, ARIMA-based methods would require recalculation of model parameters every time statistical properties of the input traffic patterns change.

It was also demonstrated that DRL can effectively circumvent the effects of latency in fronthaul via user offloading. The controller agent was able to offload users from the RU group having a path of higher latency in fronthaul to the RU group having a path with low latency in fronthaul. This offloading boosted the achievable user bit rate. The controller agent also learnt to symmetrically offload users in two directions depending on the load on the paths in fronthaul. Overall, both the controller agent and the FlexE agent were able to concurrently improve the system performance albeit without having a priori knowledge of the environment variable values. So it can be concluded that DRL can enable the network to autonomously improve its resource utilisation as well as improve achievable user bit rates.

The research in the Chapter suggests that for DRL agents to consistently increase rewards during training, fine-tuning reward functions and hyperparameters such as lookahead steps, discount factor, epsilon-greedy factor, experience buffer, and mini-batch size is necessary. While tuning can improve performance, it can also extend training time.

In the case of training FlexE agents, large experience buffers, equivalent to the total number of steps in the training process, were necessary to ensure agents consistently earned increasing rewards. It was observed that employing a very high discount factor near 0.99 and a large number of lookahead steps led to unstable training in the case of traffic predictions. However, decreasing these values resulted in more stable training and monotonically increasing rewards for FlexE agents. Therefore, it can be concluded that for environments with varying statistical properties, a large experience buffer, reduced lookahead steps, and a discount factor of less than 0.99 is required.

References

[1] M. N. Chughtai et al., ‘User and resource allocation in latency constrained Xhaul via reinforcement learning’, Journal of Optical Communications and Networking, vol. 15, no. 4, pp. 219–228, Apr. 2023, doi: 10.1364/JOCN.485029.

[2] F. G. Fernandes, R. da Rocha Lopes, and D. Z. Filho, ‘SINR Bounds for Broadcast Channels with Zero-Forcing Beamforming and Limited Feedback’, IEEE Transactions on Communications, vol. 60, no. 7, pp. 1772–1776, Jul. 2012, doi: 10.1109/TCOMM.2012.060112.100205A.

[3] D. Tse and P. Viswanath, Fundamentals of Wireless Communication. Cambridge University Press, 2005.

[4] N.-S. Kim and Y. H. Lee, “Effect of channel estimation errors and feedback delay on the performance of closed-loop transmit diversity system,” in 2003 4th IEEE Workshop on Signal Processing Advances in Wireless Communications - SPAWC 2003 (IEEE Cat. No.03EX689), Jun. 2003, pp. 542–545. doi: 10.1109/SPAWC.2003.1319019.

[5] 3GPP, “3GPP TR 38.801 V14.0.0 (2017-03): Study on new radio access technology: Radio access architecture and interfaces.” 3GPP, 2017.

[6] Kenjiro Cho, Koushirou Mitsuya and Akira Kato. "Traffic Data Repository at the WIDE Project". USENIX 2000 FREENIX Track, San Diego, CA, June 2000.

[7] V. Q. Rodriguez, F. Guillemin, A. Ferrieux, and L. Thomas, “Cloud-RAN functional split for an efficient fronthaul network,” in 2020 International Wireless Communications and Mobile Computing (IWCMC), Jun. 2020, pp. 245–250. doi: 10.1109/IWCMC48107.2020.9148093.

[8] B. Coll-Perales et al., ‘End-to-End V2X Latency Modeling and Analysis in 5G Networks’, IEEE Transactions on Vehicular Technology, vol. 72, no. 4, pp. 5094–5109, Apr. 2023, doi: 10.1109/TVT.2022.3224614.

[9] "Define Reward Signals - MATLAB & Simulink." https://www.mathworks.com/help/reinforcement-learning/ug/define-reward-signals.html (accessed Jan. 23, 2022).

[10] I. Oif-Flexe, "Flex Ethernet 2.2 Implementation Agreement," p. 51, Oct. 2021.

[11] S. R. Sinclair, T. Wang, G. Jain, S. Banerjee, and C. L. Yu, 'Adaptive Discretization for Model-Based Reinforcement Learning', in Proceedings of the 34th International Conference on Neural Information Processing Systems, in NIPS '20. Red Hook, NY, USA: Curran Associates Inc., Dec. 2020, pp. 3858–3871.

[12] X. Qian and D. Klabjan, "The Impact of the Mini-batch Size on the Variance of Gradients in Stochastic Gradient Descent." arXiv, Apr. 27, 2020. doi:10.48550/arXiv.2004.13146.

[13] Y. Liao, S. A. Hashemi, H. ElBakoury, J. Cioffi, and A. Goldsmith, "Calendar Allocation Based on Client Traffic in the Flexible Ethernet Standard," in ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Jun. 2020, pp. 1–6. doi: 10.1109/ICC40277.2020.9149293.

[14] P. J. Brockwell and R. A. Davis, Introduction to Time Series and Forecasting. Springer Science & Business Media, 2006.

Chapter 5

DRL-Assisted Transmit Signal Power Savings in Fronthaul

5.1 Introduction

In this Chapter, the simulation results of the physical layer transport of FlexE bit rates over a low-cost fronthaul segment based on discrete multitone modulation (DMT) using different advanced modulation formats are given. The results highlight the transmit DMT signal power saving resulting from traffic predictions via DRL and subsequent slot allocations in the FlexE calendar, as discussed in the previous Chapter 4.

In Section 5.2, the dependence of the power consumption of the transmitter in a short fibre optic communication link on the transmit signal power is discussed. Specifically, the importance of the power consumption of the modulator and drive amplifier is discussed which depends on transmit signal power. Section 5.2 also introduces the idea of optimising or reducing the power consumption via fronthaul bit rate reduction aided by DLR-based predictions.

In Section 5.3 the details of a fronthaul network architecture, with multiple stages of aggregation nodes, including FlexE nodes similar to Fig. 4.1 considered for the analysis of transmit DMT signal power savings via DRL are given. Section 5.3 describes the internal structure of FlexE aggregation nodes. Further, in Section 5.3 the simulation setup for FlexE transport via DMT is explained, including the use of different modulation formats and varying numbers of subcarriers.

In Section 5.4, two equalisation techniques, preamble-averaging-based equalisation, and zero-forcing equalisation are compared. It highlights the advantages of preamble-averaging-based equalisation in reducing the system error vector magnitudes (EVMs), compared to

ZF-based equalisation. For analysing the transmit DMT signal power savings in the fronthaul, preamble-averaging-based equalisation was used in the receiver to improve the receiver EVM for different FlexE bit rates and modulation formats.

In Section 5.5, the simulation results of a short DMT-type transmission link at different bit rates and modulation formats are presented. The transmit DMT signal powers, required bandwidths and sampling frequencies, for different bit rates and modulation formats are given. The transmit DMT signal powers are given for maintaining 3GPP standard EVMs for 5G mobile networks.

In Section 5.6, a model for the overall end-to-end SNR of the fronthaul link for different FlexE bit rates and modulation formats is presented. This Section 5.6 also discusses the achievable spectral efficiency of the DMT transmission link using different modulation formats using the modelled SNRs, as a means to benchmark it with Shannon's capacity.

In Section 5.7, the results for the reduction in transmit DMT signal power reduction using DRL-based traffic prediction are presented. The reductions in transmit DMT signal power consumption are compared with fixed slot allocation to prove the novelty of traffic prediction via DRL for transmit DMT signal power reduction.

5.2 Transmitter Power Consumption in DMT-Based Fibre Optic Fronthaul Links

In fronthaul links, electro-absorption modulators (EAMs), Mach-Zehnder modulators (MZM), directly modulated lasers (DML) and phase modulators (PM) may be used to modulate the optical carrier for data transmission, requiring drive signal amplification via a drive amplifier. In the current research, the analysis of power consumption is carried out for the case when MZM or EAM is used for carrier modulation. The modulator module's power consumption, including the drive amplifier, constitutes a significant portion of the overall transmitter power in a fibre optic transmission link [1],[2]. As an example, in [1], it was found that for a 100 Gbps/1100 km coherent optical fibre link, 32% of the total power consumption of the link was from the transmitter, of which 38% was due to the modulator unit.

For the case of a short DMT transmission link, such as the one under consideration discussed in Section 5.3.1, there would be less power consumption from the digital signal processing (DSP), and no power consumption in link amplifiers or for a receiver local oscillator (LO) laser. Excluding the power consumption from the local oscillator and in link amplifiers the power consumption from the modulator unit would be 15% of the total power consumption of the entire 100 Gbps/1100 km coherent fibre optic link in [1]. In the most extreme case of discounting all of the power consumption in DSP, link amplifiers and LO laser, the power consumption from the modulator and drive amplifier would rise to 28% of the total power consumption of the link. Of course, some DSP is required for a short DMT transmission link, but it was assumed that the contribution of the modulator and drive amplifier would be between 20% to 25% of the total power consumption of the link. The minimisation of the power consumption of the modulator unit will result in a reduction in the overall power consumption of the DMT transmission link.

Regarding power consumption in electro-optical modulators, studies in [3] and [4] highlight that for an MZM driven by 16-QAM signal, the power consumption of the modulator depends on the peak-to-peak voltage and power of the driving signal. The model for dynamic power consumption of the drive amplifier for an EAM discussed in [2], also indicates that it depends on the power of the driving signal, and is given as follows,

$$P_{drive} = \overline{V_{mod}}\ \overline{I_{mod}} + \frac{P_{V_{mod}}}{Z_o}, \tag{5.1}$$

where, P_{drive} is the power consumption of the driver amplifier, $\overline{V_{mod}}$ is the mean bias voltage of the EAM, $\overline{I_{mod}}$ is the mean photocurrent of EAM, $P_{V_{mod}}$ is the power of the signal required to drive the EAM. If it is assumed that the drive amplifier operates in the linear regime, then the power, P_{drive}, exhibits a linear dependence on the power of the input transmit DMT signal. When the power consumption of the drive amplifier is considered within the broader context of radio frequency (RF) amplifiers, it is revealed by studies in [5], [6] and [7] that their DC power consumption is significantly influenced by the amplitude and power of the input signal. While primarily addressing high input signal powers, the principles in these studies can be adapted for lower input signal powers.

When the fronthaul bit rates are lowered in a DMT-type transmission link such as via DRL-based traffic predictions, the transmit DMT signal powers, being input to the drive amplifier, required to maintain specific SNRs at the receiver, decrease. The reduction in the transmit DMT signal powers results in a reduction in the dynamic power consumption of the drive amplifiers as well as the modulator. When operating at lower bit rates with reduced peak-to-peak drive voltages, the power consumption of the drive amplifier can be further minimized by lowering the DC supply voltage. Notably, the drive amplifier for electro-optical modulators (EOMs), as specified in the datasheet [5], exhibits higher power consumption levels when driven by higher peak-to-peak and DC supply voltages. For example, it consumes 0.12W at 1.5V peak-to-peak voltage with a 3.3V supply and 0.5W at 2.3V peak-to-peak voltage with a 6V supply.

Studies in [8] and [9] have also analysed the impact of varying drive voltages through envelope tracking of input signals on reducing the DC power consumption of the RF amplifiers.

This concept is also demonstrated in commercially available products [10]. The dynamic variation occurs continuously, maintaining a slew rate below that of the input signal [11]. The envelope tracking of the input drive signal has been demonstrated with up to a bandwidth of 100 kHz in the studies in [12], which is well within the limits of variation of slot allocation with an interval of 4ms, By leveraging input traffic pattern tracking, the DC power supply for drive amplifiers can be reduced, contributing in the further reduction of the DC power consumption reduction of drive amplifier. Thus, the combination of the reduction of drive signal peak-to-peak voltage or power and DC supply voltage can lead to the reduction of the power consumption of the drive amplifier, which is a significant portion of the total power consumption of the transmitter.

Considering the dependence on the power consumption of the modulator and drive amplifier on the input drive signal power, one of the focuses of the thesis research is, therefore, on the reduction of the power of the input transmit DMT signal via the reduction in fronthaul bit rates aided by DRL-based traffic predictions.

Bandwidth variable transponders (BVTs) are emerging as a key technology for next-generation EONs [13]. Unlike traditional transponders with fixed data rates, BVTs offer dynamic modulation level variations and subcarrier allocation, enabling them to adapt their transmission capacity to fluctuating traffic demands. This flexibility translates to improved spectrum utilisation and reduced operational costs compared to static transponders that can suffer from wasted capacity [14]. This capability makes them particularly attractive for fronthaul networks, the critical link between RUs and the mobile core in cellular systems [15]. Fronthaul traffic can be dynamic [16], and BVTs have the potential to efficiently manage these fluctuations by varying the bandwidth and modulation levels.

Traffic prediction in conjunction with using BVTs which can vary the bandwidth used or vary the modulation level within a given bandwidth (varying spectral efficiency) can reduce the physical layer (PHY) bit rate. By predicting traffic, a PHY layer bit rate that can accommodate the predicted traffic, which is less than the maximum PHY layer bit rate (e.g., 100 Gbps), will suffice. The reduced PHY layer bit rate can be provisioned by allocating a varying number of 5 Gbps slots in the FlexE shim and a varying number of subcarriers or modulation levels in the BVT, following the predicted traffic. Reduction in PHY layer bit rate can lead to power consumption savings, as it will require less QAM levels which in turn requires less transmit DMT signal power to maintain receiver SNRs.

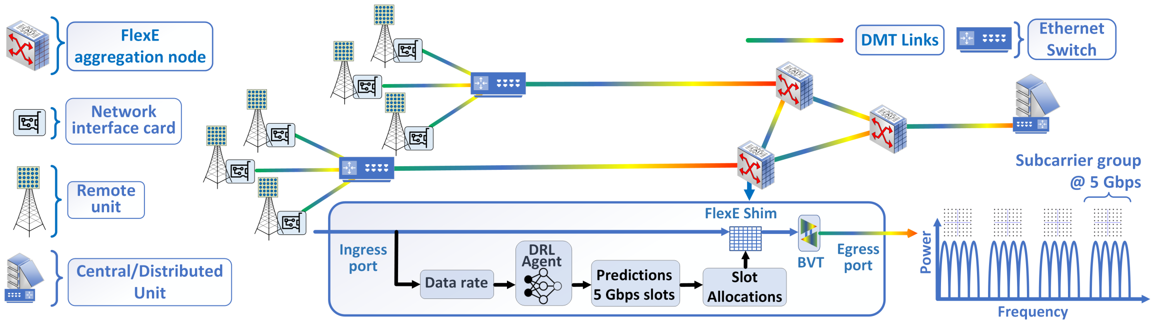


Fig. 5.1: Fronthaul for Beyond-5G network with FlexE aggregation and DRL.

5.3 Fronthaul Network and Simulation Model

As a reference for the results of the physical layer transport of FlexE slots, we assume a network architecture similar to Fig. 4.1 and shown in Fig. 5.1. The aggregation nodes in the fronthaul in Fig. 5.1 comprise two stages of aggregation. The first stage comprises conventional Ethernet, which aggregates the traffic from RUs and the second stage comprises of FlexE nodes, which aggregate traffic from the first stage Ethernet nodes. The traffic from second-stage FlexE nodes in Fig. 5.1 is aggregated by the third-stage FlexE node which is linked to a (CU/DU).

The internal structure of the FlexE aggregation nodes is shown in the lower part of Fig. 5.1. The bit rate of the input traffic is calculated by the 'data rate' block Fig. 5.1. The output of the 'data rate' block, shown in black arrows in Fig. 5.1, is then fed as input to the DRL agent in Fig. 5.1. The DRL agent predicts the bit rate of the aggregated traffic at the ingress port of the FlexE node. Based on the predictions, a varying number of 5 Gbps slots are allocated in the FlexE shim in Fig. 5.1 shown by the 'Slot Allocation' block. The BVTs within the FlexE nodes, shown in the lower part of Fig. 5.1, can then vary the bandwidth by changing the total number of subcarriers in the DMT multiplex to transport different FlexE bit rates or vary the modulation levels of different subcarriers. The transport of the 5 Gbps slots is carried out over different subcarrier groups of the DMT multiplex signal and is shown in the lower right-hand side of Fig. 5.1. The subcarrier groups are modulated using QAM. The modulation level can either be the same throughout the subcarrier groups or be different for different subcarrier groups. The predictions of the DRL agent enable the lowering of the average FlexE data rates on the DMT links in Fig. 5.1, thus contributing to the reduction in DMT signal power consumption, required to drive the BVTs.

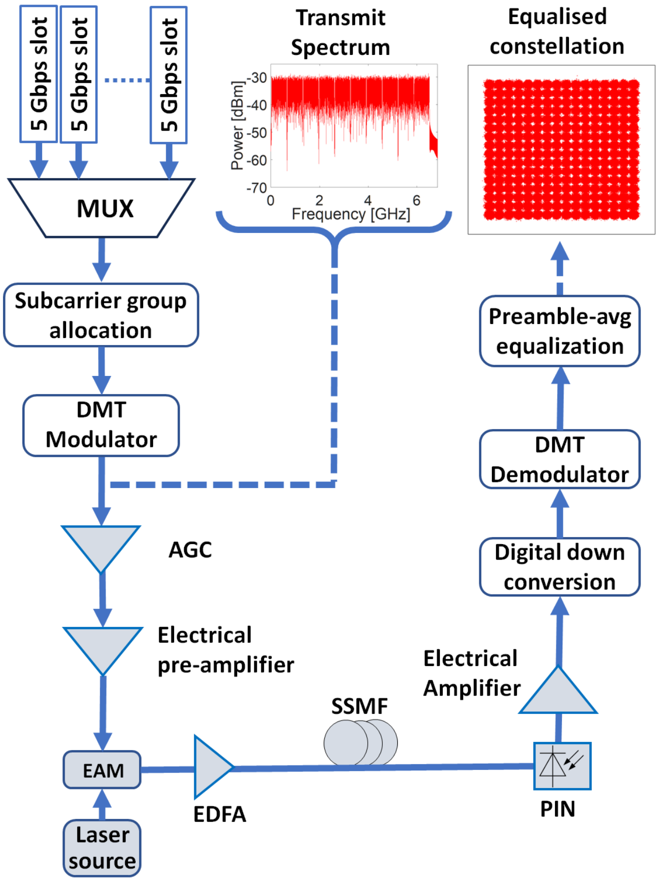


Fig. 5.2: Simulation setup in of DMT link in Fig. 5.1.

5.3.1 Simulation Setup for FlexE Transport via DMT

Within the overall 5G network architecture in Fig. 5.1, a single DMT transmission link in fronthaul was modelled with its block diagram shown in Fig. 5.2. The blocks shown in white were implemented in MATLAB®whereas the blocks shaded in grey colour were simulated in VPItransmission Maker®.

In Fig. 5.2, 5 Gbps slots were modelled in MATLAB® via independent pseudo-random bit sequences and then subsequently multiplexed. Each 5 Gbps bit stream was allocated to a group of 3800 subcarriers with a guard band of 24 MHz among them. The subcarrier spacing, Δf, was calculated via the equation given as follows,

$$\Delta f = 220 \times 10^3 \times \frac{6}{log_2 M}, \tag{5.2}$$

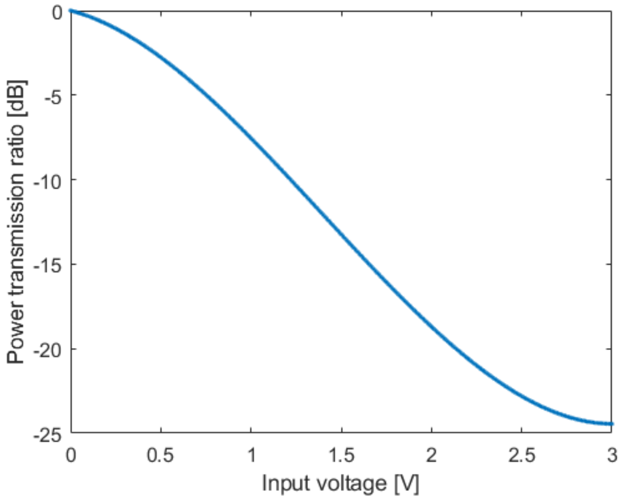


Fig. 5.3: Transmission magnitude of nonlinear EAM.

where, M is the modulation level being used. The factor of 220 kHz was used as a reference carrier spacing when 64 QAM modulation was used for the DMT signal transmission. The total number of subcarriers N_{total} were calculated using the following relation,

$$N_{total} = 3800 \times N_{groups} + N_{null\,carriers}, \tag{5.3}$$

where, N_{groups} is the number of 3800 subcarriers and $N_{null\,carriers}$ is the number of null subcarriers in the guard bands. The IFFT length N_{IFFT} for converting the frequency domain samples to the time domain for the DMT multiplex was calculated using the power of two of the ceil functions of the total number of subcarriers in Eq 5.3, given as follows,

$$N_{IFFT} = 2^{\lceil 2 \times N_{total} \rceil}. \tag{5.4}$$

The sampling frequency is thus the product of the number of IFFT points, N_{IFFT}, and subcarrier spacing, Δf, given as follows,

$$f_s = N_{IFFT} \times \Delta f. \tag{5.5}$$

Initially, all the subcarrier groups in the DMT multiplex were assigned the same modulation levels as it was assumed that the frequency response of the fibre optic channel was almost flat. Results were collected using QPSK, 16QAM, 64QAM and 256 QAM modulation of subcarrier groups. The length of the cyclic prefix was 12.5% of the DMT symbol length.

The simulation parameters for the modules simulated in VPItransmission Maker® shown in grey colour in Fig. 5.2 are given in Table 5.1. In VPItransmission Maker® the power of the

Table 5.1: Simulation Parameters in VPItransmission Maker®for the link in Fig. 5.2

Transmitter Frequency	193.1	THz
EAM bias voltage	0.4	Volts
EDFA gain	20	dB
EDFA max output power	19	dBm
EDFA noise figure	3	dB
Laser input power	10.5	dBm
Laser phase noise	1	kHz
Relative intensity noise (RIN)	-150	dB/Hz
Fibre length	10	km
Fibre Dispersion coefficient	16	ps/nm-km
Fibre Nonlinear coefficient	2.6x 10^{-20}	m^2/W
PIN diode responsivity	1.1	A/W
PIN diode thermal noise	0.33×10^{-10}	(A/$\sqrt{\text{Hz}}$)
AGC gain	-25	dB/W
Electrical pre−amplifier noise PSD	$8.62\text{x}10^{-12}$	(A/$\sqrt{\text{Hz}}$)
Rx electrical amplifier gain	19	dB
Rx electrical amplifier noise PSD	1.9143×10^{-10}	(A/$\sqrt{\text{Hz}}$)

modulated DMT signal was fixed by an automatic control gain amplifier (AGC) at -25 dBw, shown by a grey triangle labelled as 'AGC' in Fig. 5.2. The gain of the DMT signal output from the AGC was controlled by varying the gain of an 'Electrical pre-amplifier', shown by a grey triangle in Fig. 5.2. Noise with a power spectral density of $8.62\text{x}10^{-12}$ (A/$\sqrt{\text{Hz}}$) was added to the signal as a benchmark with real-world experiments [21]. The laser source in the link in Fig. 5.2 had an average transmit power of 8.5 dBm, an emission frequency of 193.1 THz, relative intensity noise (RIN) of -150 dB/Hz, a linewidth of 1 kHz, similar to an external cavity laser (ECL) in an experimental setup for benchmarking of simulations [22]. An EAM module in Fig. 5.2 was used for intensity modulation. The transmission polynomial as a function of input drive voltage, is given as follows

$$T_{dB}(V) = 0 - 2.75\,V - 6.3V^2 + 1.5V^3, \tag{5.6}$$

where, T_{dB} is the optical transmission in dB and V is the input drive voltage. The transmission polynomial is also plotted in Fig. 5.3. From the observation of Fig. 5.3, we can observe that the transmission magnitude of the EAM is a nonlinear function of input voltage. This transmission polynomial is partially linear in the range of (0∼1 V). The bias voltage was set at 0.4 V to avoid crossing the zero-voltage threshold. The parameter (α) introducing chirp in the input optical signal is also dependent on input drive voltage and the voltage-dependent polynomial is also given as follows,

$$\alpha(V) = 0.2 - 0.3V - 0.4V^2 + 0.1V^3. \tag{5.7}$$

The nonlinear transmission polynomial in Equation 5.7 was selected to represent real-world and commercially available EAM modules used in experiments [23], [24]. The frequency response of the EAM was assumed not to cause a limitation for the signal frequencies used of up to 34 GHz and EAMs with a 3dB bandwidth of up to 60 GHz are commercially available, such as that given in the datasheet in [25]. The drive amplifier was assumed to operate in the linear gain regime.

The output of the EAM was pre-amplified using an erbium-doped power amplifier (EDFA) having a noise figure (NF) of 3 dB, shown by a grey triangle in Fig. 5.2. The output optical power of EDFA was fixed at 19 dBm, as the analysis in the paper focused on the input power saving of the DMT-modulated signal. The signal was propagated through a standard single-mode fibre (SSMF) having a length of 10 km. The optical fibre link had an attenuation of 0.2 dB/km, dispersion coefficient of 16 ps/nm-km and nonlinear index of 2.6x 10^{-20} m^2/W and listed in Table 5.1.

At the receiver, the signal was detected using a positive-intrinsic-negative (PIN) photodiode shown in Fig. 5.2, for which the thermal noise was set at 0.33x10^{-10} (A/$\sqrt{\mathrm{Hz}}$), to model the load resistor of the photodiode, and shot noise enabled. The received signal was amplified by an electrical amplifier having a gain of 19 dB. Noise having a spectral density of 1.9143x10^{-10} (A/$\sqrt{\mathrm{Hz}}$) was added to the signal as a benchmark with real-world experiments [21]. In MATLAB® the received DMT multiplex signal was sampled via Nyquist zone de-mapping as discussed in Section 2.6.3 and [22], as well as shown in Fig. 5.2. Each de-mapped Nyquist zone or subchannel in the DMT multiplex, also shown in Fig. 2.25, represented a 5Gbps slot for FlexE.

5.4 Zero Forcing and Preamble-Averaging Equalisation for DMT Type Signal Transmission

The equalisation of the received signal in DMT-type signal transmission is categorized into two categories, which include time domain equalisation (TEQ) and frequency domain equalisation (FEQ) [26]. In time domain equalisation the received signal is equalised via optimal finite impulse response (FIR) filtering. The received signal is equalised on a ‘per-tone; basis in the frequency domain. The equalisation of the DMT-type signal at the receiver in the frequency domain can be carried out using zero-forcing equalisation (ZF), and minimum mean square equalisation (MMSE) [26], [27], [28]. The received signal can also be equalised using decision feedback equalisation (DFE) [27],[28].

The research in the thesis utilised preamble averaging, with the details also discussed in Equation 2.26 in Section 2.6.2. The preamble averaging can be considered an FEQ technique carrying out sub-optimal equalisation per tone. The main advantage of this technique is that it eliminated the unequal variance of noise in the post-equalisation QAM signal constellation observed using ZF equalisation, as discussed in Section 2.6.2, resulting in a decrease in the EVM of the DMT-based fronthaul link in Fig. 5.2. In the practical implementation of ZF equalisation [22], interpolation of frequency response is required for the subcarriers where the pilot signals are not transmitted. The interpolation operation will not be required in preamble-averaging as the pilot signals are transmitted over all the subcarriers, thus reducing the potential complexity of the equalisation operation.

The novelty of the thesis research lies in the direct comparison between Zero Forcing and preamble averaging equalisation, particularly in the context of DMT-based fronthaul links. While ZF equalisation is well-documented in the literature for its simplicity, the use of preamble averaging as an alternative FEQ technique has not been extensively compared.

Equalisation techniques based on optimal per-tone filtering [26], MMSE-based equalisation in the time and frequency domain [29] or based on deep learning such as convolutional neural networks [30], may outperform the preamble averaging technique, however, the aim of the thesis research was not to develop a new equalisation method from ground zero, but rather achieve the standard 3GPP EVMs at the receiver at high data rates of up to 200 Gbps, for the link shown in Fig. 5.2. Utilisation of equalisation techniques outperforming preamble averaging will not affect the results presented in this Chapter in Section 5.7, as the relative gains in transmit DMT signal power would remain almost the same.

Table 5.2: Simulation Parameters in MATLAB®for the link in Fig. 5.2.

Modulation levels	64	
Number of data sub-carriers pep group	3800	
Number of data sub-carrier groups	4	
Subcarrier spacing	220	kHz
Guard band spacing	24	MHz
IFFT length	32768	
Sampling frequency	7.209	GHz
Number of transmitted DMT symbols	50	
Pilot symbol spacing	10	subcarriers
Preamble sequence length	30	symbols
Total data rate	20	Gbps

5.4.1 Simulation results for ZF and Preamble-Averaging equalisation

The transmission of varying FlexE data rates with graduations of 5 Gbps using DMT transmission may require a large number of subcarriers, modulated by advanced modulation formats such as 256-QAM for bandwidth efficiency. In such cases, mitigation of overall link impairments via equalisation of the received signal at the receiver will be required to reduce the EVMs. As mentioned in Section 2.6.2, the equalisation based on preamble averaging can reduce the unequal variance of noise in the received signal constellation (M-QAM) resulting in the reduction of EVM of the DMT link. This can result in savings in the transmit DMT signal power required to maintain the EVMs for different data rates and modulation levels according to the 3GPP standards for 5G networks [31]. Since one of the emphases of the thesis research is on the savings in transmit DMT signal power so, the preamble-averaging-based equalisation will be used in the receiver.

As a starting point, the performance of the preamble-averaging-based equalisation was compared with pilot-based ZF equalisation. It was tested by transmitting and receiving a DMT signal in a back-to-back (b2b) configuration in a MATLAB® environment only. The performance of the two equalisation techniques was tested via the addition of Additive white Gaussian noise (AWGN) with different levels of power spectral density (PSD), resulting in varying levels of SNRs. The parameters of the DMT modulation in MATLAB®are given in Table 5.2.

The transmitted DMT signal had 4 subcarrier groups, as shown in the lower right-hand side of Fig. 5.1, where each group represented a 5 Gbps slot in FlexE. The spacing among the subcarriers of the DMT signal was 220 kHz. 64 QAM was used to modulate the input data streams. The number of subcarriers per subcarrier group was 3800. The subcarrier

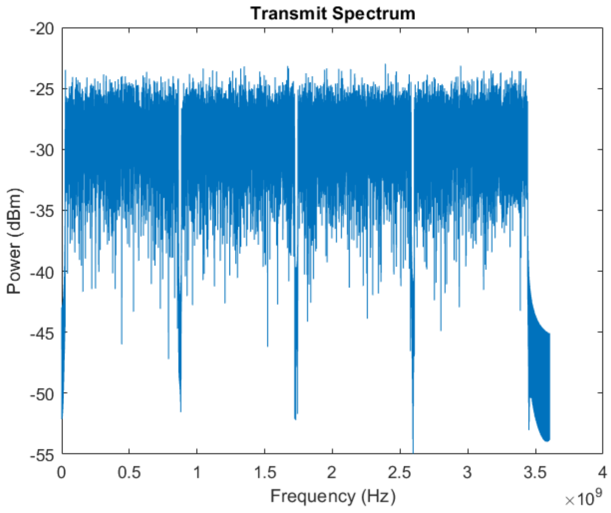
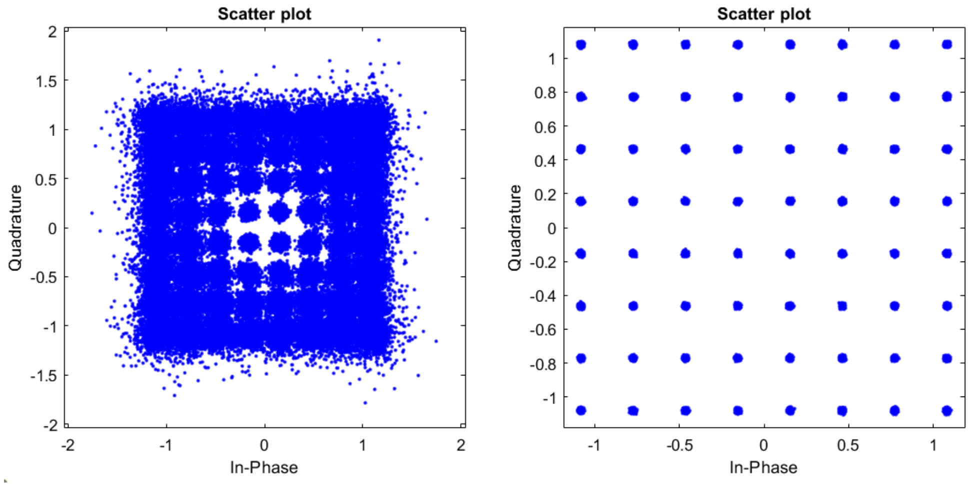


Fig. 5.4: Spectrum of Transmitted DMT signal @ 20 Gbps.



(a) **(b)**

Fig. 5.5: Constellation diagram (64-QAM) after ZF equalisation (a), after preamble averaging equalisation at 50 dB SNR (b).

groups were separated by guard bands having a bandwidth of 24 MHz. The resulting overall data rate was 20 Gbps.

For the case of ZF equalisation, every 10th subcarrier in the DMT multiplex was used to transmit pilot symbols. The pilot symbols comprised random symbols from the 64 QAM symbol set. For preamble averaging-based equalisation a total of 50 DMT symbols were transmitted out of which 30 symbols were used to transmit preambles. The preambles comprised random symbols from the 64 QAM symbol set. The channel estimation was carried out by averaging the estimates of the channel from every preamble sequence as indicated in Equation 2.26.

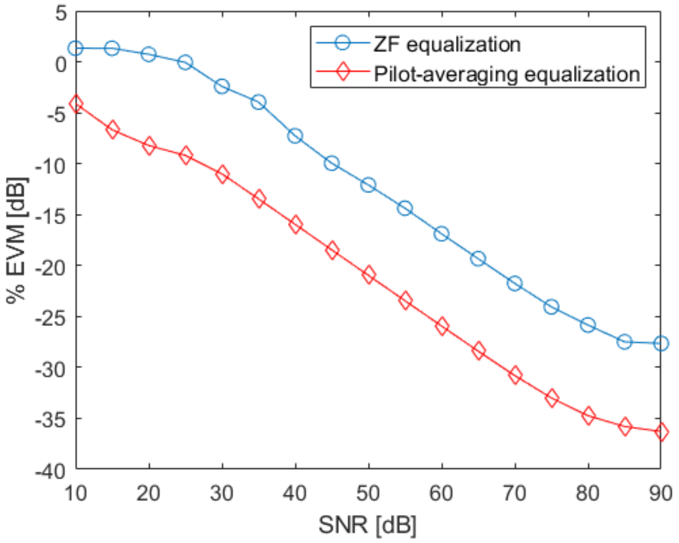


Fig. 5.6: EVMs vs SNR for ZF and preamble-averaging equalsation.

As an example, AWGN noise was added to the DMT signal such that SNR with respect to the power of transmitted DMT signal in frequency domain was 50 dB, in a b2b configuration in MATLAB® environment only. The spectrum of the transmitted DMT signal is shown in Fig. 5.4. The constellation diagrams using ZF and preamble-averaging equalisation techniques are shown in Fig. 5.5. From the observation of Fig. 5.5, we can conclude that the constellation diagram after ZF equalisation results in an unequal variance of noise with an EVM of 6.1% whereas the constellation diagram after preamble-averaging equalisation results in equal and less variance of the noise, resulting in an EVM of 0.8%.

ZF equalisation and preamble-averaging-based equalisation were also compared for different levels of SNRs in a b2b configuration in MATLAB® environment only. The results for the comparison of the two equalisation techniques in a b2b configuration are given in Fig. 5.6 for different values of SNRs. From the observation of Fig. 5.6, the preamble-based equalisation outperforms ZF equalisation by up to 5.3% between SNRs of (20 dB $\sim$ 50 dB). Considering the better performance of preamble-averaging-based equalisation, the thesis research will deploy it for the transport of FlexE through DMT transmission links.

The performance of preamble-averaging-based equalisation was tested in the co-simulation environment of VPItransmission Maker® and MATLAB®. The transmission of a DMT-type signal was simulated through the link shown in Fig. 5.2. The simulation setup parameters are given in Table 5.1 and Table 5.2. The DMT signal had an overall data rate of 20 Gbps. The simulation setup in VPItransmission Maker®is given in the following Fig. 5.7.

The constellation diagrams of the equalised signal after propagation through the link shown in Fig. 5.7 are given in Fig. 5.8. The constellation diagrams depict the impact of preamble averaging-based equalisation and zero-forcing equalisation. The EVM from ZF equalisation was 11.33 % as shown in Fig. 5.8 (a) and the EVM after pilot-averaging-based equalisation was 6.63 % as shown in Fig. 5.8 (b). From the observation of Fig. 5.8 (a) and Fig. 5.8(b) we can conclude that the pilot-averaging equalisation outperforms the zero-forcing equalisation technique.

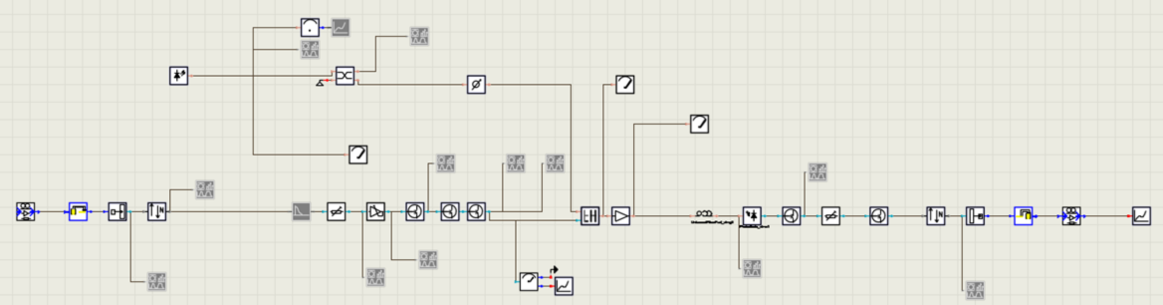
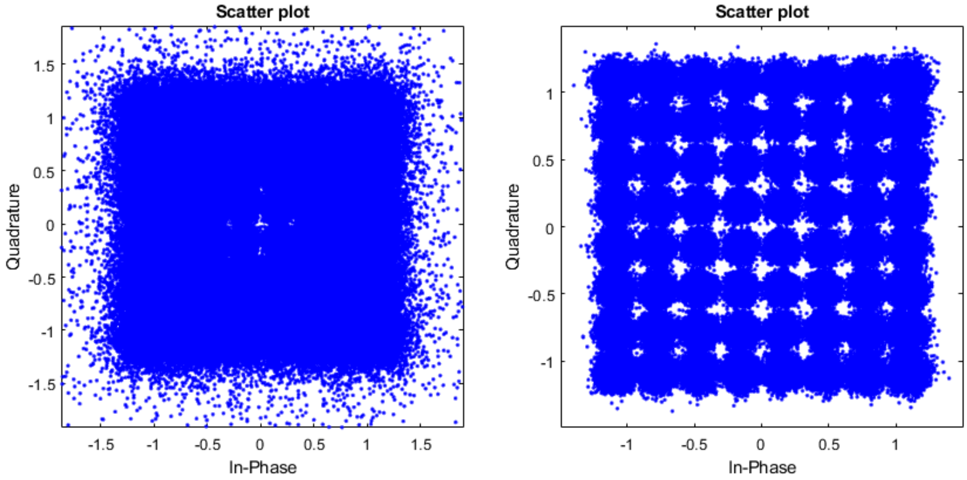


Fig. 5.7: VPItransmission Maker®setup for DMT typ signal transmission.



(a) **(b)**

Fig. 5.8: Constellation diagram (64-QAM) after ZF equalisation (a), after preamble averaging equalisation (b) after the link in Fig. 5.7.

Table 5.3: EVM for the different number of preamble symbols.

Number of preamble symbols	Number of data symbols	EVM (%)
10	50	7.7
20	50	7.3
30	50	7.1
40	50	7.0
50	50	7.0

The performance of the preamble averaging-based equalisation was also tested for its dependence on the number of preambles used for averaging the channel estimates. The performance was assessed at a data rate of 100 Gbps using the parameters listed in Table 5.2 and Table 5.1. The number of DMT symbols as preambles was varied whereas the number of DMT symbols as data symbols was fixed at 50 symbols. The EVMs dependent on the number of preamble symbols are listed in the following Table 5.3. From the analysis of Table 5.3 we can conclude that as the number of preamble symbols increases the resulting EVMs also decrease, however, the reduction stalls after 40 preamble symbols.

5.5 Transmit DMT Signal Powers for Different Bit Rates in Fronthaul

The performance of the transmission link in Fig. 5.2 was tested, via simulation in VPItransmission Maker®, at different FlexE data rates of 5, 10, 20, 50 and 200 Gbps using modulation formats of 4, 16, 64 and 256 QAM, with preamble-averaging-based equalisation applied at the receiver. The optical output power of the EDFA was fixed at 19 dBm. The power of the transmit DMT signal at the output of the ‘Electrical pre-amplifier’ before the EAM in Fig. 5.2 was varied such that the receiver maintains the standard EVMs for different modulation formats in the 3GPP standard for 5G wireless networks [31].

As an example of the simulation results, the spectrum of the transmitted DMT type signal, through the link in Fig. 5.2, at 50 Gbps and using 256 QAM modulation is shown in Fig 5.9. The received signal constellation after preamble-averaging-based equalisation is given in the following Fig. 5.10. As discussed earlier preamble-averaging-based equalisation resulted in a uniform variance of the noise, resulting in a reduction of overall EVM.

The transmit DMT signal powers required for the transmission of different FlexE bit rates via DMT signal transmission, having different modulation formats, subcarrier group bandwidths and sampling frequencies are given in Table 5.4 in the 9th column. Note that for

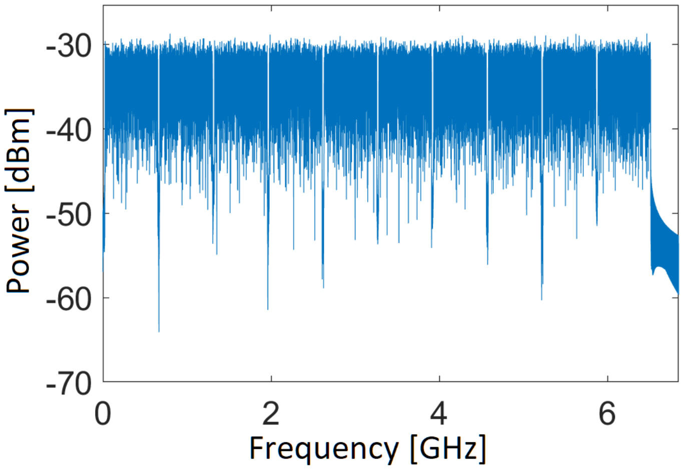


Fig. 5.9: Spectrum of transmitted DMT signal at 50 Gbps (10 FlexE slots).

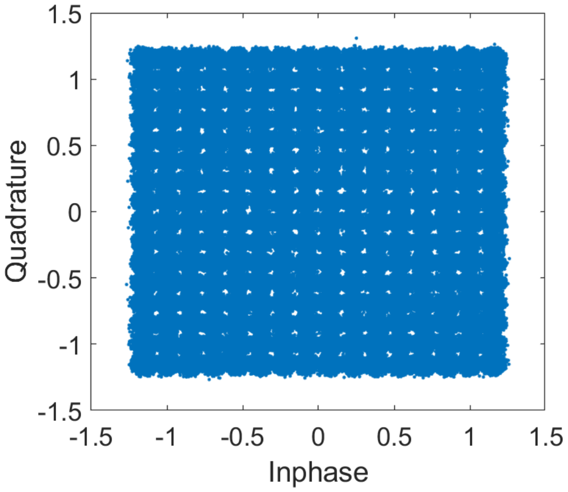


Fig. 5.10: Constellation diagram of 256 QAM signal after preamble averaging based equalisation.

the results in Table 5.4, the number of subcarriers in each subcarrier group was fixed at 3800 and the transmit signal powers were measured at the input of the EAM in Fig. 5.2. From the observation of Table 1, we can infer that the transmit DMT signal power savings in moving from 200 to 5 Gbps are 20.26 dB using 256 QAM modulation, 16.24 dB using 64 QAM, 16.21 dB using 16 QAM and 16.02 dB using 4 QAM. Similarly, a reduction in signal input power of 12.46 dB was observed at the data rate of 200 Gbps when using 4 QAM instead of 256 QAM.

Table 5.4: Transmit DMT signal power for the transmission of FlexE bit rates using 4,16,64 and 256 QAM modulation.

Modulation level	Bit rate	5 Gbps slots	Subcarriers per group	Subcarrier bandwidth	Mux bandwidth	Sampling frequency	EDFA Transmit power	Input DMT signal power	EVM
	(Gbps)			(kHz)	(GHz)	(GHz)	(dBm)	(dBm)	%
256	5	1	3800	165	0.68	2.70	19	-10.2	3.49
	10	2	3800	165	1.33	2.70	19	-7.6	3.478
	50	10	3800	165	6.53	21.63	19	-0.397	3.49
	100	20	3800	165	13.04	43.25	19	3.005	3.485
	150	30	3800	165	19.55	43.25	19	5.405	3.497
	200	40	3800	165	26.06	86.51	19	10.1	3.493
64	5	1	3800	220	0.89	1.80	19	-16.4	7.9
	10	2	3800	220	1.75	3.60	19	-13.6	7.89
	50	10	3800	220	8.62	28.84	19	-6.596	7.9
	100	20	3800	220	17.22	57.67	19	-3.593	7.95
	150	30	3800	220	25.82	57.67	19	-1.593	7.98
	200	40	3800	220	34.42	115.34	19	-0.1858	7.73
16	5	1	3800	330	1.30	2.70	19	-18.6	12.4
	10	2	3800	330	2.58	5.41	19	-15.8	12.47
	50	10	3800	330	12.80	43.25	19	-8.79	12.48
	100	20	3800	330	25.58	86.51	19	-5.389	12.3
	150	30	3800	330	38.36	86.51	19	-3.189	12.5
	200	40	3800	330	51.14	173.02	19	-2.578	12.42
4	5	1	3800	660	2.56	5.41	19	-18.6	17.3
	10	2	3800	660	5.09	10.81	19	-15.8	17.52
	50	10	3800	660	25.34	86.51	19	-8.3892	17.49
	100	20	3800	660	50.66	173.02	19	-5.378	17.01
	150	30	3800	660	75.98	173.02	19	-3.38	17.5
	200	40	3800	660	101.30	346.03	19	-2.357	17.34

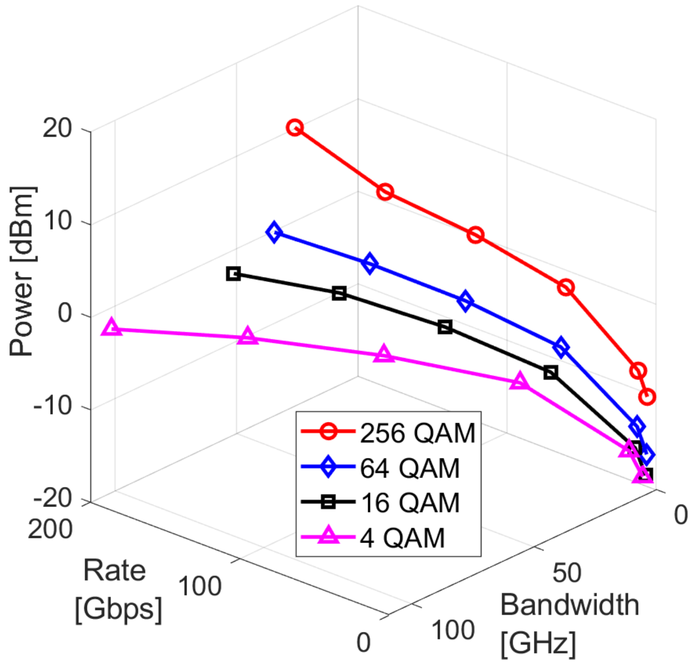


Fig. 5.11: DMT signal powers for different fronthaul bit rates and DMT multiplex bandwidths for the link in Fig. 5.2

The data presented in Table 5.4 is also plotted in Fig. 5.11, in the form of transmit DMT signal powers dependent on FlexE data rates and overall DMT signal bandwidths. From the analysis of Fig. 5.11, we can observe the expected trend of increasing transmit DMT signal power with increasing DMT signal bandwidth or FlexE bit rates.

The performance of the link in Fig. 5.2 was also benchmarked within limited overall bandwidths of the transmitted DMT signal. To transmit different FlexE data rates within fixed bandwidth limits, combinations of 5 Gbps subcarrier groups with different modulation levels were chosen. The total transmit DMT signal powers, in this case, were estimated with the assumption that the total power is the sum of individual powers of 5 Gbps subcarrier groups, plotted in Fig. 5.11.

The estimated transmit DMT signal powers for the limited bandwidths of 18 GHz, 26 GHz and 34 GHz are shown in Fig. 5.12. It can be observed that as the bandwidth limit is reduced from 34 GHz to 18 GHz, the total transmit DMT signal power increases as it requires the selection of 5 Gbps channels with higher modulation levels. It can also be observed from Fig. 5.12 that the total transmit DMT signal power saving in reducing the data rate from 100 Gbps to 40 Gbps is around 70% for the 18 GHz limit, 68% for the 26 GHz limit, and 60% for the 34 GHz limit.

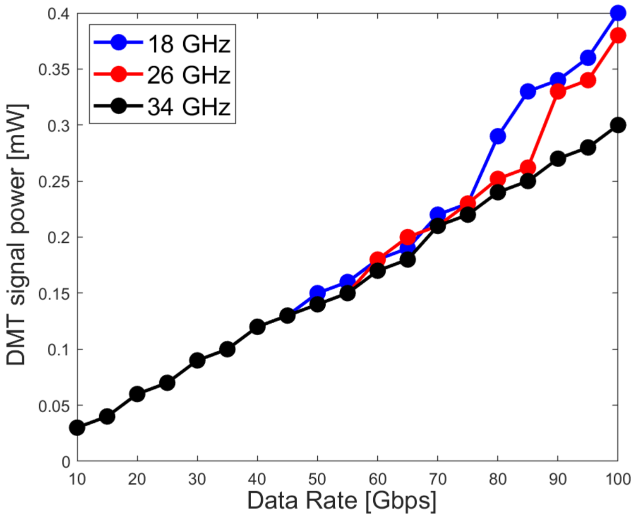


Fig. 5.12: Estimated DMT signal powers for different fronthaul bitrates and fixed DMT multiplex bandwidths for the link in Fig. 5.2.

5.6 SNR and Capacity for DMT-Based Fronthaul Link

In this Section the performance of the DMT transmission link in Fig. 5.2 was assessed by comparing its spectral efficiency at different bit rates and modulation formats in Table 5.4 with theoretical achievable Shannon capacity [32]. This was done by estimating the SNRs at the receiver of the link as a function of the transmit DMT signal powers input to the EAM in Fig. 5.2.

The input DMT signal power driving the EAM in the link in Fig. 5.2 was calculated as follows,

$$P_{DMT} = P_{AGC}G_1, \tag{5.8}$$

where P_{DMT} is the transmit DMT signal power driving the EAM, P_{AGC} is the output power of the 'AGC' amplifier in Fig. 5.2 and, G_1 is the gain of the 'Electrical pre-amplifier' in Fig. 5.2. Note that the transmit DMT signal power is swept by varying the gain, G_1.

The small signal power at the output of the EAM in Fig. 5.2 is the power of the transmit DMT signal in the optical domain, is given as follows [33],

$$P_{out,EAM} = P'_{cw}T_{lin}(V_b)'P_{DMT}, \tag{5.9}$$

where, P'_{cw} is the constant average output optical power of EAM in Fig. 5.2 defined as follows,

$$P'_{cw} = T_{lin}(V_b)P_{cw}, \tag{5.10}$$

where, P_{cw} is the power of the 'laser source' in Fig. 5.2, T_{lin} is the transmission polynomial of EAM, in Fig. 5.2, in linear scale and $T_{lin}(V_b)'$ is the 1st order derivative of T_{lin} at the bias voltage, V_b, is defined as follows,

$$T_{lin}(V_b)' = \left.\frac{\partial(T_{lin})}{\partial v}\right|_{V_b}. \tag{5.11}$$

The power of the transmit DMT signal in the optical domain, $P_{out,EAM}$, is amplified by an 'EDFA' block in Fig. 5.2, attenuated by the fibre optic channel and detected by the photodiode, labelled as 'PIN' diode in Fig. 5.2. The power of the received DMT signal after photo-detection is given as follows [34],

$$P_{rec} = \eta^2(P_{out,EAM}G^2_{EDFA}G^2_{fibre}), \tag{5.12}$$

where, η is the responsivity of the photodetector, G_{EDFA} and G_{fibre} is the gain of the EDFA and fibre optic channel respectively. The noise at the receiver is assumed to be the sum of noise from electrical amplifiers, signal-spontaneous emission beat noise, thermal noise, shot noise and relative intensity noise. The noise from the electrical amplifiers detected at the photodiode receiver is given as follows [34],

$$\sigma^2_{elect} = \eta^2(N_{out,EAM}G^2_{EDFA}G^2_{fiber}), \tag{5.13}$$

where,

$$N_{out,EAM} = P'_{cw}T_{lin}(V_b)'N_{input}, \tag{5.14}$$

where, $N_{out,EAM}$ is the power of noise at the output of EAM, N_{input} is the input noise PSD from the 'Electrical pre-amplifier' at the transmitter in Fig. 5.2. The noise due to signal-spontaneous beat noise is given as follows [34],

$$\sigma^2_{sig-spon} = 4\ \eta^2 G_{EDFA}P'_{cw}\ S_{ASE}\Delta f, \tag{5.15}$$

where, Δf is the bandwidth of the DMT multiplex and, S_{ASE} is the amplified spontaneous emission (ASE) noise spectral density of the EDFA in Fig. 5.2 defined as follows [34],

$$S_{ASE} = (G_{EDFA} - 1)Fhv/2, \tag{5.16}$$

where, F is the noise figure of the EDFA in Fig. 5.2, h is the plank's constant and, v is the central frequency of EDFA. The shot noise at the PIN diode in Fig. 5.2 is given as follows [34],

$$\sigma_{shot}^2 = 2eI_{dc}\Delta f, \tag{5.17}$$

where I_{dc} is the direct current (DC) at the receiver defined as follows,

$$I_{dc} = \eta P_{cw}' G_{EDFA} G_{fiber}, \tag{5.18}$$

Thermal noise is defined as follows,

$$\sigma_{ther}^2 = S_{ther}\Delta f, \tag{5.19}$$

where S_{ther} thermal noise spectral density of the PIN diode in Fig. 5.2. The RIN detected at the receiver PIN diode in Fig. 5.2 in the receiver is given as follows [35],

$$\sigma_{RIN}^2 = K_{RIN} I_{dc}^2 \Delta f', \tag{5.20}$$

where, K_{RIN} is the RIN noise density and, $\Delta f'$ here is the spectral width of the laser source Fig. 5.2. The noise added from the electrical amplifier in the receiver after the PIN diode in Fig. 5.2 is defined as follows,

$$\sigma_{elect,rx}^2 = N_{psd,rx}\,\Delta f, \tag{5.21}$$

where, $N_{psd,rx}$ is the PSD of the 'Electrical amplifier' after the PIN diode in Fig. 5.2. The total noise added in the detected DMT multiplex signal at the receiver is defined as the sum of all noise sources and given as follows [34],

$$\sigma_{total}^2 = \sigma_{elect}^2 + \sigma_{sig-spon}^2 + \sigma_{shot}^2 + \sigma_{ther}^2 + \sigma_{RIN}^2 + \sigma_{elect,rx}^2. \tag{5.22}$$

The SNR at the receiver of the DMT link is given as follows,

$$SNR = \frac{P_{rec}}{\sigma_{total}^2}. \tag{5.23}$$

The achievable Shannon capacity of the link is given as follows [32],

$$\eta_c = log_2\left(1 + SNR\right), \tag{5.24}$$

where SNR is the signal to noise ratio estimated in Equation 5.23.

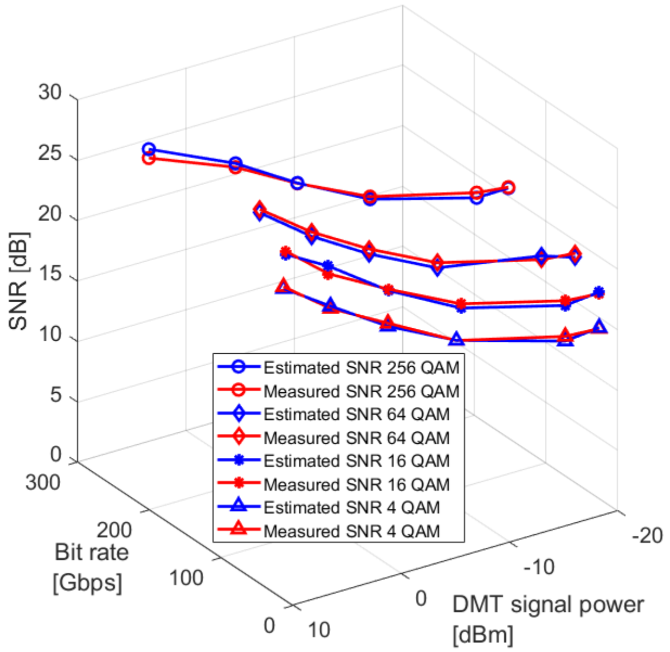


Fig. 5.13: Estimated and measured SNRs vs input DMT signal powers.

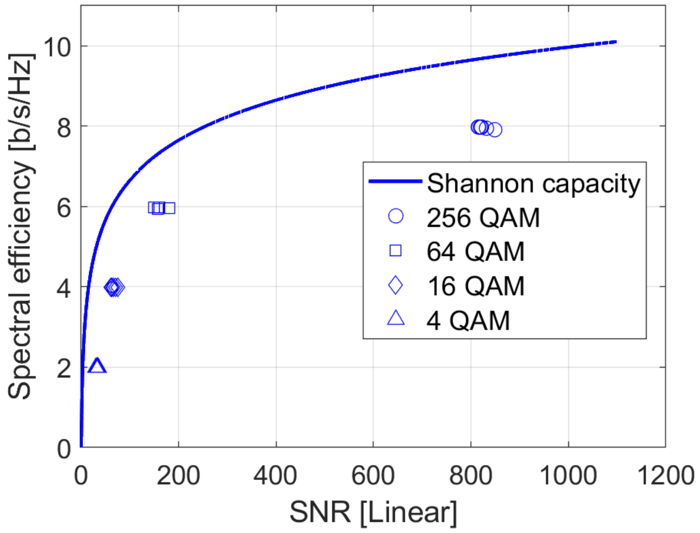


Fig. 5.14: Spectral efficiency vs SNRs of DMT transmission link in Fig. 5.2.

The results for the estimated SNRs using Equation 5.23, the simulation parameters listed in Table 5.1 and the measured SNRs from simulations results in Table 5.4, are given in Fig. 5.13. The estimated SNRs are shown in blue lines and the SNRs from simulations are shown in red lines for the data rates of (5 ∼ 200 Gbps) and using 4, 16, 64 and 256 QAM modulation schemes. From the observation of Fig. 5.13, we can conclude that the estimated

SNRs follow the trend of the measured SNRs for different input DMT signal powers to maintain the standard 3GPP SNRs.

The spectral efficiencies of the link in Fig. 5.2 were calculated for the data rates (($5 \sim 200$ Gbps) and modulation formats (4, 16, 64 and 256 QAM) listed in Table 5.4. The spectral efficiencies were calculated as the ratio of bit rates in the second column and the multiplex bandwidths in the sixth column in Table 5.4. The calculated spectral efficiencies versus the corresponding estimated SNRs, using Equation 5.23 in Fig. 5.13, are plotted in Fig 5.14. The calculated spectral efficiencies were also compared with the spectral efficiencies using the Shannon capacity using Equation 5.24 and are plotted in Fig 5.14. From the analysis of Fig 5.14, we can observe that the link in Fig. 5.2 achieves spectral efficiency which is approximately 1 [bit/sec/Hz] less than the spectral efficiency from Shannon capacity, for all the data rates and modulation formats. The difference might be due to the distortions in the EAM and post-equalisation residual noise from electrical and optical components which indicates that there exists a small margin for further improvement in system performance.

5.7 Results for Reinforcement Learning Assisted Transmit DMT Signal Power Savings

Traffic prediction in a fronthaul network can be carried out using classical prediction techniques such as ARIMA as proposed in [17], with an introduction given in Section 4.4.2. The prediction of traffic patterns through ARIMA requires fine-tuning of its parameters such as the length of filter taps and their coefficients, which depend on the statistical properties of the input traffic pattern. If there is a variation in the statistical traffic patterns, then it will require re-tuning of the parameters for ARIMA.

On the other hand, DRL can also be used to predict traffic patterns and subsequent slot allocations as proposed in Section 4.4. The advantage of predicting traffic patterns via a DRL-based agent is that it can be trained on different traffic patterns with different statistical properties of the mean and standard deviations. After training, the DRL agent will not require re-training even if the statistical properties of the input pattern change with time. The action space of the DRL agent can be set as the number of slot allocations in the FlexE shim following the incoming traffic patterns. The results in Section 4.4 have shown that DRL-based prediction can lead to better utilisation of the available bandwidth in terms of over and under-allocation of bandwidth in the physical layer, in comparison to ARIMA-based traffic predictions and slot allocations. The thesis research, therefore, focuses on using DRL for fronthaul bit rate reduction as well as a reduction in the power of the transmit DMT signal.

As mentioned in Section 4.2, the aim of the DRL agent in the FlexE aggregation node in Fig. 4.2 is to carry out predictions with a granularity of 5 Gbps, and aid subsequent slot allocations in the FlexE shim. Utilising the results in Section 4.4 for the training of a DRL agent for the prediction of the input traffic pattern from the widely integrated distributed environment (WIDE) project and subsequent slot allocation in a FlexE shim, the transmit DMT signal power required for the transmission of 5 Gbps slots was calculated for different bandwidth limits. Note that the observation space, action space, Deep-Q network training parameters, and the reward functions for the training of the DRL agent are described in detail in Section in 4.4 and Table 4.6. The deployment results of the trained DRL agent are shown in Fig. 5.15(b) in terms of predictions of the traffic pattern with a granularity of 5 Gbps. Similar to the results presented in Fig. 4.27, it can be observed in Fig. 5.15(b) that the DRL attempts to follow the input pattern yet there are under allocations.

The transmit DMT signal powers for the FlexE data rates in Fig. 5.15(a) were calculated using transmit DMT signal powers given in Fig. 5.12 for different bandwidth limits. Then the mean DMT signal power was calculated for the overall slot allocation interval of 24 seconds. The mean DMT signal powers for bandwidth limits of 18 GHz, 26 GHz and 34 GHz are shown in Fig. 5.16. The mean powers are also compared to the case where the number of 5 Gbps slots is fixed and equivalent to the maximum input traffic of 85 Gbps in Fig. 5.15(a). It can be observed that for each bandwidth limit, the mean DMT signal powers for DRL-based slot allocation are reduced compared to the DMT signal powers for the fixed allocation. The maximum power reduction of 34.6% is achieved for the bandwidth of 18 GHz with the lowest power reduction of 22.3% achieved for the bandwidth of 34 GHz. Thus it can be concluded that the tracking of input traffic via DRL results in the reduction of the FlexE bit rates, contributing to the reduction in DMT signal power required to drive the EAM in Fig. 5.2.

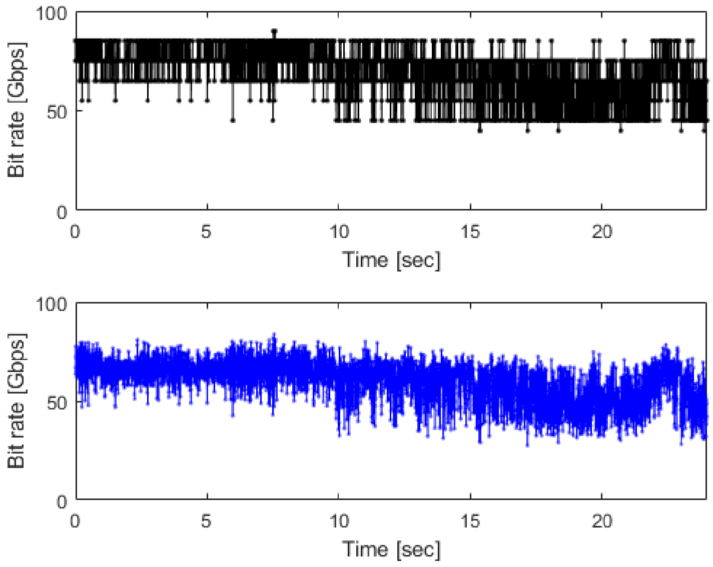


Fig. 5.15: Fronthaul bit-rate predictions from DRL agent (a), input traffic bit-rate from the WIDE project [17] (b).

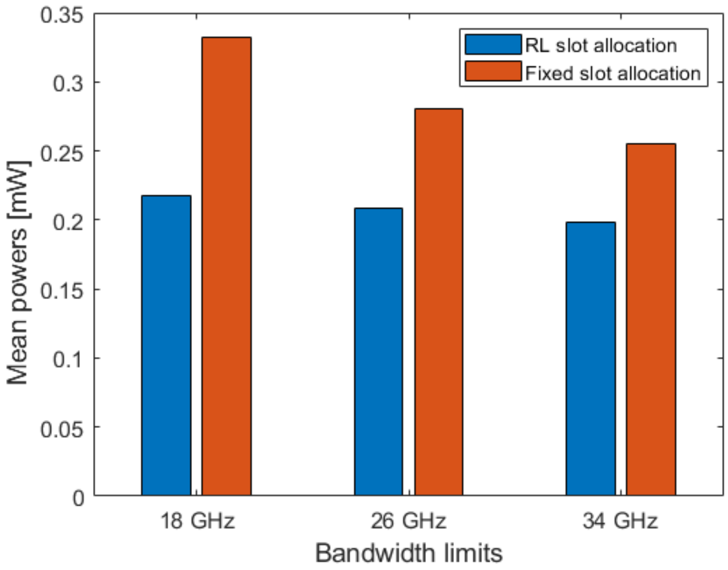


Fig. 5.16: Mean DMT signal powers for fixed and DRL-based slot allocations.

5.8 Conclusions

The research in the Chapter explored the physical layer transmission of the FlexE slots of 5 Gbps over fronthaul links in beyond-5G mobile networks based on DMT-type signal transmission. The transmission of FlexE slots was enabled by BVTs which can vary the overall physical layer bandwidth and modulation levels of different subcarriers of the DMT multiplex. The research in the Chapter also assessed the advantages of reducing the fronthaul bit rate from the maximum PHY bit rate via the allocation of slots in varying numbers which were determined by the traffic predictions from a trained DRL agent. The variation resulted in the reduction of transmit signal power which is pivotal for the reduction of power consumption of the modulator unit of a DMT-type transmission system comprising.

For the analysis, a short link DMT-type transmission system was simulated which was aided by preamble-averaging equalisation at the receiver. The simulation results indicate that the preamble-averaging equalisation outperformed the ZF equalisation in terms of percentage EVM. The utilisation of preamble-averaging equalisation resulted in the reduction of the percentage EVM to 6.3% from 11.33 % at a bit rate of 20 Gbps using 64 QAM. From this reduction in EVM, it can be concluded that the utilisation of preamble-averaging equalisation can contribute to the reduction in the transmit DMT signal power required to maintain the 5G standard EVMs using different modulation formats for different bit rates.

From the simulation of the DMT-type fronthaul link for different bit rates and advanced modulation formats, a reduction in DMT signal power of up to 20.26 dB was observed for the variation of data rates between 5 Gbps and 200 Gbps using 256 QAM. For a fixed data rate of 200 Gbps, DMT signal power saving of up to 12.46 dB was observed when the modulation format was transitioned from 256 QAM to 4 QAM. So it was concluded that the reduction of fronthaul bit rates can result in the reduction of the transmit DMT signal powers. The reduction also manifests when the modulation levels are reduced for a fixed bit rate however at the cost of a higher bandwidth consumption.

A model for the overall SNR of the DMT transmission link under test was developed, to compare the practical spectral efficiency with the Shannon's capacity. It was observed that the spectral efficiency of the link was around 1 [b/s/Hz] below Shannon's capacity for different bit rates and modulation formats, leading to the conclusion that there is a margin for further improvement in the performance of the link.

From the data gathered from simulations for different bit rates and modulation formats, the transmit DMT signal power required to transmit real-world traffic patterns, in graduations of 5 Gbps, predicted by a DRL agent, was calculated within fixed bandwidth limits. On average the DMT signal power savings of up to 34.6% within the limits of 18 GHz bandwidth was observed when input traffic prediction and subsequent slot allocation is carried out by a DRL agent, in comparison to fixed slot allocation to variable input traffic. So it can be concluded that the DRL-based automation of slot allocation in FlexE can facilitate improved transmit signal power savings, adding to the potential for power efficiency in beyond-5G networks.

References

[1] B. S. G. Pillai et al., 'End-to-End Energy Modeling and Analysis of Long-Haul Coherent Transmission Systems', Journal of Lightwave Technology, vol. 32, no. 18, pp. 3093–3111, Sep. 2014, doi: 10.1109/JLT.2014.2331086.

[2] R. S. Tucker, 'Green Optical Communications—Part I: Energy Limitations in Transport', IEEE Journal of Selected Topics in Quantum Electronics, vol. 17, no. 2, pp. 245–260, Mar. 2011, doi: 10.1109/JSTQE.2010.2051216.

[3] S. Wolf et al., 'DAC-Less Amplifier-Less Generation and Transmission of QAM Signals Using Sub-Volt Silicon-Organic Hybrid Modulators', Journal of Lightwave Technology, vol. 33, no. 7, pp. 1425–1432, Apr. 2015, doi: 10.1109/JLT.2015.2394511.

[4] C. Haffner et al., 'Plasmonic Organic Hybrid Modulators—Scaling Highest Speed Photonics to the Microscale', Proceedings of the IEEE, vol. 104, no. 12, pp. 2362–2379, Dec. 2016, doi: 10.1109/JPROC.2016.2547990.

[5] 'HMC7150 Datasheet and Product Info | Analog Devices'. Accessed: Dec. 05, 2023. [Online]. Available: https://www.analog.com/en/products/hmc7150.html#product-overview.

[6] H. Enzinger, K. Freiberger, and C. Vogel, 'A joint linearity-efficiency model of radio frequency power amplifiers', in 2016 IEEE International Symposium on Circuits and Systems (ISCAS), May 2016, pp. 281–284. doi: 10.1109/ISCAS.2016.7527225.

[7] L. A. da Silva, W. Tatinian, and G. Jacquemod, 'BER and power consumption estimation based on hierarchical modeling of a 2.4 GHz power amplifier', in 2010 IEEE International Behavioral Modeling and Simulation Workshop, Sep. 2010, pp. 31–35. doi: 10.1109/BMAS.2010.6156594.

[8] M. Vasic, O. Garcia, J. A. Oliver, P. Alou, D. Diaz, and J. A. Cobos, 'Multilevel Power Supply for High Efficiency RF Amplifiers', in 2009 Twenty-Fourth Annual IEEE

Applied Power Electronics Conference and Exposition, Feb. 2009, pp. 1233–1238. doi: 10.1109/APEC.2009.4802821.

[9] C.-M. Lai, J.-K. Jau, and J.-Y. Li, ‘High efficiency RF power amplifier using band limited dynamic supply control method’, in 2009 European Microwave Conference (EuMC), Sep. 2009, pp. 350–353. doi: 10.23919/EUMC.2009.5296175.

[10] ‘DC/DC Converter Modules Optimized for RF Power Amplifiers’, DigiKey. Accessed: Jan. 11, 2024. [Online]. Available: https://www.digikey.com/en/articles/dcdc-converter-modules-optimized-for-rf-power-amplifiers

[11] X. Ruan, Y. Wang, and Q. Jin, ‘A review of envelope tracking power supply for mobile communication systems’, CPSS Transactions on Power Electronics and Applications, vol. 2, no. 4, pp. 277–291, Dec. 2017, doi: 10.24295/CPSSTPEA.2017.00026.

[12] M. Rodriguez, P. F. Miaja, A. Rodriguez, and J. Sebastian, ‘Multilevel converter for Envelope Tracking in RF power amplifiers’, in 2009 IEEE Energy Conversion Congress and Exposition, Sep. 2009, pp. 503–510. doi: 10.1109/ECCE.2009.5316317.

[13] M. Jinno, ‘Elastic Optical Networking: Roles and Benefits in Beyond 100-Gb/s Era’, Journal of Lightwave Technology, vol. 35, no. 5, pp. 1116–1124, Mar. 2017, doi: 10.1109/JLT.2016.2642480.

[14] J. P. Fernández-Palacios, V. López, B. de la Cruz, O. Gerstel, N. Sambo, and E. Riccardi, ‘Sliceable Bandwidth Variable Transponders’, in Elastic Optical Networks: Architectures, Technologies, and Control, V. López and L. Velasco, Eds., Cham: Springer International Publishing, 2016, pp. 159–188. doi: 10.1007/978-3-319-30174-7-7.

[15] R. Proietti, H. Lu, G. Liu, A. Castro, M. Shamsabardeh, and S. J. B. Yoo, ‘Experimental Demonstration of Elastic RF-Optical Networking (ERON) for 5G mm-wave Systems’, in 2017 European Conference on Optical Communication (ECOC), Sep. 2017, pp. 1–3. doi: 10.1109/ECOC.2017.8346014.

[16] J. Bartelt et al., ‘5G transport network requirements for the next generation fronthaul interface’, EURASIP Journal on Wireless Communications and Networking, vol. 2017, no. 1, p. 89, May 2017, doi: 10.1186/s13638-017-0874-7.

[17] Y. Liao, S. A. Hashemi, H. ElBakoury, J. Cioffi, and A. Goldsmith, ‘Calendar Allocation Based on Client Traffic in the Flexible Ethernet Standard’, in ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Jun. 2020, pp. 1–6. doi: 10.1109/ICC40277.2020.9149293.

[18] M. N. Chughtai et al., ‘User and resource allocation in latency constrained Xhaul via reinforcement learning’, Journal of Optical Communications and Networking, vol. 15, no. 4, pp. 219–228, Apr. 2023, doi: 10.1364/JOCN.485029.

[19] Y. Ueda et al., ‘Low Driving Voltage Operation of MZI-Type EA Modulator Integrated With DFB Laser Using Optical Absorption and Interferometric Extinction’, IEEE Journal of Selected Topics in Quantum Electronics, vol. 21, no. 6, pp. 195–200, Nov. 2015, doi: 10.1109/JSTQE.2015.2420591.

[20] J. A. Hernández, G. Otero, D. Larrabeiti, and Ó. G. de Dios, ‘Dimensioning Flex Ethernet Groups for the transport of 5G NR fronthaul traffic in C-RAN scenarios’, in 2021 International Conference on Optical Network Design and Modeling (ONDM), Jun. 2021, pp. 1–3. doi: 10.23919/ONDM51796.2021.9492417.

[21] E. Moutaly et al., "Phase Modulated Radio-Over-Fiber for Efficient 5G Fronthaul Uplink," in Journal of Lightwave Technology, vol. 37, no. 23, pp. 5821-5832, 1 Dec.1, 2019, doi: 10.1109/JLT.2019.2940200.

[22] S. Noor, P. Assimakopoulos, and N. J. Gomes, “A Flexible Subcarrier Multiplexing System With Analog Transport and Digital Processing for 5G (and Beyond) Fronthaul,” Journal of Lightwave Technology, vol. 37, no. 14, pp. 3689–3700, Jul. 2019, doi: 10.1109/JLT.2019.2918215.

[23] G. C. Wilson et al., ‘Predistortion of electroabsorption modulators for analog CATV systems at 1.55 /spl mu/m’, Journal of Lightwave Technology, vol. 15, no. 9, pp. 1654–1662, Sep. 1997, doi: 10.1109/50.622890.

[24] Y. Shen, B. Hraimel, X. Zhang, G. E. R. Cowan, K. Wu, and T. Liu, ‘A Novel Analog Broadband RF Predistortion Circuit to Linearize Electro-Absorption Modulators in Multiband OFDM Radio-Over-Fiber Systems’, IEEE Transactions on Microwave Theory and Techniques, vol. 58, no. 11, pp. 3327–3335, Nov. 2010, doi: 10.1109/TMTT.2010.2074530.

[25] ‘60G-R-EAM-1550 Reflective Electro absorption Modulator (EAM)’. [Online]. Available: http://www.amstechnologies.com/fileadmin/amsmedia/downloads/3719_60G-R-EAM-1550_%28A%29.pdf

[26] K. Van Acker, G. Leus, M. Moonen, O. van de Wiel, and T. Pollet, “Per tone equalization for DMT-based systems,” IEEE Transactions on Communications, vol. 49, no. 1, pp. 109–119, Jan. 2001, doi: 10.1109/26.898255.

[27] J. M. Cioffi, “Equalization.” [Online]. Available: https://cioffi-group.stanford.edu/doc/book/chap3.pdf

[28] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, “Channel Estimation,” in MIMO-OFDM Wireless Communications with MATLAB, 1st edition., Singapore; Hoboken, NJ: Wiley-IEEE Press, 2010.

[29] D. Che and X. Chen, “Modulation Format and Digital Signal Processing for IM-DD Optics at Post-200G Era,” Journal of Lightwave Technology, vol. 42, no. 2, pp. 588–605, Jan. 2024, doi: 10.1109/JLT.2023.3311716.

[30] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, “Deep Learning-Based Channel Estimation,” IEEE Communications Letters, vol. 23, no. 4, pp. 652–655, Apr. 2019, doi: 10.1109/LCOMM.2019.2898944.

[31] 3GPP, ‘5G NR; Base Station (BS) radio transmission and reception (3GPP TS 38.104 version 16.4.0 Release 16)’. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138100_138199/138104/16.04.00_60/ts_138104v160400p.pdf

[32] D. Tse and P. Viswanath, ‘Capacity of Wireless Channels’, in Fundamentals of Wireless Communication, Illustrated edition., Cambridge, UK; New York: Cambridge University Press, 2005.

[33] G. L. Li and P. K. L. Yu, "Optical intensity modulators for digital and analog applications," in Journal of Lightwave Technology, vol. 21, no. 9, pp. 2010-2030, Sept. 2003, doi: 10.1109/JLT.2003.815654.

[34] G. P. Agrawal, ‘Loss Management’, in Fiber-Optic Communication Systems, John Wiley and Sons, 2012.

[35] W. S. C. Chang, ‘Noise figure’, in RF Photonic Technology in Optical Fiber Links, Cambridge University Press, 2002.

Chapter 6

Conclusions and Future Research Directions

The research in the thesis aims to introduce autonomous user allocation in the RAN to different paths and resource allocation in fronthaul for beyond-5G mobile networks via deep reinforcement learning. For this, a pre-developed CoMP beamforming model for user throughput calculations was used with a model of a fronthaul with a FlexE calendar or shim. The model for fronthaul incorporated resource allocation in graduations of 5 Gbps in a FlexE shim. For simulations, three agents were concurrently trained, two agents for the fronthaul traffic prediction and subsequent resource allocation in the FlexE shim and one agent for user allocation to the paths with lower latency in fronthaul. In conducting the research for user and resource allocation using DRL, a model for end-to-end latency in a packet network similar to a fronthaul network was also developed to indicate the impact of fronthaul network latency on user throughput if CoMP is used for beamforming.

In the third aspect of the research, the physical layer transport of fronthaul traffic with 5 Gbps granularity was demonstrated using a simulation of a DMT-based fronthaul link equipped with BVTs. Simulation for different bit rates was carried out using different modulation formats and bandwidths. The results for fronthaul bit rate reduction via traffic predictions using DRL in the first aspect of the research were used to demonstrate reductions in transmit DMT signal power in the fronthaul link.

6.1 Conclusions

The conclusions for the different aspects of research in the thesis are also summarized below for each chapter.

Conclusions from Chapter 3

In Chapter 3, a model was developed for end-to-end percentile delays for a packet network similar to a fronthaul. The model utilised the network parameters of loads, input arrival rate, packet service rate, statistics of inter-arrival times of packets, and service time of packets. The model was benchmarked with percentile delays measured from simulations of a packet network similar to a fronthaul.

To model the percentile delays, the mean delays and the statistics of inter-arrival times of packets were required for the G/G/1 and Lindley's recursion-based model. From the comparison of mean waiting times calculated from packet traces with those calculated using the G/G/1 and Lindley's recursion models, it was concluded that, for loads above 0.85, Lindley's recursion-based estimation proved to be a better upper bound on the estimation of mean waiting times than the G/G/1 queuing delay model.

From the analysis of packet traces, it was concluded that the statistics of inter-arrival times of packets matched a generalised Pareto distribution at aggregation nodes, and the variance of inter-arrival times of packets was correlated with the mean inter-arrival times of packets at aggregation nodes. The statistical distribution of the overall end-to-end delays was found to be most similar to the log-normal distribution in terms of RMS error between the delay distribution calculated from packet traces and the modelled delay distribution. Additionally, based on the results of percentage errors between the predicted and measured percentile delays, it was found that the model did not underpredict the percentile delays and can be considered a better upper bound than using only the G/G/1 queuing model. It was also observed from the comparison of simulation results that over-predictions increase with increasing percentiles.

The predicted percentile delays were limited to 99% and 99.9% due to time and memory resource constraints. The results can be extended to higher percentiles (i.e., 99.999% or above) using the proposed model. However, testing these cases will require significantly longer simulation times and larger packet traces to gather sufficient statistics.

In the simulation cases for testing end-to-end delay, packet lengths were randomly generated from a Normal distribution. For networks with different statistics of packet lengths and inter-arrival times, the model can be easily modified by incorporating relevant probability density functions (PDFs) into Lindley's recursions for estimating end-to-end percentile delays.

Conclusions from Chapter 4

In Chapter 4, a combined resource and user allocation scheme in a fronthaul network with Flexible Ethernet through DRL was shown. The DRL agent for slot allocation in the FlexE calendar predicted real-world traffic patterns from the WIDE MAWI project and allocated slots accordingly. These slot allocations improved bandwidth efficiency in fronthaul, as measured by mean overallocation, in comparison to constant allocation and ARIMA-based predictions. The DRL-based setup efficiently allocated resources by up to 11.6% compared to classical ARIMA-based allocations.

A second agent was deployed to improve the total bit rate of users linked to different paths of fronthaul by offloading them from paths with higher latency to paths with lower latency. It was observed that the total bit rate of users improved by up to 7% via user offloading enabled by the DRL agent.

It was concluded that the DRL agents can interact with the environment and learn different traffic patterns with different statistical properties. The subsequent resource or slot allocation enhances the bandwidth efficiency. The DRL agents can also learn to offload users among RUs to improve the user bit rates in RAN to circumvent the impact of latency in the fronthaul network. However, this learning came at the cost of significant computation time and resources related to training the deep-Q networks and storing experiences in an experience buffer. Additionally, fine-tuning DRL also involved adjusting reward functions and hyper-parameters, including the number of steps to look ahead, the discount factor, the epsilon greedy factor, the experience buffer size, and the mini-batch size.

Conclusions from Chapter 5

In Chapter 5, the benefit of using DRL for traffic prediction in transporting varying numbers of FlexE slots within a 5G network fronthaul link was discussed. The benefit was demonstrated through the reduction in transmit DMT signal power when employing BVTs, for modulation level and bandwidth variation, to maintain the receiver SNRs.

A reduction in DMT signal power consumption of up to 20.26 dB was observed across data rates ranging from 5 Gbps to 200 Gbps using 256 QAM. Additionally, a fixed data rate of 200 Gbps exhibited a DMT signal power saving of up to 12.46 dB when transitioning from 256 QAM to 4 QAM modulation.

On average, transmit DMT signal power savings of up to 34.6% within an 18 GHz bandwidth were observed when employing input traffic prediction and subsequent slot allocation by a DRL agent, compared to fixed slot allocation for variable input traffic. It was concluded that automating slot allocation in FlexE using DRL can significantly enhance transmit signal power savings, via a reduction of the bit rates from a maximum constant bit rate in the fronthaul segment. The transmit DMT signal power reduction has the potential to lower the transmitter power consumption in a DMT-based fibre optic transmission link, which can further contribute to improving the overall power efficiency of the beyond-5G mobile networks.

Overall conclusion

The thesis research presents a comprehensive approach to optimising fronthaul of beyond-5G mobile networks, through DRL-based traffic pattern predictions and user offloading. DRL's ability to predict traffic and allocate resources in FlexE-based fronthaul outperformed classical approaches such as ARIMA-based schemes by adapting to traffic patterns dynamically. The DRL controller agent was also able to improve user bit rates by offloading users from paths with higher latency to those with lower latency, thus mitigating the latency effects in the fronthaul.

Additionally, the study also models percentile delays in packet networks to provide upper-bound estimations for a wide range of load scenarios, applicable to fronthaul networks. The percentile delay model can be utilised by DRL agents in network controllers to assess the quality of a path in terms of latency for user offloading.

In addition to resource allocation, DRL has demonstrated significant transmit signal power savings in fibre optic fronthaul links. By utilising DRL for traffic prediction and slot-based bandwidth allocation, up to 34.6% savings in the DMT signal driving the EAM for fronthaul links were achieved. This was possible by adapting the QAM levels for different subcarriers, leading to reduced power consumption in the transmitter's modulator unit, which is responsible for a significant portion of the overall link power consumption.

Beyond the conclusions of individual chapters, the study illustrates how the combined use of DRL and advanced queuing models represents a framework for optimising resource allocation, reducing power consumption, and managing latency in fronthaul of beyond 5G mobile networks. This positions DRL-based automation as a key enabler of sustainable, scalable beyond-5G mobile network infrastructure capable of meeting future demands.

6.2 Future Research Directions

The overall DRL-based system discussed in Chapter 4 had three agents that were trained jointly and deployed to allocate resources and users. The fronthaul traffic, that was predicted by the FlexE agents in the research, comprised a sum of the background traffic and user traffic. The training of the controller agent, impacted the training of the FlexE agent since the user traffic being added to the background traffic depended on the number of users being offloaded by the controller agent. On the other hand, the training of the FlexE agent did not directly impact the controller agent. To make the system complete it is proposed for future research, that the estimates of the latency should be based on the estimates of the FlexE agent actions or prediction of total input traffic. This will in turn impact the user bit rates in accordance with the estimates of latency based on FlexE agent traffic predictions and ultimately impact the training of the controller agent.

In a fronthaul network, the CU/DU performs many functions such as offloading of UEs as discussed in the thesis, selection of function splits and routing of traffic in the fronthaul network under different constraints such as latency of end-to-end path. For future research, the routing function of the controller agent aided by DRL should be implemented, in conjunction with the functions of resource and user allocations. The implementation should be tested in a more complex fronthaul network model comprising a larger number of aggregation and routing nodes.

The model for end-to-end latency used in the CoMP model discussed in Chapter 3 used the G/G/1 queuing model and Lindley's recursions. The modelling of PDFs at source and aggregation nodes using Lindley's recursion involves convolution operations. The computation time of the convolution operations can be reduced by implementing the convolutions in the frequency domain using the IFFT operation. However, the precision of convolution operations utilizing IFFT operations for Lindley's recursion warrants assessment in future research. The research carried out in Chapter 3 for modelling of percentile delays can extended to the extreme percentiles above 99.99 % by running longer runs of simulations, requiring more memory resources. Further, the model can also be evaluated by analysing end-to-end percentile delays measured from packet traces extracted from real-world 5G fronthaul networks. The evaluation can also add more validity to the results.

In Chapter 5, the reduction of transmit DMT signal power in driving the electrical pre-amplifier and the optical modulator within the modulator unit was examined. By modelling the power consumption of commercially available drive amplifiers and modulators used in practical and experimental DMT-type transmission links, precise figures for the reductions in the transmitter power consumption can be obtained. Subsequently, other components' power consumption in both the transmitter and receiver can be modelled which include the power consumption of the laser source, the EDFA, the digital-to-analogue converter, the receiver photo-diode, the receiver electrical amplifier, the analogue-to-digital converter and DSP operations. Modelling these components will aid in assessing the overall reduction in power consumption of the entire fronthaul link resulting from decreased bit rates.

For future research, further power savings can be achieved via the development of optimisation algorithms that can concurrently adjust the parameters such as the laser source output optical power, EDFA gain, and gains of electrical amplifiers in the receiver, at different FlexE bit rates. The optimisation algorithms can also be aided by deep reinforcement learning to adjust different component parameters to achieve the highest possible power efficiency for every possible fronthaul bit rate.